Bayesian Risk Markov Decision Processes

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider finite-horizon Markov Decision Processes where distributional parameters, such as transition probabilities, are unknown and estimated from data. The popular distributionally robust approach to addressing the parameter uncertainty can sometimes be overly conservative. In this paper, we propose a new formulation, Bayesian risk Markov decision process (BR-MDP), to address parameter uncertainty in MDPs, where a risk functional is applied in nested form to the expected total cost with respect to the Bayesian posterior distributions of the unknown parameters. The proposed formulation provides more flexible risk attitudes towards parameter uncertainty and takes into account the availability of data in future time stages. To solve the proposed formulation with the conditional value-at-risk (CVaR) risk functional, we propose an efficient approximation algorithm by deriving an analytical approximation of the value function and utilizing the convexity of CVaR. We demonstrate the empirical performance of the BR-MDP formulation and the proposed algorithms on a gambler's betting problem and an inventory control problem.

1 Introduction

Markov decision process (MDP) is a paradigm for modeling sequential decision making under uncertainty. From a modeling perspective, some parameters of MDPs are unknown and need to be estimated from data. In this paper, we consider MDPs where transition probability and cost parameters are not known. A natural question would be: given a finite and probably small set of data, how does a decision maker find an "optimal" policy that minimizes the expected total cost under the uncertain transition probability and cost parameters?

A possible approach that mitigates the parameter uncertainty lies in the framework of robust MDPs (e.g. [1, 2, 3, 4, 5]). In robust MDPs, parameters are usually assumed to belong to a known set referred to as the ambiguity set, and the optimal decisions are chosen according to their performance under the worst possible parameter realizations within the ambiguity set. Later [6] extends the distributionally robust approach to MDPs (DR-MDPs) with parameter uncertainty and utilizes the probabilistic information of the unknown parameters. The distributionally robust approach originates from stochastic optimization (e.g. [7], [8]), which regards the unknown parameters as random variables and assumes the associated distributions belong to an ambiguity set that is constructed from the data. DR-MDP then finds the optimal policy that minimizes the expected total cost with the parameters following the most adversarial (worst-case) distribution within the ambiguity set.

However, distributionally robust approaches might yield overly conservative solutions that perform poorly for scenarios that are more likely to happen than the worst case. In view of such drawback, [9, 10] propose a Bayesian risk optimization (BRO) framework that reformulates a stochastic optimization

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

problem with parameter uncertainty. They quantify the parameter uncertainty using a Bayesian posterior distribution as opposed to an ambiguity set, and impose a risk functional on the objective function with respect to the posterior distribution. The risk functionals provide more flexible attitudes towards risk than the conservative worst-case measure. BRO is essentially seeking a trade-off between the posterior expected performance and the robustness in the actual performance, which is measured by the width of confidence intervals for the solution's actual performance. Intuitively speaking, BRO framework tries to avoid a scenario where a solution performs well under the estimated model but performs badly under the true model, by possibly giving up some good expected performance and trading for more confidence about the actual performance of a solution (in terms of width of confidence interval). Apart from the overly conservative concern, as pointed out in [11], DR-MDP does not explicitly specify the dynamics of the considered problem, in the sense that the distribution of the unknown parameters does not depend on realizations of the data process.

In this paper, we take a similar perspective as BRO in [9, 10] and propose a new Bayesian risk formulation for MDPs (BR-MDPs) to address the parameter uncertainty in MDPs. To clearly specify the dynamics and the randomness in the system, we take the perspective of stochastic optimal control to model the state transition by a difference equation that involves current state, next state, action, and the randomness in the system (see Chapter 3.5 in [12]). We assume the distribution of the randomness belongs to some parametric family, and model the uncertainty over the unknown parameters via a Bayesian posterior distribution that is updated based on the realization of the randomness at every time stage. It is worth noting that applying the Bayesian approach to MDPs has been considered in [13], which proposes a Bayes-adaptive MDP (BAMDP) formulation with an augmented state composed of the underlying MDP state and the posterior distribution of the unknown parameters. In BAMDP, each transition probability is treated as an unknown parameter associated with a Dirichlet prior distribution, and an expectation is taken with respect to the Dirichlet posterior on the expected total cost. In contrast, our BR-MDP formulation imposes a risk functional, taken with respect to the posterior distribution (which could be chosen as Dirichlet distribution but is more general), on the expected total cost in a nested form.

On a related note, risk-averse decision making has been widely studied in MDPs. Apart from the robust MDPs that address the parameter uncertainty (also termed as epistemic uncertainty), risk-sensitive MDPs (e.g. [14, 15, 16, 17]) address the intrinsic uncertainty (also termed as aleatoric uncertainty) that is due to the inherent stochasticity of the underlying MDP, by replacing the risk-neutral expectation (with respect to the state transition) by general risk measures, such as conditional value-at-risk (CVaR, see [18]). Most of the existing literature on risk-sensitive MDPs consider a static risk functional applied to the total return (e.g. [19, 20, 21, 22]). There are two recent works closely related to ours, both of which apply a risk functional to BAMDP. Specifically, [23] formulates the risk-sensitive planning problem as a two-player zero-sum game and then applies a (static) risk functional to the expected total cost, which adds robustness to the incorrect priors over model parameters; [24] optimizes a CVaR risk functional over the total cost and simultaneously addresses both epistemic and aleatoric uncertainty. In contrast, we consider a nested risk functional and seek robustness with respect to the epistemic uncertainty. A primary motivation for considering such nested risk functional is the issue of time consistency (see [25, 26, 27, 11]) in the sense that the optimal solution to the formulated problem keeps optimal at every stage of the decision process with respect to the conditional risk functional. Optimizing a static risk measure can lead to "time-inconsistent" behavior, where the optimal policy at the current time stage can become suboptimal in the next time stage simply because a new piece of information is revealed (see [25, 26]). In this work, we show the proposed BR-MDP formulation is time consistent, and derive the corresponding dynamic programming equation with an augmented state that incorporates the posterior information. The proposed framework works for an offline planning problem, where the decision maker, after loaded with the learned optimal policy and put to the real environment, acts like it adapts to the environment. Also note that our problem setting relies on partial knowledge of the model (state transition equation, cost function etc.) and works in an offline planning setting with no interaction with the environment, and hence is different from Bayesian reinforcement learning (e.g. [28, 29]).

