Planning for Risk-Aversion and Expected Value in MDPs

Marc Rigter, Paul Duckworth, Bruno Lacerda, Nick Hawes

Oxford Robotics Institute, University of Oxford, United Kingdom
{mrigter, pduckworth, bruno, nickh }@robots.ox.ac.uk

Abstract

Planning in Markov decision processes (MDPs) typically optimises the *expected* cost. However, optimising the expectation does not consider the risk that for any given run of the MDP, the total cost received may be unacceptably high. An alternative approach is to find a policy which optimises a risk-averse objective such as conditional value at risk (CVaR). However, optimising the CVaR alone may result in poor performance in expectation. In this work, we begin by showing that there can be multiple policies which obtain the optimal CVaR. This motivates us to propose a lexicographic approach which minimises the expected cost subject to the constraint that the CVaR of the total cost is optimal. We present an algorithm for this problem and evaluate our approach on four domains. Our results demonstrate that our lexicographic approach improves the expected cost compared to the state of the art algorithm, while achieving the optimal CVaR.

Introduction

Markov decision processes (MDPs) are a common framework for decision-making under uncertainty, and have been applied to many domains such as inventory control (Ahiska et al. 2013) and robot navigation (Lacerda et al. 2019). The solution for an MDP typically optimises the expected total return, defined as either a reward to maximise or a cost to minimise. In this work, we consider the cost minimisation setting (Fig 1a). For any single run of the MDP, the total cost received is uncertain due to the MDP's stochastic transitions. In some applications we wish to compute *risk-averse* policies which prioritise avoiding the worst outcomes, rather than simply minimising the expected total cost irrespective of the variability.

As an example, consider an inventory control problem where a decision-maker decides how much stock to purchase each day, while subject to uncertain demand from customers. The policy which optimises expected value purchases large quantities of stock to maximise the expected profit. However, if demand is much lower than expected, significant losses may be incurred. A risk-averse policy purchases less stock, and is therefore less profitable on average, but avoids the possibility of large losses.

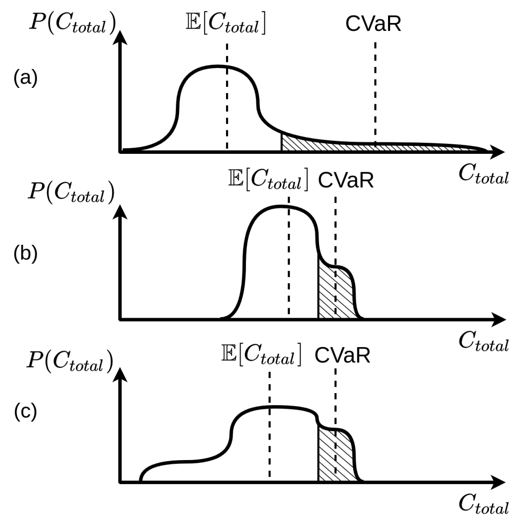

Figure 1: Illustration of distributions over the total cost, C_{total}, for different approaches: (a) expected value optimisation, (b) CVaR optimisation, and (c) our approach, which optimises the expected value subject to the constraint that CVaR is optimal. Shaded regions indicate the α-portion of the right tail of each distribution.

In this work, we focus on *conditional value at risk* (CVaR), a well known coherent risk metric (Rockafellar, Uryasev et al. 2000), in the *static* risk setting. In MDPs, the static CVaR at confidence level $\alpha \in (0, 1]$ corresponds to the mean total cost in the worst α-fraction of runs. Therefore, optimising the CVaR corresponds to optimising the α-portion of the right tail of the distribution over the total cost (Fig. 1b).

We propose that for risk-sensitive applications the *primary* objective should be to avoid the risk of a poor outcome (i.e. avoid significant losses). However, if the risk of a poor outcome has been avoided, then the *secondary* objective should be to optimise the expected value (i.e. maximise the expected profit). This motivates us to propose a lexicographic approach that optimises the expected total cost subject to the constraint that the CVaR of the total cost is optimal. During execution, the resulting policy is initially risk-averse. However, the policy may begin taking more aggressive actions to improve the expected cost, provided that there is no longer any risk of incurring a bad run which would influence the CVaR. Fig. 1c illustrates the cost distribution for our approach, compared to optimising the expected cost or the CVaR only. As indicated by the dashed lines, our approach obtains the same CVaR as optimising for CVaR alone (Fig. 1b), but the expected cost is improved.

Our main contributions are: 1) showing that there can be multiple policies that obtain the optimal CVaR in an MDP, 2) proposing and formalising the lexicographic problem of optimising expected value in MDPs subject to the constraint of minimising CVaR, and 3) an algorithm to solve this problem, based on reducing it to a two-stage optimisation in a stochastic game. To the best of our knowledge, this is the first work to address lexicographic optimisation of CVaR and expected value in sequential decision making problems.

We evaluate our algorithm on four domains, including a road network navigation domain which uses real data from traffic sensors (Chen et al. 2001) to simulate journey times. Our experimental results demonstrate that our approach significantly improves the expected cost on all domains while attaining the optimal conditional value at risk.

