Environmental-Computational Epistemic Agents: Observations for Optimal Decisions

Thong Quoc Huynh[1][0009−0002−7614−8952], Oren Salzman[2][0000−0003−4541−8219], and Neil T. Dantam[1][0000−0002−0907−2241]

[1] Colorado School of Mines, Golden CO 80401, USA
{huynh,ndantam}@mines.edu
[2] Technion - Israel Institute of Technology, Haifa 3200003, Israel
osalzman@cs.technion.ac.il

Abstract. Uncertainty is a fundamental challenge in robotics. While probabilistic methods effectively handle inherent randomness, epistemic uncertainty—arising from a robot's limited knowledge—remains a largely open problem. However, a robot's embodiment offers a unique solution: the ability to actively resolve this uncertainty through targeted action and observation. In this work, we address two distinct sources of epistemic uncertainty: limits on environmental knowledge and limits on computational tractability. We unify these within a Bounded-parameter Markov Decision Process (BMDP), where transition probabilities are modeled as intervals rather than exact values. We propose a metric that quantifies the deviation from the true optimal policy by accounting for both environmental ambiguity and computational optimality gaps. To minimize this loss, we develop a hierarchical optimization framework that identifies the most informative parameter observations. Experimental results on grid-world domains demonstrate that our efficient approximation method reduces computation time by a factor of roughly 30× compared to exact methods. Furthermore, our active observation strategy converges to the optimal policy with up to 5× fewer observations than random baselines in small domains, and maintains a 1.5× to 2× convergence advantage in larger environments where exact reasoning is intractable.

Keywords: Formal Methods · Task Planning · Mathematical Modeling and Analysis

1 Introduction

Robots would be more effective if they understood the limits of their understanding. Uncertainty is a defining challenge in robotics. Probabilistic approaches have offered fruitful capabilities [35], yet *epistemic uncertainty* presents open challenges. Classically, epistemic uncertainties arise from incomplete knowledge of the world. We argue that a further source of epistemic uncertainty relevant to robots arises from computational limits due to complexity [22] and decidability [7, 36]. Embodiment, a defining capability in robotics, offers opportunities to resolve epistemic uncertainty through action and observation. A running example for

observations that we will use in this work is a measurement of road conditions to determine iciness and therefore slippage, which reduces uncertainty about the environment to better determine optimal actions.

We jointly address both limits on knowledge and limits on reasoning by formulating action under epistemic uncertainty as a Bounded-parameter Markov Decision Process and developing algorithms for optimal and near-optimal observations of parameters. Incomplete knowledge about the world becomes bounds or intervals for Markov Decision Process (MDP) parameters. Uncertainty in reasoning presents optimality gaps between a known policy value and a potential true optimum policy value. Fusing these two sources of epistemic uncertainty produces a *policy loss* between best-case and worst-case policy values given our uncertainty. We formulate an optimization model to identify parameter observations to minimize policy loss. Exact solutions to this optimization model are computationally challenging, so we identify efficient approximations with resulting optimality gaps that we incorporate as epistemic uncertainties into our policy loss optimization. Evaluation results show that our method efficiently identifies parameter observations to reduce uncertainty and converges to the true optimal policy with up to 5× fewer observations than the random selection approach.

2 Related Work

Our work bridges the gap between uncertainty modeling in Markov Decision Processes (MDPs) and computational optimization. We review relevant literature in bounded-parameter models, optimization-based approaches to sequential decision-making, computational tools required, and some prior works in robotic epistemic uncertainty.

2.1 Bounded-Parameter and Robust MDPs

Standard MDPs assume precise knowledge of transition probabilities and rewards. However, when these parameters are uncertain, they are often modeled as intervals. Givan et al. formally introduced Bounded-parameter Markov Decision Processes (BMDPs), utilizing closed real intervals to represent state transition probabilities and rewards without assuming a prior distribution [15]. This interval-based representation captures epistemic uncertainty (i.e., uncertainty due to a lack of knowledge) which is distinct from the inherent aleatoric uncertainty of the stochastic process [12, 17].

A closely related field is Robust Dynamic Programming. Iyengar [21] and others have developed methods to find policies that maximize performance under the worst-case parameter realization within these ambiguity sets [13]. While robust MDPs focus on finding a safe policy given a fixed level of uncertainty (typically maximizing the minimum value), our work takes a different approach: we actively seek to *reduce* this uncertainty through observation. We model the ambiguity [13] not just as a constraint to be robust against, but as a variable that can be manipulated and tightened via cost-incurring observations to recover the true optimal policy.

2.2 Optimization Approaches to MDPs

The connection between linear programming (LP) and MDPs was established in the seminal works of De Ghellinck [8], d'Epenoux [11], and Manne [25]. These early approaches demonstrated that finding the optimal value function for a discounted MDP is equivalent to solving a linear program. This formulation is particularly powerful because it allows for the application of extensive constraints and sensitivity analysis, which are difficult to incorporate into standard value-iteration methods [30, 37].

Recent work has extended this LP formulation to handle parameter variations. Avrachenkov et al. [2] investigated singularly perturbed linear programs in the context of MDPs, analyzing how small perturbations in the constraint matrix (representing transition probabilities) affect the stability and limiting behavior of the optimal solution. Our approach builds on this foundation by treating the interval bounds as constraints in a Mixed Integer Program (MIP), where binary decision variables determine which parameter intervals are "perturbed" or tightened via observation.

2.3 Computational Tools and Solvers

The integration of discrete observation decisions with continuous value function optimization transforms the problem into a Mixed Integer Program (MIP). Solving such problems to optimality is computationally demanding due to their NP-hard nature [22].

While classical simplex methods [24] often suffice for standard LPs, modern applications increasingly rely on advanced commercial solvers like Gurobi [16] or CPLEX [20] to handle the combinatorial complexity of integer constraints. These solvers employ sophisticated branch-and-bound and branch-and-cut algorithms [38] that are essential for scaling our proposed method to larger state spaces. By formulating our epistemic planning problem as a MIP, we leverage these robust optimization tools to find the most cost-effective set of observations that guarantee a reduction in policy loss.