To solve the proposed BR-MDP formulation, we develop an efficient algorithm by drawing a similarity between BR-MDP and partially observable MDP (POMDP) and utilizing the convexity of CVaR. In a POMDP, the optimal value function (in a minimization problem) can be expressed as an lower envelope of a set of linear functions (also called α-functions, see [30]). We show a similar α-function representation of the value function for BR-MDP, where the number of α-functions grow exponentially over time. To have computationally feasible algorithm, we derive an analytical approximation of the value function that keeps a constant number of α-functions over time. Note that [31] also reformulates the BAMDP as a POMDP and proposes a point-based value iteration algorithm. However, their approach is not applicable for the risk-averse case, as it relies on the piecewise linearity of the optimal value function, which does not hold for risk functionals other than expectation. Additionally, [24] proposes an approximate algorithm based on Monte Carlo tree search and Bayesian optimization to solve risk-averse BAMDP, but it works only for a static risk functional and does not easily generalize to our formulation with a nested risk functional.

To summarize, the contributions of this paper are two folds. First, we propose a Bayesian risk MDP formulation to handle the parameter uncertainty in MDPs. Second, we propose an efficient algorithm to solve the proposed formulation with a CVaR risk functional, and the algorithm can be easily extended to other coherent risk measures (see [32] for an overview on coherent risk measures).

2 Preliminaries and problem formulation

2.1 Preliminaries: Bayesian risk optimization and CVaR

Bayesian risk optimization (BRO, see [9, 10]) considers a general stochastic optimization problem: $\min_x \mathbb{E}_{\mathbb{P}^c}[h(x,\xi)]$, where x is the decision vector, ξ is a random vector with distribution $\mathbb{P}^c$, h is a deterministic cost function. The true distribution $\mathbb{P}^c$ is rarely known in practice and often needs to be estimated from data. It is very likely that the solution obtained from solving an estimated model performs badly under the true model. To avoid such a scenario, BRO seeks robustness in the actual performance by imposing a risk functional to the expected cost function and solving the following problem: $\min_x \rho_{\mathbb{P}_n}\{\mathbb{E}_{\mathbb{P}_\theta}[h(x,\xi)]\}$, where ρ is a risk functional, and $\mathbb{P}_n$ is the posterior distribution of θ after observing n data points. It is assumed that the unknown distribution $\mathbb{P}^c$ belongs to a parametric family $\{\mathbb{P}_\theta | \theta \in \Theta\}$, where Θ is the parameter space and $\theta^c \in \Theta$ is the true parameter. Through a Bayesian perspective, θ^c can be viewed as a realization of a random variable θ.

In particular, conditional value at risk (CVaR), a common coherent risk measure (see [32]), is considered for the risk functional. For a random variable X defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, $\mathrm{VaR}^\alpha(X)$ is defined as the α-quantile of X, i.e., $\mathrm{VaR}^\alpha(X) := \inf\{t : \mathbb{P}(X \le t) \ge \alpha\}$, where the confidence level α takes value in $(0,1)$. Assuming there is no probability atom at $\mathrm{VaR}^\alpha(X)$, CVaR at confidence level α is defined as the mean of the α-tail distribution of Z, i.e., $\mathrm{CVaR}_\alpha(X) = \mathbb{E}\left[X \mid X \ge \mathrm{VaR}_\alpha(X)\right]$. It is shown in [18] that CVaR can be written as a convex optimization:

$$\mathrm{CVaR}_\alpha(X) = \min_{u\in\mathbb{R}} \left\{ u + \frac{1}{1-\alpha}\mathbb{E}\left[(X-u)^+\right] \right\}. \tag{1}$$

2.2 New formulation: Bayesian risk MDPs

Consider a finite-horizon MDP defined as $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{C})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the transition probability with $\mathcal{P}(s_{t+1}|s_t, a_t)$ denoting the probability of transitioning to state s_{t+1} from state s_t when action a_t is taken, $\mathcal{C}$ is the cost function with $\mathcal{C}_t(s_t, a_t, s_{t+1})$ denoting the cost at time stage t when action a_t is taken and state transitions from s_t to s_{t+1}. A Markovian deterministic policy π is a function mapping from $\mathcal{S}$ to $\mathcal{A}$. Given an initial state s_0, the goal is to find an optimal policy that minimizes the expected total cost: $\min_\pi \mathbb{E}^{\pi,\mathcal{P},\mathcal{C}}\left[\sum_{t=0}^{T-1} \mathcal{C}_t\left(s_t, a_t, s_{t+1}\right)\right]$, where $\mathbb{E}^{\pi,\mathcal{P},\mathcal{C}}$ is the expectation with policy π when the transition probability is $\mathcal{P}$ and the cost is $\mathcal{C}$. In practice, $\mathcal{P}$ and $\mathcal{C}$ are often unknown and estimated from data.

To deal with the parameter uncertainty in MDPs, we propose a Bayesian risk MDP formulation that extends the BRO formulation in stochastic optimization to MDPs. We assume the state transition is specified by the state equation $s_{t+1} = g_t(s_t, a_t, \xi_t)$ with a known transition function g_t, which involves state $s_t \in \mathcal{S}$, action $a_t \in \mathcal{A}$, and the randomness $\xi_t \in \Xi$ in the system. We assume ξ_t at time stages $t = 0, \cdots, T-1$ to be independent and identically distributed. Note that the state equation together with the distribution of ξ uniquely determines the transition probability of the MDP, i.e., $\mathcal{P}(s_{t+1}|s_t, a_t) = \mathbb{P}(\Xi_t \in \{\xi_t \in \Xi : s_{t+1} = g_t(s_t, a_t, \xi_t)\})$. We refer the readers to Chapter 3.5 in [12] for the equivalence between stochastic optimal control and MDP formulation. We use the representation of state equation instead of transition probability in MDPs, for the purpose of decoupling the randomness and the policy, leading to a cleaner formulation in the nested form. We assume the distribution of ξ, denoted by $f(\cdot;\theta^c)$, belongs to a parametric family $\{f(\cdot;\theta)|\theta \in \Theta\}$, where Θ is the parameter space, and $\theta^c \in \Theta$ is the true but unknown parameter value. The parametric assumption implies that we have partial knowledge of the problem structure. For example, in inventory control problems, the customer demand is unknown but often assumed to follow a Poisson process (see [33]). The cost at time stage t is assumed to be a function of state s_t, action a_t, and randomness ξ_t, i.e., $\mathcal{C}_t(s_t, a_t, \xi_t)$.