Related Work

Many existing works address risk-averse optimisation in MDPs. Early work considered the expected utility framework (Howard and Matheson 1972), where the cost is transformed according to a convex utility function to achieve risk-aversion. However, it is difficult to "shape" the cost function appropriately to achieve the desired behaviour (Majumdar and Pavone 2020). Other works consider risk metrics such as the mean-variance criterion (Sobel 1982), or the value at risk (Filar, Krass, and Ross 1995). However, these risk metrics are not *coherent*, meaning that they do not satisfy properties consistent with rational decision-making. See Artzner et al. (1999) or Majumdar and Pavone (2020) for an introduction to coherent risk metrics.

Examples of coherent risk metrics include the conditional value at risk (CVaR) (Rockafellar, Uryasev et al. 2000), the entropic value at risk (Ahmadi-Javid 2012), and the Wang risk measure (Wang 2000). In this work we focus on CVaR because it is intuitive to understand, and it is the prevailing risk metric used in risk-sensitive domains such as finance (Basel Committee on Banking Supervision 2014). We consider *static* risk, where the risk metric is applied to the total cost, rather than *dynamic* risk which penalises risk at each time step (Ruszczyński 2010). For the static CVaR setting, approaches based on value iteration over an augmented state space have been proposed (Bäuerle and Ott 2011; Chow et al. 2015). Other works propose policy gradient methods to find a locally optimal solution for CVaR (Borkar and Jain 2010; Chow and Ghavamzadeh 2014; Tamar, Glassner, and Mannor 2015; Tamar et al. 2015; Prashanth 2014; Chow et al. 2017; Tang, Zhang, and Salakhutdinov 2020), or general coherent risk metrics (Tamar et al. 2016). Keramati et al. (2020) present an RL algorithm based on optimism in the face of uncertainty, while Rigter, Lacerda, and Hawes (2021) propose an approach based on Monte Carlo tree search and also consider model uncertainty.

Existing works have considered trading off the conflicting objectives of expected cost and risk. Petrik and Subramanian (2012) optimise a weighted combination of dynamic CVaR and expected value. For any given set of weightings, the performance for conditional value at risk may be sacrificed to improve the expected value. In contrast, our lexicographic approach ensures that the optimal CVaR is always attained.

An alternative approach to finding risk-averse solutions to MDPs is to impose constraints. Constrained MDPs optimise the expected value subject to a constraint on the expected value of a secondary cost function (Altman 1999; Santana, Thiébaux, and Williams 2016). By adding cost penalties to undesirable states, constraints may be used to discourage unwanted behaviour. However, this approach only constrains the expected cost and it is unclear how to choose cost penalties which result in the desired risk-averse behaviour.

CVaR-constrained problems have been addressed by existing works (Chow and Ghavamzadeh 2014; Chow et al. 2017; Borkar and Jain 2010; Prashanth 2014). In this approach, the user defines a CVaR threshold. The expected value is optimised subject to a constraint that the CVaR is less than the chosen threshold. A disadvantage of this approach is that it may be difficult to choose an appropriate CVaR threshold which is both feasible, and results in the desired level of risk-aversion.

Lexicographic approaches to multi-objective decision making in MDPs have been proposed (Mouaddib 2004; Wray, Zilberstein, and Mouaddib 2015; Lacerda, Parker, and Hawes 2015). These approaches optimise the expected value for each objective in a lexicographic ordering. Unlike these approaches, we consider risk-averse planning. To the best of our knowledge, this is the first work to propose a lexicographic approach to the optimisation of CVaR and expected value in sequential decision making problems.

Preliminaries

Conditional Value at Risk

Let Z be a bounded-mean random variable, i.e. $E[|Z|] < \infty$, on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$, with cumulative distribution function $F(z) = \mathcal{P}(Z \leq z)$. In this paper, we interpret Z as the total cost which is to be minimised. The *value at risk* (VaR) at confidence level $\alpha \in (0, 1]$ is defined as $\mathrm{VaR}_\alpha(Z) = \min\{z | F(z) \geq 1 - \alpha\}$. The *conditional value at risk* at confidence level α is defined as

$$\mathrm{CVaR}_\alpha(Z) = \frac{1}{\alpha} \int_{1-\alpha}^{1} \mathrm{VaR}_{1-\gamma}(Z) d\gamma. \tag{1}$$