2.4 Epistemic Uncertainty in Robotics

Prior approaches have addressed epistemic uncertainty in relation to robotics [4]. Research in this domain ranges from uncertainty quantification to predict future system states [26], to filtering approaches that respect [23] or predict [29] epistemic uncertainty. In the context of planning, Bramblett et al. [6] utilize dynamic epistemic logic and mixed integer programming to propagate belief states and allocate tasks. Uncertain MDP parameters have also been explored in various works [3, 15, 28, 33]. Bayesian reinforcement learning [9, 10, 14, 32] updates beliefs over uncertain parameters after data observations. We do not require explicit prior distributions over the parameters nor full distribution updates, using a formulation following Givan et al. [15]. Our work extends this developing area of epistemic uncertainty in robotics to now leverage robots' key capability of embodiment to resolve epistemic uncertainties.

3 Problem Formulation

We address sequential decision-making under aleatoric and epistemic uncertainty, where we may take observations to reduce epistemic uncertainty. We consider decision-making under aleatoric uncertainty as a Markov Decision Process (MDP) [34], i.e., transition probabilities represent the world's inherent randomness. Epistemic uncertainty about the world means we only know this true MDP's transition probabilities to some interval, forming a Bounded-parameter Markov Decision Process (BMDP) [15]. We may observe the world to collapse an interval in the BMDP to its true value. Returning to our running example about observations, before taking an observation, we do not have information about road conditions—icy versus clear—and only know an interval of probability for successful traversal. After an observation, we reduce this interval and become more certain about the chances of successfully traversing the road. Our aim is to find the best sequence of observations that will enable us to produce a policy with value closest to the true optimal policy.

The true MDP $\mathcal{M} = (\mathcal{S},\ \mathcal{A},\ \mathcal{P},\ \mathcal{R})$ consists of state space $\mathcal{S}$, action space $\mathcal{A}$, transition probabilities $\mathcal{P}$, and rewards $\mathcal{R}$ [34]. At each time step, an agent is in some state $s \in \mathcal{S}$ and may choose to take some action $a \in \mathcal{A}$. Taking an action moves the agent to possible successor states $s' \in \mathcal{S}$ with probability $p(s'|s,a)$ and yielding reward $r(s,a,s')$. The objective is to find an optimal policy $\pi_* : \mathcal{S} \mapsto \mathcal{A}$ that will maximize the agent's expected reward.

Our knowledge of the true MDP $\mathcal{M}$ is limited to a BMDP $[\mathcal{M}] = (\mathcal{S},\ \mathcal{A},\ [\mathcal{P}],\ \mathcal{R})$, where we know transition probabilities only as intervals instead of exact values [15]. Taking an action moves the agent to possible successor states $s' \in \mathcal{S}$ with probability $\underline{p}(s'|s,a) \leq p(s'|s,a) \leq \overline{p}(s'|s,a)$; we know only bounds $\underline{p}(s'|s,a)$ and $\overline{p}(s'|s,a)$, not the exact probability $p(s'|s,a)$. The objective remains finding an optimal policy $\pi_* : \mathcal{S} \mapsto \mathcal{A}$ to maximize the agent's expected reward; however, imprecise parameters in a BMDP mean we can only determine the utility of a policy to an interval. Thus, among the set of policies Π, there is a set of potential optimal policies, $\Pi^{\text{pot}}_* \subset \Pi$ with overlapping utility. Specifically, a potential optimal policy $\pi^{\text{pot}}_* \in \Pi^{\text{pot}}_*$ has upper bound utility $\overline{[\sum v(s)]}$ that is no less than the lower bound $\underline{[\sum v(s)]}$ of any other policy. Denoting v_π as the state value under π, a potential optimal policy π^{pot}_* satisfies:

$$\overline{\left[\sum_s v_{\pi^{\text{pot}}_*}(s)\right]} \geq \underline{\left[\sum_s v_\pi(s)\right]} \quad \forall\ \pi \in \Pi\ . \tag{1}$$

We extend the formulation of BMDPs [15] to incorporate observations of unknown parameters. Observing a transition probability interval will yield its precise value $p(s'|s,a) \in \big[\underline{p}(s'|s,a),\ \overline{p}(s'|s,a)\big]$. Our ultimate goal is to find the minimal sequence of probability intervals to observe so we can identify an optimal π^{true}_*. To do so, we require a metric for deviation from the true optimal policy.

We propose the following metric for deviation from the true optimal policy that we call the *policy loss* Δ, representing how close the potential optimal policies $\pi \in \Pi^{\text{pot}}_*$ are to the true optimal policy π^{true}_*.

Definition 1 (Policy Loss). *For a set of potential optimal policies Π_*^{pot}, the policy loss $\Delta \in \mathbb{R}$ is,*

$$\Delta = \max_{\pi \in \Pi_*^{pot}} \left| v(\pi) - v\left(\pi_*^{true}\right) \right| ,$$

where $v(\pi)$ is the utility of a policy π and π_*^{true} is the true optimal policy.

Computing the exact policy loss for a given BMDP is challenging since we do not know the true optimal policy and finding bounds for utility can be computationally expensive. However, relaxations and approximations can efficiently determine upper and lower bounds on utility, in turn placing an upper bound on policy loss.

Definition 2 (Epistemic Policy Loss). *For a set of potential optimal policies Π_*^{pot}, the policy loss $\Delta \in \mathbb{R}$ is bounded by,*

$$\Delta \leq \max_{\pi \in \Pi_*^{pot}} \overline{v}(\pi) - \min_{\pi \in \Pi_*^{pot}} \underline{v}(\pi) ,$$

where $\overline{v}(\pi), \underline{v}(\pi)$ are upper and lower bounds on utility of policy π.

We describe techniques to find bounds on utility in section 5.

A key feature of this formulation is the combination of environmental epistemic uncertainty (lack of knowledge of the world) and computational epistemic uncertainty (lack of ability to compute exact solutions) into a single measure.

To summarize, we desire a minimal sequence of observations to identify an optimal π_*^{true}. However, as we will see, this will be extremely challenging from a computational point of view. Thus, we focus on computing an observation to minimize the epistemic policy loss (Definition 2), and develop a sequential algorithm iteratively choosing the next best observation.

4 Background

4.1 Markov Decision Processes

We give a brief overview of Markov Decision Process (MDP) fundamentals. For further details, we refer the reader to Sutton and Barto [34] and Puterman [30]. An MDP is a recurring decision problem where the current decision affects the future. A decision maker called the *agent* interacts with the *environment* outside the agent, over a *planning horizon* (timeline). For simplicity, we consider discrete time horizons.