Now given a data set, we can compute the prior distribution μ_0 over the support Θ, which captures the initial uncertainty in the parameter estimate. Then given an observed realization of the data process, we update the posterior distribution μ_t according to the Bayes' rule. Let the policy be a sequence of mappings from physical state s_t and posterior μ_t to the action space, i.e., $\pi = \{\pi_t | \pi_t : \mathcal{S} \times \mathcal{M}_t \to \mathcal{A}, t = 0, \cdots, T-1\}$, where $\mathcal{M}_t$ is the space of posterior distributions at time stage t. The details of the BR-MDP formulation is presented below.

$$\min_{\pi} \ \rho_{\mu_0}\mathbb{E}_{f(\cdot;\theta_0)}\left[\mathcal{C}_0(s_0, a_0, \xi_0) + \cdots + \rho_{\mu_{T-1}}\mathbb{E}_{f(\cdot;\theta_{T-1})}\left[\mathcal{C}_{T-1}(s_{T-1}, a_{T-1}, \xi_{T-1}) + \mathcal{C}_T(s_T)\right]\right] \tag{2}$$

$$s.t. \ \ s_{t+1} = g_t(s_t, a_t, \xi_t), \ \ t = 0, \cdots, T-2; \tag{3}$$

$$\mu_{t+1}(\theta) = \frac{\mu_t(\theta) f(\xi_t;\theta)}{\int_\Theta \mu_t(\theta) f(\xi_t;\theta)\, d\theta}, \ t = 0, \cdots, T-2. \tag{4}$$

The cost at the last time stage only depends on the state, which is denoted by $\mathcal{C}_T(s_T)$. Also note $a_t = \pi_t(s_t, \mu_t)$, $\mathbb{E}_{f(\cdot;\theta_t)}$ denotes the expectation with respect to $\xi_t \sim f(\cdot;\theta_t)$ conditional on θ_t, and ρ_{μ_t} denotes a risk functional taken with respect to $\theta_t \sim \mu_t$. Equation (3) is the transition of the physical state s_t, and without loss of generality we assume that the initial state s_0 takes a deterministic value. Equation (4) is the update of the posterior μ_t, given the initial prior μ_0 that is constructed from the historical data set.

2.3 Time consistency and dynamic programming

It is important to note that the BR-MDP formulation (2) takes a nested form of the risk functional to ensure time consistency of its optimal policy. Time consistency, simply put, means that the optimal policy solved at time stage 0 is still optimal for any remaining time stage $t \geq 1$ even when the realization of the randomness ξ_t is revealed up to that time stage. To illustrate, consider the simple case of a two-stage problem where the risk functional ρ is CVaR with confidence level α. Our BR-MDP solves a nested formulation $\min_{a_0,a_1} \rho_{\mu_0}\mathbb{E}_{f(\cdot;\theta_0)}[\mathcal{C}_0(s_0, a_0, \xi_0) + \rho_{\mu_1}\mathbb{E}_{f(\cdot;\theta_1)}[\mathcal{C}_1(s_1, a_1, \xi_1)]]$, while the non-nested counterpart solves $\min_{a_0,a_1} \rho_{\mu_0}\mathbb{E}[\mathcal{C}_0(s_0, a_0, \xi_0) + \mathcal{C}_1(s_1, a_1, \xi_1)]$, where the expectation is taken with respect to (w.r.t.) the joint distribution of ξ_0 and ξ_1, and the static risk functional is applied to the total cost. Then we have the following relation between the two formulations:

$$\begin{aligned}&\rho_{\mu_0}\mathbb{E}_{f(\cdot;\theta_0)}[\mathcal{C}_0(s_0, a_0, \xi_0) + \rho_{\mu_1}\mathbb{E}_{f(\cdot;\theta_1)}[\mathcal{C}_1(s_1, a_1, \xi_1)]]\\ &\geq \rho_{\mu_0}\mathbb{E}_{f(\cdot;\theta_0)}[\mathcal{C}_0(s_0, a_0, \xi_0) + \mathbb{E}_{\xi_1|\xi_0}\mathcal{C}_1(s_1, a_1, \xi_1)] = \rho_{\mu_0}\mathbb{E}[\mathcal{C}_0(s_0, a_0, \xi_0) + \mathcal{C}_1(s_1, a_1, \xi_1)]\end{aligned}$$

where the inequality is justified by CVaR being the right tail average of the distribution, and the equality follows from the tower property of conditional expectation.

Note that for the non-nested formulation, the derivation of the dynamic programming equation is based on the interchangeability principle and the decomposability property of the risk functional,

where such decomposability property holds only for expectation and max-type risk functionals (see [34]). On the other hand, for our nested formulation (2), the corresponding dynamic programming equation is easily obtained as follows.

$$V_t^*(s_t, \mu_t) = \min_{a_t \in \mathcal{A}} \ \rho_{\mu_t} \mathbb{E}_{f(\cdot;\theta_t)} \left[\mathcal{C}_t(s_t, a_t, \xi_t) + V_{t+1}^*(s_{t+1}, \mu_{t+1}) | s_t, \mu_t, a_t \right], \forall s_t, \mu_t, \tag{5}$$

where $s_{t+1} = g_t(s_t, a_t, \xi_t)$, $\mu_{t+1}(\theta) = \frac{\mu_t(\theta) f(\xi_t;\theta)}{\int_\Theta \mu_t(\theta) f(\xi_t;\theta) d\theta}$, for $t = 0, \cdots, T-1$. Therefore, our BR-MDP formulation provides a time-consistent risk-averse framework to deal with epistemic uncertainty in MDPs. The exact dynamic programming is summarized in Algorithm 1.

Algorithm 1: Exact dynamic programming for finite-horizon BR-MDPs.

input: finite horizon T, initial state s_0, prior distribution μ_0;
output: optimal value function $V_0^*(s_0, \mu_0)$ and corresponding optimal policy π^*;
set $V_T^*(s_T, \mu_T) = \mathcal{C}_T(s_T), \forall (s_T, \mu_T) \in \mathcal{S} \times \mathcal{M}_t$;
for $t \leftarrow T-1$ **to** *0* **do**
  for *each* $(s_t, \mu_t) \in \mathcal{S} \times \mathcal{M}_t$ **do**
    solve dynamic programming equation (5);
    set $\pi_t^*(s_t, \mu_t) := a_t^*$, where a_t^* attains $V_t^*(s_t, \mu_t)$;
  end
end

3 An analytical approximate solution to BR-MDPs

Although the exact dynamic programming works for a general risk functional, there are two challenges to carry it out. First, the expectation and the risk functional are generally impossible to compute analytically and estimation by (nested) Monte Carlo simulation can be computationally expensive. Second, the update of the posterior distribution μ_t does not have a closed form in general and often results in an infinite-dimensional posterior. We circumvent the latter difficulty by using conjugate families of distributions (see Chapter 5 in [35]), where the posterior distribution falls into the same parametric family as the prior distribution, and thus maintain the dimensionality of the posterior to be the finite (and often small) dimension of the parameter space of the conjugate distribution. However, the posterior parameters usually take continuous values. Therefore, BR-MDP with the augmented state (s, μ) is in fact a continuous-state MDP. The computational cost for solving a continuous-state MDP increases exponentially in dimension if we simply discretize the continuous state.

In view of the computational challenges in carrying out the exact dynamic programming, we derive an efficient approximation algorithm for the BR-MDP with the risk functional CVaR. The main idea is that for CVaR, once we know the variable u in (1), it is reduced to an expectation of a convex function. If there is a way to approximate the value function for a given u, we can utilize the convexity of CVaR and apply gradient descent to solve for u. Hence, we proceed by first deriving the approximate value function for a fixed u and time stage. Similar to POMDP, by viewing the unknown parameter θ as unobservable state and physical state s as observable state, we maintain a belief (i.e. posterior distribution) μ on θ. It is well known that the value function in a POMDP can be represented by a set of so-called α-functions. We first show an α-function representation of the value function in BR-MDP, and then derive the approximate value function based on this representation.

3.1 α-function representation

By the definition of CVaR (see (1)) with a confidence level α, we can rewrite the dynamic programming equation (5) for the BR-MDP as:

$$V_t^*(s_t, \mu_t) = \min_{\substack{a_t \in \mathcal{A} \\ u_t \in \mathbb{R}}} \left\{ u_t + \frac{1}{1-\alpha} \sum_{\theta \in \Theta} \mu_t(\theta) \left(\int_\Xi f(\xi;\theta) \left(\mathcal{C}_t(s_t, a_t, \xi) + V_{t+1}^*(s_{t+1}, \mu_t) \right) d\xi - u_t \right)^+ \right\} \tag{6}$$

where we assume that ξ takes continuous values and the parameter space Θ is a finite set. If ξ is discrete, we can replace the integral by summation. The assumption of a finite parameter space

is practical in many real-world problems. It also can be viewed as a discrete approximation of a continuous parameter set and the discretization can be of any precision. The next proposition shows that the optimal value function corresponds to the lower envelope of a set of α-functions.

Proposition 3.1 (α-function representation)**.** *The optimal value function in* (6) *can be represented by the lower envelope of a set of α-functions denoted by $\Gamma_t = \{\alpha_t\}_{a_t \in \mathcal{A}}$, i.e.,*

$$V_t^*(s_t, \mu_t) = \min_{\alpha_t \in \Gamma_t} \sum_{\theta \in \Theta} \alpha_t(s_t, \theta)\mu_t(\theta),$$

$$\alpha_t(s_t, \theta) = u_t + \tfrac{1}{1-\alpha}\left(\int_\Xi f(\xi;\theta)\left(\mathcal{C}_t(s_t, a_t, \xi) + \min_{\alpha_{t+1}} \textstyle\sum_\theta \alpha_{t+1}(s_{t+1}, \theta)\frac{\mu_t(\theta)f(\xi;\theta)}{\sum_\theta \mu_t(\theta)f(\xi;\theta)}\right) d\xi - u_t\right)^+.$$

The detailed proof of Proposition 3.1 can be found in the appendix. Note that there is a major distinction between the α-function representations of a POMDP and a BR-MDP: the optimal value function in the risk-neutral POMDP is piecewise linear and convex in the belief state (see [30]), whereas the optimal value function in BR-MDP is no longer piecewise linear in the posterior due to the $(\cdot)^+$ operator in CVaR. In addition, it is computationally impossible to obtain the α-functions except for the last time stage. Specifically, denote the cardinality of the α_{t+1} set as $|\Gamma_{t+1}|$. Since for each realization of ξ there are $|\Gamma_{t+1}|$ candidates for α_{t+1}, there are a total of $|\mathcal{A}||\Gamma_{t+1}|^{|\Xi|}$ candidates for α_t, let alone the optimization over u_t. When Ξ is uncountable (i.e., when ξ is continuous random variable), the set Γ_t is also uncountable. To deal with the difficulty in computing the α-functions in POMDPs, [36] proposes several approximation algorithms, and later [37] extends the analysis to a continuous-state optimal stopping problem by applying Jensen's inequality to the exact value iteration in different ways and obtains a more computationally efficient approximation to the optimal value function. Inspired by these works, we derive the approximation approach in the next section.