At each time step, the environment is at a *state* s in a set of possible states $\mathcal{S}$. The agent chooses to perform an *action* a in the set $\mathcal{A}(s)$ of actions available at state s. The environment then responds with a new state s' and real numerical *reward* (or *cost*) r. Such decision process is called *Markov* if it satisfies the Markov property, where the future only depends on the current state and action. We consider the reward r as a function of the current state, action, and next

state: $r(s, a, s')$. We denote the *transition probability* of arriving at state s' after taking action a at state s as $p(s'|s, a)$. We can calculate the *expected reward* of a state-action pair as:

$$r(s, a) = \sum_{s' \in \mathcal{S}} p(s'|s, a) r(s, a, s'). \tag{2}$$

The agent essentially implements a *policy* $\pi : \mathcal{S} \mapsto \mathcal{A}$ that maps each state s to a desired action a, i.e., $\pi(s) = a$. At each time step, the probability of taking action a given state s is $\pi(a|s)$. The expected (E) total reward or *value function* v of a given state s when following a policy π is:

$$v_\pi(s) = \lim_{N \to \infty} E \left[\sum_{n=0}^{N} \gamma^n r(s_n, \pi(s_n)) \;\middle|\; s_0 = s \right]. \tag{3}$$

where s_n is the state at time step n, and $0 \leq \gamma \leq 1$ is the discount factor to reduce the effect of future rewards. The recursive form of Eq. (3) is called the *Bellman equation* for v_π:

$$v_\pi(s) = \sum_a \pi(a|s) \left\{ \sum_{s'} p(s'|s, a) \left(r(s, a, s') + \gamma v_\pi(s') \right) \right\}. \tag{4}$$

The general goal of the MDP problem is to find an *optimal policy* π_*, to maximize the total rewards or minimize total costs over the planning horizon. Such optimal policy π_* specifies action decisions that correspond to the optimal value function v_* below, for all $s \in \mathcal{S}$:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} \left\{ \sum_{s' \in \mathcal{S}} p(s'|s, a) \left(r(s, a, s') + \gamma v_*(s') \right) \right\} \tag{5}$$

$$= \max_{a \in \mathcal{A}(s)} \left\{ \sum_{s' \in \mathcal{S}} p(s'|s, a) r(s, a, s') + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, a) v_*(s') \right\}. \tag{6}$$

The term inside max is known as the *action-value function* $q(s, a)$, to distinguish from the *state-value function* $v(s)$:

$$q(s, a) = \sum_{s' \in \mathcal{S}} p(s'|s, a) r(s, a, s') + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, a) v_*(s'). \tag{7}$$

4.2 Interval Linear Programming

Linear programming finds an optimal vector $\mathbf{x}$ under a linear objective and linear constraints. The standard form of a linear program (LP) is,

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{8}$$

where $\mathbf{x}$ is a vector of real decision variables, $\mathbf{c}$ and $\mathbf{b}$ are constant real vectors, and $\mathbf{A}$ is a constant real matrix. Solving LPs has been extensively studied [5,24,38], with many efficient algorithms and tools developed [16, 19, 20].

Interval Linear Programs (ILP) generalize the traditional LP to intervals for $\mathbf{c}, \mathbf{A}, \mathbf{b}$ [18, 37]. Such a formulation effectively defines a family of LPs that satisfy the intervals. The standard form of an ILP is,

$$\begin{array}{ll} \min & \sum_{j=1}^{n} \left[\underline{c}_j,\, \overline{c}_j\right] x_j \\ \text{s.t.} & \sum_{j=1}^{n} \left[\underline{a}_{ij},\, \overline{a}_{ij}\right] x_j \leq \left[\underline{b}_i,\, \overline{b}_i\right],\ \forall i \\ & \mathbf{x} \geq \mathbf{0}. \end{array} \quad \rightsquigarrow \quad \begin{array}{ll} \min & [\mathbf{c}]^T \mathbf{x} \\ \text{s.t.} & [\mathbf{A}]\mathbf{x} \leq [\mathbf{b}] \\ & \mathbf{x} \geq \mathbf{0}. \end{array} \tag{9}$$

Within the family of LPs defined by an ILP, there are two LPs that give lower and upper bounds on the objective across the family. A lower bound on the objective is the solution to,

$$\begin{aligned} \min \quad & \underline{\mathbf{c}}^T \mathbf{x} \\ \text{s.t.} \quad & \overline{[\mathbf{A}]}\mathbf{x} \geq \underline{[\mathbf{b}]} \\ & \mathbf{x} \geq 0\ , \end{aligned} \tag{10}$$

while an upper bound on the objective is the solution to,

$$\begin{aligned} \min \quad & \overline{\mathbf{c}}^T \mathbf{x} \\ \text{s.t.} \quad & \underline{[\mathbf{A}]}\mathbf{x} \geq \overline{[\mathbf{b}]} \\ & \mathbf{x} \geq 0\ . \end{aligned} \tag{11}$$

5 Method

We develop a method to choose observations of BMDP parameters to find a policy closest to the true optimum. We consider the best observations to be those that minimize policy loss (Definition 1). We bound policy loss under Definition 2 by developing techniques to determine upper and lower bounds on the expected reward. Then, we develop an optimization approach to choose the best observations.

5.1 Interval Linear Programs for Bounded-Parameter MDPs

We extend the established LP solution for MDPs [8, 11, 25] to identify bounds for BMDPs. Intervals on BMDP parameters result in an ILP. This ILP provides a lower bound on the BMDP's expected reward, but further considerations are required for upper bounds, which we address in the next subsections.

Linear Programs for MDPs The established LP approach to standard MDPs [8, 11, 25] recognizes that the solution to the Bellman optimal equation (6) is the smallest $v(s)$ that satisfies the inequality:

$$v(s) \geq \sum_{s' \in \mathcal{S}} p(s'|s,a) r(s,a,s') + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) v(s') \quad \forall\ s \in \mathcal{S},\ a \in \mathcal{A}(s).$$

Therefore, the following LP [30] provides the optimal value function v_* corresponding to the optimal policy π_*:

$$\begin{aligned} \min \quad & \sum_{s\in\mathcal{S}} v(s) \\ \text{s.t.} \quad & v(s) - \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a)v(s') \geq \sum_{s'\in\mathcal{S}} p(s'|s,a)r(s,a,s') \quad \forall\ s\in\mathcal{S},\ a\in\mathcal{A}(s), \end{aligned} \tag{12}$$

where each state value $v(s)\ \forall\ s\in\mathcal{S}$ is a continuous decision variable.