3.2 α-function approximation

In this section, we derive the α-function approximation for a fixed vector $(u_0, u_1, \cdots, u_{T-1})$. Without loss of generality, we assume the cost function at each time stage is non-negative (otherwise add a large constant to the cost at each time stage). For the ease of exposition, we rewrite (6) as $V_t^*(s_t, \mu_t) = \min_{a_t \in \mathcal{A}, u_t \in \mathbb{R}} Q_t^*(s_t, \mu_t, a_t, u_t)$, where $Q_t^*(s_t, \mu_t, a_t, u_t) = u_t + \frac{1}{1-\alpha}\sum_{\theta\in\Theta}\mu_t(\theta)\left(\int_\Xi f(\xi;\theta)\left(\mathcal{C}_t(s_t, a_t, \xi) + V_{t+1}^*(s_{t+1}, \mu_t)\right) d\xi - u_t\right)^+$. Let $V_t(s_t, \mu_t) := \min_{a_t} Q_t^*(s_t, \mu_t, a_t, u_t)$ be the "optimal" value function. Let $\underline{V}_t(s_t, \mu_t) := \min_{\underline{\alpha}_t \in \underline{\Gamma}_t} \sum_{\theta\in\Theta} \underline{\alpha}_t(s_t, \theta)\mu_t(\theta)$, where $\underline{\Gamma}_t = \{\underline{\alpha}_t\}_{a_t\in\mathcal{A}}$ and

$$\underline{\alpha}_t(s_t, \theta) = u_t + \frac{1}{1-\alpha}\int_\Xi \left(\mathcal{C}_t(s_t, a_t, \xi)f(\xi|\theta) - u_t + \min_{\alpha_{t+1}} \alpha_{t+1}(s_{t+1}, \theta)f(\xi;\theta)\right) d\xi.$$

$\underline{V}_t(s_t, \mu_t)$ serves as a lower bound for $V_t(s_t, \mu_t)$ (see Proposition 3.2), and is similar to the fast informed bound in POMDP (see [36]). Note that the set $\underline{\Gamma}_t$ has a constant cardinality of $|\mathcal{A}|$. However, it involves a minimum within an integral, which can be hard to compute numerically. Also, the lower bound is loose in the sense that it could be negative, while the true CVaR value is always non-negative (due to the non-negative cost). Next, let $\bar{V}_t(s_t, \mu_t) := \min_{\bar{\alpha}_t \in \bar{\Gamma}_t} \sum_{\theta\in\Theta} \bar{\alpha}_t(s_t, \theta)\mu_t(\theta)$, where $\bar{\Gamma}_t = \{\bar{\alpha}_t\}_{a_t\in\mathcal{A}}$ and

$$\bar{\alpha}_t(s_t, \theta) = u_t + \frac{1}{1-\alpha}\left(\int_\Xi \mathcal{C}_t(s_t, a_t, \xi)f(\xi;\theta)d\xi - u_t\right)^+ + \frac{1}{1-\alpha}\int_\Xi \alpha_{t+1}(s_{t+1}, \theta)f(\xi;\theta)d\xi.$$

$\bar{V}_t(s_t, \mu_t)$ serves as an upper bound for $V_t(s_t, \mu_t)$ (see Proposition 3.2), and is similar to the unobservable MDP bound in POMDP (see [36]), obtained by discarding all observations available to the decision maker. Suppose the cardinality at time stage $t+1$ is $|\bar{\Gamma}_{t+1}|$, then $\bar{\Gamma}_t$ has a cardinality of $|\mathcal{A}||\bar{\Gamma}_{t+1}|$. In the following, we derive another approximate value function $\widetilde{V}_t$ that is bounded by $\underline{V}_t$ and $\bar{V}_t$, and is at least better than one of the above bounds. It keeps a constant number $|\mathcal{A}|$ of α-functions at each time stage, thus drastically reducing the computational complexity. Let

$\widetilde{V}_t(s_t,\mu_t) := \min_{\widetilde{\alpha}_t \in \widetilde{\Gamma}_t} \sum_{\theta\in\Theta} \widetilde{\alpha}_t(s_t,\theta)\mu_t(\theta)$, where $\widetilde{\Gamma}_t = \{\widetilde{\alpha}_t\}_{a_t\in\mathcal{A}}$ and

$$\widetilde{\alpha}_t(s_t,\theta) = u_t + \frac{1}{1-\alpha}\left(\int_\Xi \mathcal{C}_t(s_t,a_t,\xi)f(\xi;\theta)d\xi - u_t + \min_{\alpha_{t+1}} \int_\Xi \alpha_{t+1}(s_{t+1},\theta)f(\xi;\theta)d\xi\right)^+. \tag{7}$$

Proposition 3.2. *For all $t < T$ and any given $u_t \in \mathbb{R}$, the following inequalities hold:*

$$\underline{V}_t(s_t,\mu_t) \le \widetilde{V}_t(s_t,\mu_t) \le \bar{V}_t(s_t,\mu_t), \quad \underline{V}_t(s_t,\mu_t) \le V_t(s_t,\mu_t) \le \bar{V}_t(s_t,\mu_t).$$

The detailed proof of Proposition 3.2 can be found in the appendix. To have an implementable algorithm, we need to use the approximate updating of α-functions iteratively and replace the true α-functions at the $t+1$ time stage in (7) by the approximate $\widetilde{\alpha}$-functions from the previous iteration. It is clear that the iterative approximations preserve the directions of the inequalities. Define $u_{t:} := (u_t, \cdots, u_{T-1})$. The approximate value function at time stage t for a given $u_{t:}$ is given by $\widetilde{V}_t(s_t,\mu_t,u_{t:}) = \min_{\tilde{\alpha}_t\in\tilde{\Gamma}_t} \sum_{\theta\in\Theta} \widetilde{\alpha}_t(s_t,\theta)\mu_t(\theta)$, where $\tilde{\Gamma}_t = \{\tilde{\alpha}_t\}_{a_t\in\mathcal{A}}$ and

$$\widetilde{\alpha}_t(s_t,\theta) = u_t + \frac{1}{1-\alpha}\left(\int_\Xi \mathcal{C}_t(s_t,a_t,\xi)f(\xi;\theta)d\xi - u_t + \min_{\tilde{\alpha}_{t+1}} \int_\Xi \widetilde{\alpha}_{t+1}(s_{t+1},\theta)f(\xi;\theta)d\xi\right)^+. \tag{8}$$

3.3 Approximate dynamic programming with gradient descent

In this section, we incorporate α-function approximation with gradient descent on $(u_0, u_1, \cdots, u_{T-1})$ based on the convexity of the approximate value function w.r.t. $(u_0, u_1, \cdots, u_{T-1})$, as formally shown in the theorem below.