Linear Programs for BMDPs With interval transition probabilities $[p(s'|s,a)]$, Eq. (12) becomes an ILP,

$$\begin{aligned} \min \quad & \sum_{s\in\mathcal{S}} v(s) \\ \text{s.t.} \quad & v(s) - \gamma \sum_{s'\in\mathcal{S}} [p(s'|s,a)]\, v(s') \geq \sum_{s'\in\mathcal{S}} [p(s'|s,a)]\, r(s,a,s') \quad \forall\ s\in\mathcal{S},\ a\in\mathcal{A}(s) \end{aligned} \tag{13}$$

Applying the ILP lower bound (Eq (10)) offers a lower bound on the expected utility. However, Eq. (10) uses lower bounds on transition probabilities which may sum to less than one, introducing looseness into this lower bound.

On the other hand, the ILP upper bound (Eq (11)) does not offer us a way to find upper bounds on expected utility. The upper bound transition probabilities in Eq. (11) may sum up to more than one, resulting in an infeasible system. In the following subsections, we identify other approaches to find upper bounds on the expected utility.

5.2 An Optimization Model for Bounded-Parameter MDPs

We develop an optimization model for BMDPs. The key feature of the model is to treat transition probabilities as decision variables and to constrain probabilities to sum to one. This constraint on transition probabilities addresses the issues of the ILP formulation in subsection 5.1 leading to loose lower bounds and infeasible upper bound problems. We further formulate all constraints as equalities by introducing additional decision variables for the value of a state-action pair $q(s,a)$, so we can pose the lower and upper bound models as minimization and maximization, respectively.

The lower bound on the expected reward is given by the following model,

$$\min \sum_{s\in\mathcal{S}} v_{\mathrm{LB}}(s) \tag{14}$$

$$\text{s.t.} \quad v_{\mathrm{LB}}(s) = \max_{a\in\mathcal{A}(s)} q_{\mathrm{LB}}(s,a) \quad \forall\ s\in\mathcal{S} \tag{15}$$

$$q_{\mathrm{LB}}(s,a) = \sum_{s'\in\mathcal{S}} p_{sas'} r(s,a,s') + \gamma \sum_{s'\in\mathcal{S}} p_{sas'} v_{\mathrm{LB}}(s') \quad \forall\ s,a\in\mathcal{S},\mathcal{A}(s) \tag{16}$$

$$\overline{p(s'|s,a)} \geq p_{sas'} \geq \underline{p(s'|s,a)} \quad \forall\ s,s'\in\mathcal{S},\ a\in\mathcal{A}(s) \tag{17}$$

$$\sum_{s'\in\mathcal{S}} p_{sas'} = 1 \quad \forall\ s\in\mathcal{S},\ a\in\mathcal{A}(s)\ , \tag{18}$$

and the upper bound on the expected reward is given by the following model,

$$\max \sum_{s\in\mathcal{S}} v_{\mathrm{UB}}(s) \tag{19}$$

$$\text{s.t.} \quad v_{\mathrm{UB}}(s) = \max_{a\in\mathcal{A}(s)} q_{\mathrm{UB}}(s,a) \quad \forall\ s\in\mathcal{S} \tag{20}$$

$$q_{\mathrm{UB}}(s,a) = \sum_{s'\in\mathcal{S}} \hat{p}_{sas'} r(s,a,s') + \gamma \sum_{s'\in\mathcal{S}} \hat{p}_{sas'} v_{\mathrm{UB}}(s') \quad \forall\ s,a\in\mathcal{S},\mathcal{A}(s) \tag{21}$$

$$\overline{p(s'|s,a)} \geq \hat{p}_{sas'} \geq \underline{p(s'|s,a)} \quad \forall\ s,s'\in\mathcal{S},\ a\in\mathcal{A}(s) \tag{22}$$

$$\sum_{s'\in\mathcal{S}} \hat{p}_{sas'} = 1 \quad \forall\ s\in\mathcal{S},\ a\in\mathcal{A}(s)\ , \tag{23}$$

where we have additional decision variables $q(s,a)$ being the values of state-action pairs, $p_{sas'}$ being probabilities for the lower bound, and $\hat{p}_{sas'}$ being probabilities for the upper bound.

Eq. (16) and (21) contain products of decision variables, resulting in a bilinear program. Such programs are solvable via spatial branch and bound techniques [38], which iteratively yield a best-known incumbent solution and an *optimality gap* ε describing a deviation between the incumbent solution's value and a bound on the true optimum. Solvers typically [16, 20] report optimality gaps as relative values,

$$\varepsilon = \frac{|\mathsf{bestbound} - \mathsf{incumbent}|}{|\mathsf{incumbent}|}\ . \tag{24}$$

We incorporate optimality gaps into epistemic policy loss under Definition 2,

$$\Delta \leq \overbrace{(1+\varepsilon_{\mathrm{UB}})\sum_{s\in\mathcal{S}} v_{\mathrm{UB}}(s)}^{\max_{\pi\in\Pi^{\mathrm{pot}}_*} \overline{v}(\pi)} - \overbrace{(1-\varepsilon_{\mathrm{LB}})\sum_{s\in\mathcal{S}} v_{\mathrm{LB}}(s)}^{\min_{\pi\in\Pi^{\mathrm{pot}}_*} \underline{v}(\pi)}, \tag{25}$$

where $\varepsilon_{\mathrm{UB}}, \varepsilon_{\mathrm{LB}}$ are optimality gaps for the upper and lower bound problems, respectively. Such incorporation of optimality gaps accounts for computational epistemic uncertainty.

Bounds on decisions variables help to efficiently solve this optimization model. Bounds on transition probabilities come from the BMDP. State values are loosely bound by a geometric series of discount factor γ and the maximum reward across states $r_{\max}$,

$$v(s) \leq \sum_{k=0}^{\infty} (\gamma \cdot r_{\max}) = \frac{r_{\max}}{1-\gamma} \ . \tag{26}$$

We tighten the bound in Eq. (26) by replacing the initial h terms of the geometric series with the maximum reward reachable at the step corresponding to the term, which algorithm 1 computes.

Algorithm 1: Reachable Set Reward Bound

Input: $(\mathcal{S},\ \mathcal{A},\ [p],\ r)$; /* Interval MDP */
Input: $(h,\ \gamma)$; /* Lookahead Horizon, Discount Factor */
Output: B; /* Value Upper Bounds */

function lookeahead(σ, β, k) **is**
  if $k \geq h$ **then return** β;
  else
    $\beta' \leftarrow \beta + \gamma^k \cdot \max_\sigma r(\sigma)$; // upper bound of current reward
    $\sigma' \leftarrow \{\, s' \in \mathcal{S} \mid \exists s \in \sigma,\ a \in \mathcal{A},\ \overline{p_{sas'}} > 0\}$; // next reachable states
    return lookahead$(\sigma', \beta', k+1)$;

foreach $s \in \mathcal{S}$ **do** $B_s \leftarrow \frac{r_{\max}}{1-\gamma} - \sum_{k=0}^{h-1} \gamma^k \cdot r_{\max} +$ lookahead$(\{s\}, 0, 0)$;

5.3 Optimal Observation

We now formulate an optimization model to choose an observation that best minimizes policy loss. For our running example, choosing an optimal observation is equivalent to selecting the road section whose condition measurement would be the most informative to the uncertain chances of successful traversal. A particular observation of a probability interval introduces a coupling between the upper and lower bound problems of subsection 5.2, $p_{sas'} = \hat{p}_{sas'}$, bringing the bounds closer and reducing policy loss. Since we do not know what specific probability value we will observe, we consider a worst-case, i.e., the observed probability that would result in a maximum policy loss after the observation.

We consider the selection of a limited number of observations at a time. Each observation introduces coupling $p_{sas'} = \hat{p}_{sas'}$ between upper and lower bound problems. We represent taking an observation of $p_{sas'}$ with Boolean decision variable $y_{sas'}$ and an indicator constraint for the coupling,

$$y_{sas'} \implies (p_{sas'} = \hat{p}_{sas'}) \ . \tag{27}$$

The resulting optimization model merges the upper and lower bound problems of subsection 5.2, modifies the objective (28), adds additional indicator coupling

constraints (38), and limits the number of possible observations to $y_{\max}$ (39),

$$\underset{y_{sas'}}{\operatorname{argmin}} \left\{ \max_{\hat{p}_{sas'},\, p_{sas'}} \left(\overbrace{\max_{v_{\mathrm{UB}}(s)} \sum_{s\in\mathcal{S}} v_{\mathrm{UB}}(s)}^{\mathrm{UB}} - \overbrace{\min_{v_{\mathrm{LB}}(s)} \sum_{s\in\mathcal{S}} v_{\mathrm{LB}}(s)}^{\mathrm{LB}} \right) \right\} \tag{28}$$

$$\text{s.t.} \quad v_{\mathrm{UB}}(s) = \max_{a\in\mathcal{A}(s)} q_{\mathrm{UB}}(s,a) \quad \forall\ s\in\mathcal{S} \tag{29}$$

$$q_{\mathrm{UB}}(s,a) = \sum_{s'\in\mathcal{S}} \hat{p}_{sas'} r(s,a,s') + \gamma \sum_{s'\in\mathcal{S}} \hat{p}_{sas'} v_{\mathrm{UB}}(s') \quad \forall\ s\in\mathcal{S},\ a\in\mathcal{A}(s) \tag{30}$$

$$v_{\mathrm{LB}}(s) = \max_{a\in\mathcal{A}(s)} q_{\mathrm{LB}}(s,a) \quad \forall\ s\in\mathcal{S} \tag{31}$$

$$q_{\mathrm{LB}}(s,a) = \sum_{s'\in\mathcal{S}} p_{sas'} r(s,a,s') + \gamma \sum_{s'\in\mathcal{S}} p_{sas'} v_{\mathrm{LB}}(s') \quad \forall\ s\in\mathcal{S},\ a\in\mathcal{A}(s) \tag{32}$$

$$\overline{p(s'|s,a)} \geq \hat{p}_{sas'} \geq \underline{p(s'|s,a)} \quad \forall\ s,s'\in\mathcal{S},\ a\in\mathcal{A}(s) \tag{33}$$

$$\overline{p(s'|s,a)} \geq p_{sas'} \geq \underline{p(s'|s,a)} \quad \forall\ s,s'\in\mathcal{S},\ a\in\mathcal{A}(s) \tag{34}$$

$$\sum_{s'\in\mathcal{S}} \hat{p}_{sas'} = 1 \quad \forall\ s\in\mathcal{S},\ a\in\mathcal{A}(s) \tag{35}$$

$$\sum_{s'\in\mathcal{S}} p_{sas'} = 1 \quad \forall\ s\in\mathcal{S},\ a\in\mathcal{A}(s) \tag{36}$$

$$y_{sas'} \in \{0,1\} \quad \forall\ s,s'\in\mathcal{S},\ a\in\mathcal{A}(s) \tag{37}$$

$$y_{sas'} \implies (p_{sas'} = \hat{p}_{sas'}) \quad \forall\ s,s'\in\mathcal{S},\ a\in\mathcal{A}(s) \tag{38}$$

$$\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}(s)} \sum_{s'\in\mathcal{S}} y_{sas'} \leq y_{\max} \tag{39}$$

While this optimization model describes the best observations, the complex objective (28) is unsupported by commonly available tools [16,20]. Instead, we develop the hierarchical approach described in the next subsection.

5.4 Hierarchical Optimization of Observations

We develop a hierarchical optimization approach to solve the optimal observation model described in subsection 5.3. The key insight of this hierarchical approach is that the subparts of the model in subsection 5.3 are loosely coupled only by the selected observations. We take a greedy approach and fix $y_{\max}$ in (39) to be 1, selecting the single best observation at a time and applying this approach iteratively to select a sequence of observations. Thus, we factor the subparts and hierarchically solve four optimization problems: (1) a lower bound policy value, (2) an upper bound policy value, (3) an upper bound on policy loss, and (4) the optimal observation minimizing worst-case policy loss.

The hierarchical optimization approach is shown in algorithm 2. Function `max-`Δ finds an upper bound on policy loss after observing probability interval i

(line 1). We solve for the upper bound on policy loss as a nonlinear program over the single decision variable z, representing the observed probability that maximizes policy loss. The objective of this nonlinear program separately finds the lower bound (line 3) and upper bound (line 4) to evaluate policy loss. We find worst-case policy loss as a nonlinear program (line 6). We enumerate worst-case policy losses over possible observations (line 8), and select the best observation (line 10).