Theorem 3.3. *Suppose the cost function $\mathcal{C}_t(s,a,\xi)$ is jointly convex in (s,a) for any fixed ξ, and the state transition function $g_t(s,a,\xi)$ is jointly convex in (s,a) for any fixed ξ. Then the approximate value function $\widetilde{V}_t(s_t,\mu_t,u_{t:})$ is convex in $u_{t:}$.*

The detailed proof of Theorem 3.3 can be found in the appendix. We give the full algorithm in Algorithm 2.

Algorithm 2: Approximate dynamic programming for finite-horizon CVaR BR-MDPs.

input: finite horizon T, initial state s_0, prior distribution μ_0, initial vector $u^0 = (u_0^0, u_1^0, \cdots, u_{T-1}^0)$, gradient descent step size η_k for $k = 0, 1, \cdots$;
initialization: set $\widetilde{\alpha}_T(s_T,\theta) = \mathcal{C}_T(s_T), \forall s_T \in \mathcal{S}, \forall \theta \in \Theta$; set $k = 0$.
while *some stopping criterion is met* **do**
  for $t \leftarrow T-1$ **to** *0* **do**
    for each action $a_t \in \mathcal{A}$, compute $\widetilde{\alpha}_t(s_t,\theta)$ according to (8);
  end
  approximate the value function $\widetilde{V}_0(s_0,\mu_0,u^k) := \min_{\widetilde{\alpha}_0} \sum_{\theta\in\Theta} \widetilde{\alpha}_0(s_0,\theta)\mu_0(\theta)$;
  compute the gradient $\frac{\partial \widetilde{V}_0}{\partial u^k}$, update the vector $u^{k+1} = u^k - \eta_k \frac{\partial \widetilde{V}_0}{\partial u^k}$, set $k = k+1$.
end
output the approximate value function $\widetilde{V}_0(s_0,\mu_0,u^k)$;
output the optimal policy $\widetilde{\pi}_t(s_t,\mu_t) := \arg\min_{a_t\in\mathcal{A}} \sum_{\theta\in\Theta} \widetilde{\alpha}_t(s_t,a_t,\theta)\mu_t(\theta)$.

Note that in Algorithm 2, when applying the gradient descent, we need to compute the gradient of the approximate value function w.r.t. the vector $(u_0, u_1, \cdots, u_{T-1})$. For $\frac{\partial \widetilde{V}_0}{\partial u_0}$, we have

$$\frac{\partial \widetilde{V}_0}{\partial u_0} = \sum_{\theta\in\Theta}\left(1 - \frac{1}{1-\alpha}\mathbb{1}\left\{\int_\Xi \mathcal{C}_0(s_0,\widetilde{a}_0^*,\xi)f(\xi;\theta)d\xi - u_0 + \min_{\tilde{\alpha}_1}\int_\Xi \widetilde{\alpha}_1(s_1,\theta)f(\xi;\theta)d\xi \ge 0\right\}\right)\mu_0(\theta),$$

where $\widetilde{a}_0^* = \arg\min_{a_0\in\mathcal{A}} \sum_{\theta\in\Theta} \widetilde{\alpha}_0(s_0,\theta)\mu_0(\theta)$. For $\frac{\partial \widetilde{V}_0}{\partial u_t}$, $t = 1, \cdots, T-1$, we have

$$\frac{\partial \widetilde{V}_0}{\partial u_t} = \sum_{\theta\in\Theta}\frac{1}{1-\alpha}\mathbb{1}\left\{\int_\Xi \mathcal{C}_0(s_0,\widetilde{a}_0^*,\xi)f(\xi;\theta)d\xi - u_0 + \min_{\tilde{\alpha}_1}\int_\Xi \widetilde{\alpha}_1(s_1,\theta)f(\xi;\theta)d\xi \ge 0\right\}\cdot\left[\int_\Xi \frac{\partial \widetilde{\alpha}_1}{\partial u_t}f(\xi;\theta)d\xi\right]\mu_0(\theta),$$

where $\frac{\partial \widetilde{\alpha}_l}{\partial u_t}$ can be computed recursively from $\frac{\partial \widetilde{\alpha}_{l+1}}{\partial u_t}$ for $l = 1, \cdots, t-1$.

The approximate value function output by Algorithm 2 provides an upper bound on the optimal value function, which is shown in the theorem below.

Theorem 3.4. $\min_{u_{t:}} \widetilde{V}_t(s_t, \mu_t, u_{t:})$ *is an upper bound for the optimal value function* $V_t^*(s_t, \mu_t)$.

The detailed proof of Theorem 3.4 can be found in the appendix. We will later show in the numerical experiments that even though the approximate value function is an upper bound on the optimal value function, the gap between these two is small. As a final note, even though we develop the algorithm for the risk functional CVaR, it can be extended easily to other coherent risk measures. More specifically, for a class of coherent risk measures which can be represented in the following parametric form $\mathcal{R}(Z) := \inf_{\lambda \in \Lambda} \mathbb{E}[\Psi(Z, \lambda)]$, where $\Psi : \mathbb{R} \times \Lambda \to \mathbb{R}$ is a real-valued function and $\Psi(z, \lambda)$ is convex in (z, λ), we can apply the same technique to approximate the α-functions for a given vector $\lambda_0, \cdots, \lambda_{T-1}$, and then apply gradient descent on the approximate value function. The convergence is guaranteed due to the convexity. We refer the readers to [38] for more discussion on this class of coherent risk measures.