Algorithm 2: Hierarchical Optimization of Observation

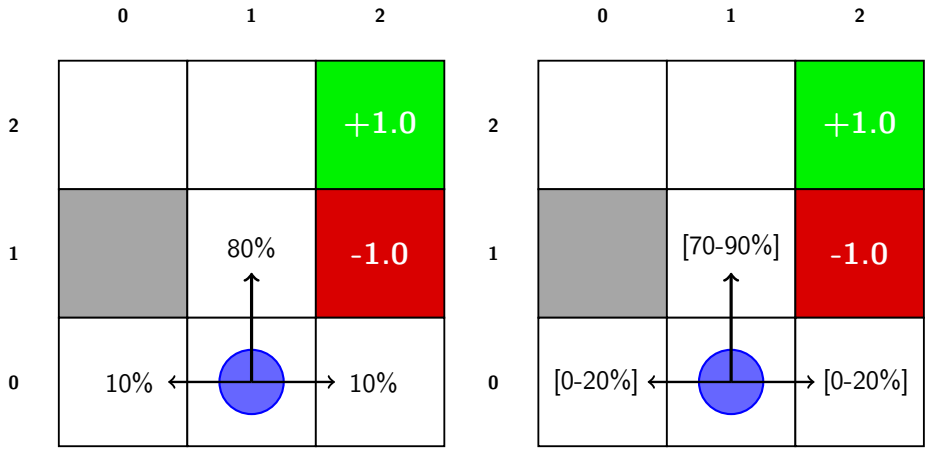
```
   Input: (S, A, [p], r);                                   /* Interval MDP */
   Output: i*;                                  /* The optimal observation */
1  function max-Δ(i) is // worst-case policy loss after observation i
2      function obj(z) is // policy when observation i is z
3          LB ← min_{π∈Π*_pot} v̲(π) s.t. (p_i = z);
4          UB ← max_{π∈Π*_pot} v̄(π) s.t. (p̂_i = z);
5          return UB − LB;
6      return max_z obj(z) s.t. p̲_i ≤ z ≤ p̄_i; // Solve the nonlinear program
   /* Enumerate possible observations to minimize loss                  */
7  (i*, Δ*) ← (1, max-Δ(1));
8  for i = 2, ..., n do
9      Δ ← max-Δ(i);
10     if Δ ≤ Δ* then (i*, Δ*) ← (i, Δ);
```

The hierarchical approach of algorithm 2 is independent of the particular method we use to find lower and upper bounds (line 3 and line 4). We select different bounds computation approaches to trade-off precision and efficiency. In our experiments, we compute lower bounds using the BMDP optimization model of subsection 5.2 (tighter/slower) and the ILP of subsection 5.1 (looser/faster), and we compute upper bounds using the BMDP optimization model of subsection 5.2 (tighter/slower) and the modified geometric series of algorithm 1 (looser/faster).

6 Experiments

We evaluate our approach for observing parameters to reduce epistemic uncertainty in Bounded-parameter MDPs. We first define the specific algorithmic variants used in our evaluation, distinguishing between the precision of the bound computation and the strategy used for observation selection.

Recall that our approach for computing the next observation (algorithm 2) can use different computation bounds. We refer to the tight bounds derived from the exact optimization mode of subsection 5.2 as Opt-Bounds. Conversely, we refer to the efficient approximations where we combine the Interval Linear Program (subsection 5.1) for lower bounds and the reachable set geometric series



(a) $\mathcal{M}_{\text{True}}$ (unknown to agent) (b) Bounded-parameter MDP

Fig. 1: An example problem instance on a 3×3 grid environment. The agent is the blue circle. The gray cell is an obstacle that blocks traversal. White cells are free to traverse. Cells with an incoming reward of $(+1.0)$ at coordinates (2,2) and a reward of (-1.0) at coordinates (1,2) are respectively the positive and negative terminate states. The transition probabilities are exact numbers for the underlying true MDP $(80\%, 10\%)$ and are intervals for the Bounded-parameter MDP $([70-90\%], [0-20\%])$. A small negative reward for not reaching terminal states yet is -0.01 to incentivize getting there faster. A discount factor $\gamma = 0.9$ is given to prioritize immediate rewards.

(algorithm 1) for upper bounds as Est-Bounds. In order to compute an optimal policy corresponding to the true underlying MDP $\mathcal{M}_{\text{True}}$, a series of observations need to be performed. As discussed in subsection 5.4, we suggest to greedily choose the next observation computed using algorithm 2 until convergence. We call such an approach Greedy(X) with $X \in \{\text{Opt-Bounds}, \text{Est-Bounds}\}$. An alternative approach we use as a baseline and refer to as Random(X) would be to iteratively sample a random observation, using the same bounds as the corresponding Greedy(X) for comparison purposes.

We now continue to describe the experimental setup (subsection 6.1) followed by a comparison between the computational cost of Opt-Bounds and Est-Bounds when used in algorithm 2 (subsection 6.2). We conclude (subsection 6.3) with a comparison between our greedy approach and the baseline.

6.1 Experimental Setup

Our experiments consist of scenarios involving an agent moving on a grid, a common example MDP [31]. We consider a $k \times k$ grid environments for $k \in \{3, 5, 7\}$ (see Figure 1 for a visualization of a 3×3 environments). Each environment consists of obstacles that cannot be traversed, a positive and negative terminal state where high and low rewards are obtained, respectively. In addition each

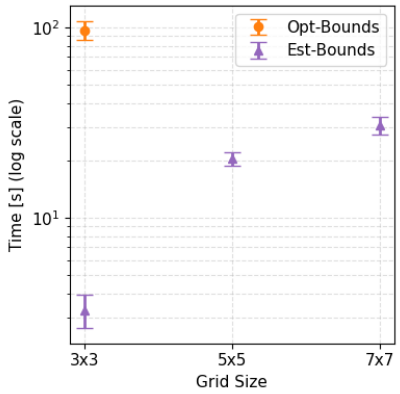


Fig. 2: Computation time to select a single observation (algorithm 2) using Opt-Bounds and Est-Bounds for varying grid sizes. We average running times over 30 trials for each point. Errors bars indicate samples standard deviation.

agent transition incurs a low negative reward motivating the agent to reach a terminal state earlier rather than later. At each state, the agent's actions include moving along one of the four cardinal directions. Each environment is associated with a true MDP $\mathcal{M}_{\text{True}}$ (unknown to the agent) as well as a Bounded-parameter MDP that contains $\mathcal{M}_{\text{True}}$, as in Figure 1.