4 Numerical experiments

Code in Python for the numerical experiments is included in the supplementary. Computational time is reported for a 1.4 GHz Intel Core i5 processor with 8 GB memory. We illustrate the performance of our proposed formulation and algorithms with two **offline planning** problems. (1) **Gambler's betting problem**. Consider a gambler betting in the casino with initial money of s_0. At each time stage the gambler chooses how much to bet from a set $\{0, 1, 2, 5, 10\}$. To ensure the cost for each time stage is non-negative, we add a constant $c = 10$ to each time stage, and the cost is given by $c - a \cdot \xi$, where a stands for action of how much to bet, $\xi = 1$ stands for a win and $\xi = -1$ stands for a loss, and the winning rate $\theta^c = \mathbb{P}(\xi = 1)$ is unknown. The gambler bets for $T = 6$ rounds. The data set consists of historical betting record with size 10. (2) **Inventory control problem**. Consider a warehouse manager with initial inventory level s_0. At each time stage the manager chooses how much to replenish from the set $\{0, 1, \cdots, M - s\}$, where M is the storage capacity. The customer demand is a random variable ξ, which is assumed to follow a Poisson distribution with parameter θ^c that is truncated below M_C, the maximal customer demand the warehouse can handle. The state transition is given by $s_{t+1} = (s_t + a_t - \xi_t)^+$, the cost function is given by $\mathcal{C}_t(s_t, a_t, \xi_t) = h_t \cdot (s_t + a_t - \xi_t)^+ + p_t \cdot (\xi_t - s_t - a_t)^+$, where h_t is the holding cost and p_t is the penalty cost. The final stage cost is set to 0 for simplicity. The manager has to plan for $T = 6$ time stages. The data set consists of historical customer demand with size 10.

We compare the following approaches. (1) CVaR BR-MDP (exact): exact dynamic programming (Algorithm 1) presented in this paper. (2) CVaR BR-MDP (approx): approximate dynamic programming (Algorithm 2) presented in this paper. (3) Nominal: maximal likelihood estimation (MLE) estimator θ_{MLE} is computed from the given data. The policy is obtained by solving an MDP with parameter θ_{MLE}. (4) DR-MDP: distributionally robust MDP presented in [6]. For each of the considered approaches, we obtain the corresponding optimal policy for a given data set, and then evaluate the actual performance of the obtained policy on the true system, i.e., MDP with the true distributional parameter θ^c. This is referred to as one replication, and we repeat the experiments for 100 replications. Results for the gambler's betting problem can be found in Table 1a, Figure 1 and Figure 1b. Results for the inventory control problem can be found in Table 1b.

Table 1 reports the average time to obtain the optimal policy and the average actual performance of the obtained policy over the 100 replications. We have the following observations: (1) The nominal approach has the largest variability (see standard deviation of the actual performance). (2) DR-MDP is the most conservative since the standard deviation is 0, but this conservativeness is not always desirable.In the betting problem, the conservative action is not to bet. DR-MDP will always choose not to bet, which is not optimal when $\theta^c > 0.5$. This can be seen from the worst actual performance of DR-MDP in the betting problem when $\theta^c = 0.55$ and in the inventory control problem when

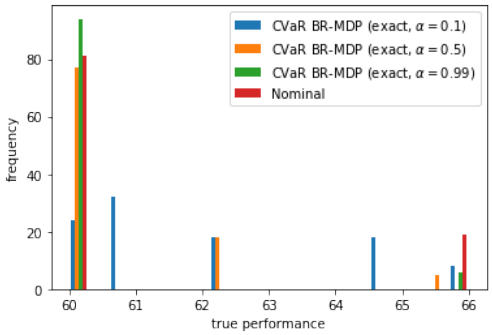
Table 1: Average time to solve different formulations per replication and average actual performances. Standard deviation is reported in the bracket. Experiments are run on 100 replications.

Approach	$\theta^c = 0.45$		$\theta^c = 0.55$
	Time (s)	performance	performance
CVaR BR-MDP (exact, $\alpha = 0.4$)	69.15(1.06)	60.42(1.38)	57.62(2.07)
CVaR BR-MDP (approx, $\alpha = 0.4$)	6.26(0.14)	60.97(1.67)	59.35(1.73)
Nominal	–	61.14(2.35)	57.06(3.00)
CVaR BR-MDP (exact, $\alpha = 1$)	66.78(3.13)	60.18(1.02)	59.64(1.42)
DR-MDP	–	60.00(0.00)	60.00(0.00)

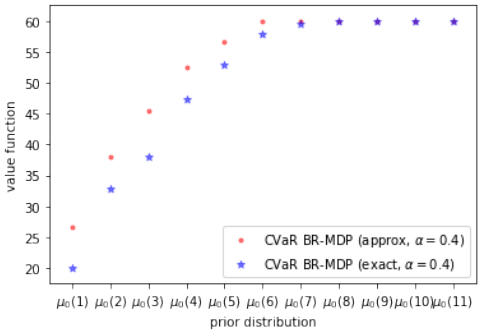
(a) Betting problem.

$\theta^c = 12$	
Time (s)	performance
2951.12(68.38)	81.63(2.27)
224.57(12.65)	83.55(3.58)
–	84.44(7.36)
2947.20(98.03)	83.25(1.86)
–	99.77(0.00)

(b) Inventory problem.



(a) Histogram of actual performance over 100 replications for CVaR BR-MDP (exact) with different α.



(b) Value functions of CVaR BR-MDP (exact and approx) under different priors.

Figure 1: Betting problem

$\theta^c = 12$. (3) For CVaR BR-MDP, the computational time for the approximate algorithm is greatly reduced compared to the exact algorithm. Figure 1a shows the histogram of actual performance over 100 replications for the nominal approach and CVaR BR-MDP (exact) with different confidence levels $\alpha = 0.1, 0.5, 0.99$ under distributional parameter $\theta^c = 0.45$. We have the following observations combining Table 1 and Figure 1a: (1) Our CVaR BR-MDP formulation (both exact and approximate) produces more consistent solution performance across a wide range of input data compared to the nominal approach, which can be seen from the smaller standard deviation in Table 1 and more concentrated distribution of the actual performance in Figure 1a. (2) Compared with DR-MDP, our BR-MDP formulation provides less conservative solution by not fixating on the worst-case scenario, which can be seen from the better actual performance in the inventory control problem (see Table 1b) and the betting problem (see Table 1a when $\theta^c = 0.55$). As such, our CVaR BR-MDP formulation balances the trade-off between the expected performance and robustness in the actual performance. (3) Figure 1a shows that in the betting problem with $\theta^c = 0.45$, the nominal approach has 20 replications where MLE estimator $\theta_{\text{MLE}} > 0.5$ and the gambler chooses to bet, which is not the optimal action. In contrast, CVaR BR-MDP formulation will learn from the future data realization and update its posterior distribution on θ. As a result, in those 20 replications, the gambler initially chooses to bet, but after some time chooses not to bet, which results in the left-shift of the actual performance distribution. This illustrates time consistency (or in other words, adaptivity to the data process) of our BR-MDP formulation. This illustration is even more evident by the comparison between DR-MDP and CVaR BR-MDP with $\alpha = 1$ (CVaR with $\alpha = 1$ corresponds to the worst-case measure), where the only difference is that BR-MDP takes a nested form of risk functional while DR-MDP uses a static one. (4) The confidence level α affects the conservativeness of the CVaR BR-MDP formulation. As α increases, the gambler is more likely to take a conservative action (which is not to bet), so the actual performance distribution will shift more to the left. Lastly, Figure 1b plots the value function $V_0^*(s_0, \mu_0)$ of CVaR BR-MDP (exact) and $\tilde{V}_0^*(s_0, \mu_0)$ of CVaR BR-MDP (approx) under different prior distributions μ_0 with $\theta^c = 0.45$, verifying Theorem 3.4 that $\tilde{V}_0^*(s_0, \mu_0)$ is indeed an upper bound for $V_0^*(s_0, \mu_0)$ but the difference between these two is small.

References

[1] Arnab Nilim and Laurent Ghaoui. Robustness in markov decision problems with uncertain transition matrices. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2004.

[2] Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

[3] Erick Delage and Shie Mannor. Percentile optimization for markov decision processes with parameter uncertainty. *Operations research*, 58(1):203–213, 2010.

[4] Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

[5] Marek Petrik and Reazul Hasan Russel. Beyond confidence regions: Tight bayesian ambiguity sets for robust mdps. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[6] Huan Xu and Shie Mannor. Distributionally robust markov decision processes. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[7] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

[8] Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

[9] Enlu Zhou and Wei Xie. Simulation optimization when facing input uncertainty. In L. Yilmaz, W. K V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, editors, *Proceedings of the 2015 Winter Simulation Conference*, pages 3714–3724, Piscataway, New Jersey, 2015. Institute of Electrical and Electronics Engineers, Inc.

[10] Di Wu, Helin Zhu, and Enlu Zhou. A bayesian risk approach to data-driven stochastic optimization: Formulations and asymptotics. *SIAM Journal on Optimization*, 28(2):1588–1612, 2018.

[11] Alexander Shapiro. Tutorial on risk neutral, distributionally robust and risk averse multistage stochastic programming. *European Journal of Operational Research*, 288(1):1–13, 2021.

[12] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[13] Michael O'Gordon Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. University of Massachusetts Amherst, 2002.

[14] Ronald A Howard and James E Matheson. Risk-sensitive markov decision processes. *Management science*, 18(7):356–369, 1972.

[15] Andrzej Ruszczyński. Risk-averse dynamic programming for markov decision processes. *Mathematical programming*, 125(2):235–261, 2010.

[16] Marek Petrik and Dharmashankar Subramanian. An approximate solution method for large risk-averse markov decision processes. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, page 805–814, Arlington, Virginia, USA, 2012. AUAI Press. ISBN 9780974903989.

[17] Takayuki Osogami. Robustness and risk-sensitivity in markov decision processes. *Advances in Neural Information Processing Systems*, 25, 2012.

[18] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2: 21–41, 2000.

[19] Nicole Bäuerle and Jonathan Ott. Markov decision processes with average-value-at-risk criteria. *Mathematical Methods of Operations Research*, 74(3):361–379, 2011.

[20] Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for cvar optimization in mdps. *Advances in neural information processing systems*, 27, 2014.

[21] William B Haskell and Rahul Jain. A convex analytic approach to risk-aware markov decision processes. *SIAM Journal on Control and Optimization*, 53(3):1569–1598, 2015.

[22] Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a cvar optimization approach. *Advances in neural information processing systems*, 28, 2015.

[23] Apoorva Sharma, James Harrison, Matthew Tsao, and Marco Pavone. Robust and adaptive planning under model uncertainty. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 410–418, 2019.

[24] Marc Rigter, Bruno Lacerda, and Nick Hawes. Risk-averse bayes-adaptive reinforcement learning. *arXiv preprint arXiv:2102.05762*, 2021.

[25] Takayuki Osogami and Tetsuro Morimura. Time-consistency of optimization problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 1945–1953, 2012.

[26] Dan A Iancu, Marek Petrik, and Dharmashankar Subramanian. Tight approximations of dynamic risk measures. *Mathematics of Operations Research*, 40(3):655–682, 2015.

[27] Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. *Advances in neural information processing systems*, 28, 2015.

[28] Mahdi Imani, Seyede Fatemeh Ghoreishi, and Ulisses M. Braga-Neto. Bayesian control of large mdps with unknown dynamics in data-poor environments. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[29] Esther Derman, Daniel Mankowitz, Timothy Mann, and Shie Mannor. A bayesian approach to robust reinforcement learning. In *Uncertainty in Artificial Intelligence*, pages 648–658. PMLR, 2020.

[30] Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

[31] Pascal Poupart, Nikos Vlassis, Jesse Hoey, and Kevin Regan. An analytic solution to discrete bayesian reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 697–704, New York, NY, USA, 2006. Association for Computing Machinery.

[32] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

[33] Guillermo Gallego and Garrett Van Ryzin. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management science*, 40(8):999–1020, 1994.

[34] Alexander Shapiro. Interchangeability principle and dynamic equations in risk averse stochastic programming. *Operations Research Letters*, 45(4):377–381, 2017.

[35] Robert Schlaifer and Howard Raiffa. *Applied statistical decision theory*. Division of Research, Harvard Business School, Boston, Massachusetts, 1961.

[36] Milos Hauskrecht. Value-function approximations for partially observable markov decision processes. *Journal of artificial intelligence research*, 13:33–94, 2000.

[37] Enlu Zhou. Optimal stopping under partial observation: Near-value iteration. *IEEE Transactions on Automatic Control*, 58(2):500–506, 2013.

[38] Vincent Guigues, Alexander Shapiro, and Yi Cheng. Risk-averse stochastic optimal control: an efficiently computable statistical upper bound. *arXiv preprint arXiv:2112.09757*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] This work is mostly theoretical, and we do not foresee any negative societal impacts of our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]

(b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]