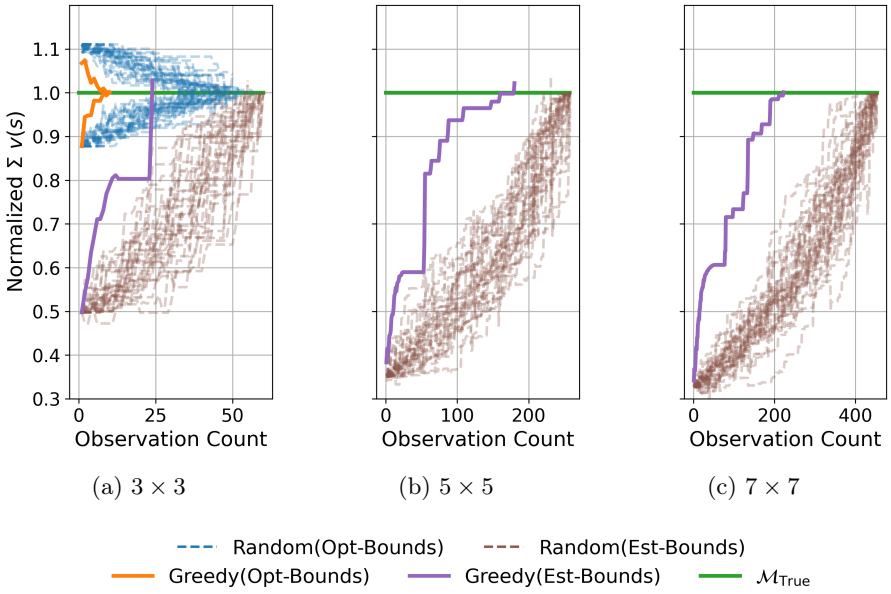
We run our tests on an Intel(R) Core(TM) i5-10310U CPU @ 1.70GHz running Debian 12. We solve constraints and nonlinear programs using Gurobi [16].

6.2 Computational cost of choosing the next observation.

We start by comparing the computational cost of Opt-Bounds and Est-Bounds by computing the first (i.e., most rewarding) observation on a bounded parameter MDP. We perform 30 such trials for each environment size and plot the running time as a function of grid size. Results, depicted in Figure 2 show that at 3×3 grid size, Est-Bounds achieves a $\times 29.5$ reduction in average run time compared to Opt-Bounds. At grid sizes larger than 3×3, Opt-Bounds timed out with 20 minutes allowed per policy loss calculation for an observation choice. Est-Bounds scaled better to 5×5 and 7×7 grids at $\times 6.2$ and $\times 9.3$ longer average run time than 3×3, respectively. Est-Bounds demands significantly less computational cost than Opt-Bounds and directly affects tractability for grid worlds larger than 3×3.

6.3 Convergence to optimal state values.

We continue to compare the convergence rate of Greedy(X) with the baseline Random(X). We consider how fast the UB $\sum v_{\text{UB}}(s)$ and/or LB $\sum v_{\text{LB}}(s)$ converge to the corresponding value of the true MDP $\mathcal{M}_{\text{True}}$ as a function of consecutive observations count. Because the particular values of $\sum v(s)$ do not matter as much as the convergence pattern in these plots, we normalize $\sum v_{\text{UB}}(s)$ values by



(a) 3×3 (b) 5×5 (c) 7×7

Fig. 3: Convergence of the normalized state value bounds on respective grid sizes: (a) 3×3 (b) 5×5 (c) 7×7. For Est-Bounds, the UB is still helpful in optimization but much looser than the LB, and we use only the LB for a clearer picture on convergence. The Est-Bounds LB may overshoot the true MDP value because transition probabilities used in Est-Bounds LB all assume the lowest value in their respective intervals, reducing the effect of negative living rewards in the expected total rewards.

that of $\mathcal{M}_{\text{True}}$. As mentioned, Opt-Bounds did not scale well to 5×5 and 7×7 grid sizes, hence only the 3×3 case in Fig. 3a has both Opt-Bounds and Est-Bounds results. With Est-Bounds, the UB is a loose estimate that is helpful in optimization but does not fully converge to the true value. We leave out UB values and only show LB values for Est-Bounds to avoid compressing the scale of curves and affecting legibility.

For the 3×3 case in Fig. 3a, Greedy(Opt-Bounds) converges to the optimal state values roughly between $\times 3$ and $\times 5$ faster than 30 instances of baseline Random(Opt-Bounds). Also for the 3×3 case in Fig. 3a, Greedy(Est-Bounds) shows that using estimated bounds converges roughly twice as slow as using exact bounds but still takes roughly half the observation count of the baseline Random(Est-Bounds) and still faster than Random(Opt-Bounds). With a similar trend, Greedy(Est-Bounds) converges faster than Random(Est-Bounds) about $\times 1.5$ for the 5×5 case in Fig. 3b and about $\times 2$ for the 7×7 case in Fig. 3c.

Note that Est-Bounds LB values may overshoot the true MDP $\mathcal{M}_{\text{True}}$ value. Est-Bounds LB transition probabilities assume the lowest value in their respective intervals to achieve the desired coefficient bounds in Eq. (10). The low probabilities reduce the effect of negative living penalties in the expected total rewards.

7 Conclusion, Discussion, and Future Work

We introduced an approach to action under epistemic uncertainty that not only recovers a safe policy but also identifies which uncertain model parameters to observe for reduced uncertainty in the utility of the chosen policy. By fusing environmental uncertainty (BMDP parameters) and computational uncertainty (optimality gaps) into a single policy loss metric, we allow an agent to reason about the value of refining its own knowledge.

Limitations and Future Work. While our results demonstrate the efficacy of the proposed hierarchical optimization, several limitations remain. Our *greedy* observation strategy (Alg. 2) is myopic; it selects the single most informative observation at each step. This significantly reduces computational complexity but may fail to identify sequences of observations that are only valuable when taken together. Investigating non-myopic lookahead strategies for observation selection could yield significant performance gains in complex environments where information is sparse or highly correlated.

Our exact method struggles to *scale* beyond small grid sizes in experiments, warranting the use of looser estimates for larger state spaces. Future work will explore more aggressive pruning techniques, the value iteration approach as in [15], or learning-based heuristics, to approximate utility bounds more efficiently.

As we transition this framework to embodied robot systems, observing a parameter is no longer a free query; it is a physical action—such as deploying a drone or navigating to a tactile inspection site—that incurs energy and time *costs*. We envision incorporating such costs into future versions of the objective function. Consequently, the agent must weigh the cost of acquisition against the value of the information. In this context, our work serves as an agent-centric dual to frameworks like Value of Assistance [1, 27].

Our current formulation assumes that an observation reveals the true parameter value perfectly, while physical *sensors are inherently noisy*. A realistic extension of this work would require us to treat observations as narrowing the parameter interval $[\underline{p}, \overline{p}]$ rather than collapsing it to a point value. With the same general approach, we can measure the impact of such partial interval reduction on utility bounds and policy loss.

Besides, real robots typically have *continuous state spaces*. Instead of discretization of the state space, which has scaling issues, we will investigate basing initial guesses of transition probabilities on the Euclidean distance between continuous states, with preset uncertainty intervals that can be refined and memoized after observations.

Acknowledgments. This study was supported in part by the ARL DCIST CRA [W911NF-17-2-0181] and ARL TBAM-CRP [W911NF-22-2-0235]. Many thanks to Dr. Alexandra Newman of the Department of Operations Research with Engineering, Colorado Schoool of Mines, Colorado, USA, for guidance on optimization efforts.

References

1. Amuzig, A., Dovrat, D., Keren, S.: Value of assistance for mobile agents. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1594–1600 (2023)
2. Avrachenkov, K., Filar, J.A., Gaitsgory, V., Stillman, A.: Singularly perturbed linear programs and markov decision processes. Operations Research Letters **44**(3), 297–301 (2016)
3. Badings, T., Romao, L., Abate, A., Jansen, N.: Probabilities are not enough: Formal controller synthesis for stochastic dynamical models with epistemic uncertainty. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) **37**(12), 14701–14710 (2023)
4. Badings, T., Simao, T.D., Suilen, M., Jansen, N.: Decision-making under uncertainty: beyond probabilities: Challenges and perspectives. International Journal on Software Tools for Technology Transfer **25**(3), 375–391 (2023)
5. Bertsimas, D., Tsitsiklis, J.N.: Introduction to linear optimization, vol. 6. Athena Scientific Belmont, MA (1997)
6. Bramblett, L., Bezzo, N.: Epistemic planning for heterogeneous robotic systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 691–698 (2023)
7. Church, A.: An unsolvable problem of elementary number theory. American Journal of Mathematics **58**(2), 345–363 (1936)
8. De Ghellinck, G.: Les problemes de decisions sequentielles. Cahiers du Centre d'Etudes de Recherche Opérationnelle **2**(2), 161–179 (1960)
9. Dearden, R., Friedman, N., Andre, D.: Model-based bayesian exploration. arXiv preprint arXiv:1301.6690 (2013)
10. Delage, E., Mannor, S.: Percentile optimization for markov decision processes with parameter uncertainty. Operations research **58**(1), 203–213 (2010)
11. d'Epenoux, F.: Sur un probleme de production et de stockage dans l'aléatoire. Revue Française de Recherche Opérationelle **14**(3-16), 4 (1960)
12. Der Kiureghian, A., Ditlevsen, O.: Aleatory or epistemic? does it matter? Structural Safety **31**(2), 105–112 (2009)
13. Epstein, L.G., Schneider, M.: Recursive multiple-priors. Journal of Economic Theory **113**(1), 1–31 (2003)
14. Ghavamzadeh, M., Mannor, S., Pineau, J., Tamar, A.: Bayesian reinforcement learning: A survey. Foundations and Trends® in Machine Learning **8**(5-6), 359–483 (2015)
15. Givan, R., Leach, S., Dean, T.: Bounded-parameter markov decision processes. Artificial Intelligence **122**(1-2), 71–109 (2000)
16. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2026), https://www.gurobi.com
17. Helton, J.C., Johnson, J.D., Oberkampf, W.L., Sallaberry, C.J.: Representation of analysis results involving aleatory and epistemic uncertainty. International Journal of General Systems **39**(6), 605–646 (2010)
18. Hladık, M.: Interval linear programming: A survey. Linear programming-new frontiers in theory and applications pp. 85–120 (2012)
19. Huangfu, Q., Hall, J.J.: Parallelizing the dual revised simplex method. Mathematical Programming Computation **10**(1), 119–142 (2018)
20. IBM Corp: IBM ILOG CPLEX Optimization Studio V22.1.1 User's Manual. International Business Machines Corporation, Armonk, NY (2026), https://www.ibm.com/products/ilog-cplex-optimization-studio

21. Iyengar, G.N.: Robust dynamic programming. Mathematics of Operations Research **30**(2), 257–280 (2005)
22. Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of computer computations, pp. 85–103. Springer (1972)
23. Li, S., Stouraitis, T., Gienger, M., Vijayakumar, S., Shah, J.A.: Set-based state estimation with probabilistic consistency guarantee under epistemic uncertainty. IEEE Robotics and Automation Letters **7**(3), 5958–5965 (2022)
24. Luenberger, D.G., Ye, Y.: Linear and nonlinear programming, vol. 2. Springer (1984)
25. Manne, A.S.: Linear programming and sequential decisions. Management Science **6**(3), 259–267 (1960)
26. Marques, L., Berenson, D.: Quantifying aleatoric and epistemic dynamics uncertainty via local conformal calibration (2024)
27. Masarwy, M., Goshen, Y., Dovrat, D., Keren, S.: Value of assistance for grasping. CoRR **abs/2310.14402** (2023)
28. Meggendorfer, T., Weininger, M., Wienhöft, P.: Solving robust markov decision processes: Generic, reliable, efficient. In: AAAI Conference on Artificial Intelligence (AAAI). vol. 39, pp. 26631–26641 (2025)
29. Nagami, K., Schwager, M.: State estimation and belief space planning under epistemic uncertainty for learning-based perception systems. IEEE Robotics and Automation Letters **9**(6), 5118–5125 (2024)
30. Puterman, M.L.: Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons (2014)
31. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson, 4th edn. (2020)
32. Strens, M.: A bayesian framework for reinforcement learning. In: ICML. vol. 2000, pp. 943–950 (2000)
33. Suilen, M., Badings, T., Bovy, E.M., Parker, D., Jansen, N.: Robust Markov Decision Processes: A Place Where AI and Formal Methods Meet, pp. 126–154. Springer Nature Switzerland (2025)
34. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction, vol. 1. MIT Press Cambridge (1998)
35. Thrun, S.: Probabilistic robotics. Communications of the ACM **45**(3), 52–57 (2002)
36. Turing, A.M., et al.: On computable numbers, with an application to the entscheidungsproblem. J. of Math **58**(345-363), 5 (1936)
37. Vajda, S., Evans, G.W.: An introduction to linear programming and the theory of games (1961)
38. Winston, W.L.: Operations research: applications and algorithm. Thomson Learning, Inc. (2004)