If Z has a continuous distribution, $\mathrm{CVaR}_\alpha(Z)$ can be defined using the more intuitive expression: $\mathrm{CVaR}_\alpha(Z) = \mathrm{E}[Z \mid Z \geq \mathrm{VaR}_\alpha(Z)]$. Thus, $\mathrm{CVaR}_\alpha(Z)$ may be interpreted as the expected value of the α-portion of the right tail of the distribution of Z. CVaR may also be defined as the expected value under a perturbed distribution using its dual representation (Rockafellar, Uryasev et al. 2000; Shapiro, Dentcheva, and Ruszczyński 2014):

$$\mathrm{CVaR}_\alpha(Z) = \max_{\xi \in \mathcal{B}(\alpha, \mathcal{P})} \mathrm{E}_\xi \big[Z\big], \tag{2}$$

where $\mathrm{E}_\xi[Z]$ denotes the ξ-weighted expectation of Z, and the *risk envelope*, $\mathcal{B}$, is given by

$$\mathcal{B}(\alpha, \mathcal{P}) = \left\{ \xi \mid \xi(\omega) \in \left[0, \frac{1}{\alpha}\right], \int_{\omega \in \Omega} \xi(\omega)\mathcal{P}(\omega) d\omega = 1 \right\}. \tag{3}$$

where $\mathcal{P}(\omega)$ is the probability density function if Z is continuous, and the probability mass function if Z is discrete.

Therefore, the CVaR of a random variable Z may be interpreted as the expectation of Z under a worst-case perturbed distribution, $\xi\mathcal{P}$. The risk envelope is defined so that the probability of any outcome can be increased by a factor of at most $1/\alpha$, whilst ensuring the perturbed distribution is a valid probability distribution.

Markov Decision Processes

In this work, we consider stochastic shortest path (SSP) Markov decision processes (MDPs). An SSP MDP is a tuple, $\mathcal{M} = (S, A, C, T, G, s_0)$, where S and A are finite state and action spaces; $C : S \times A \to \mathrm{R}_+$ is the cost function; $T : S \times A \times S \to [0, 1]$ is the probabilistic transition function; $G \subset S$ is the set of absorbing goal states, from which the model incurs zero cost; and s_0 is the initial state. A history of $\mathcal{M}$ is a sequence $h = s_0 a_0 s_1 a_1 \ldots$ such that $T(s_i, a_i, s_{i+1}) > 0$ for all $i \in \{0, \ldots, |h|\}$, where $|h|$ denotes the length of h. We denote the set of all finite-length histories over $\mathcal{M}$ as $\mathcal{H}^{\mathcal{M}}_{\text{fin}}$ and the set of all infinite-length histories over $\mathcal{M}$ as $\mathcal{H}^{\mathcal{M}}_{\text{inf}}$, and define the set of all histories over $\mathcal{M}$ as $\mathcal{H}^{\mathcal{M}} = \mathcal{H}^{\mathcal{M}}_{\text{fin}} \cup \mathcal{H}^{\mathcal{M}}_{\text{inf}}$. The cumulative cost function $\text{cumul}_C : \mathcal{H}^{\mathcal{M}} \to \mathrm{R}^+$ is defined such that, given history $h = s_0 a_0 s_1 a_1 \ldots$, $\text{cumul}_C(h) = \sum_{t=0}^{|h|} C(s_t, a_t)$. A history-dependent policy is a function $\pi : \mathcal{H}^{\mathcal{M}}_{\text{fin}} \to A$, and we write $\Pi^{\mathcal{M}}_{\mathcal{H}}$ to denote the set of all history-dependent policies for $\mathcal{M}$. If π only depends on the last state s_t of h, then we say π is *Markovian*, and we denote the set of all Markovian policies as $\Pi^{\mathcal{M}}$. A policy π induces a probability distribution $\mathcal{P}^{\mathcal{M}}_{\pi}$ over $\mathcal{H}^{\mathcal{M}}_{\text{inf}}$, and we define the cumulative cost distribution $\mathcal{C}^{\mathcal{M}}_{\pi}$ as the distribution over the value of cumul_C for infinite-length histories of $\mathcal{M}$ under policy π. A policy is proper at s if it reaches $s_g \in G$ from s with probability 1. A policy is proper if it is proper at all states. In an SSP MDP, the following assumptions are made (Kolobov 2012): a) there exists a proper policy, and b) every improper policy incurs infinite cost at all states where it is improper. These assumptions ensure the expected value of the cumulative cost distribution is finite for at least one policy in the SSP.

Stochastic Games

In this paper, we formulate CVaR optimisation as a turn-based two-player zero-sum stochastic shortest path game (SSPG). An SSPG between an agent and an adversary is a generalisation of an SSP MDP and can be defined using a similar tuple $\mathcal{G} = (S, A, C, T, G, s_0)$. The elements of $\mathcal{G}$ are interpreted as with MDPs, but are extended to model a two-player game. S is partitioned into a set of agent states S_{agt}, and a set of adversary states S_{adv}. Similarly, A is partitioned into a set of agent actions A_{agt}, and a set of adversary actions A_{adv}. The transition function is defined such that agent actions can only be executed in agent states, and adversary actions can only be executed in adversary states.

We denote the set of Markovian agent policies mapping agent states to agent actions as $\Pi^{\mathcal{G}}$ and the set of Markovian adversary policies, defined similarly, as $\Sigma^{\mathcal{G}}$. Similar to MDPs, a pair (π, σ) of agent-adversary policies induces a probability distribution $\mathcal{P}^{\mathcal{G}}_{(\pi,\sigma)}$ over infinite-length histories, and we define $\mathcal{C}^{\mathcal{G}}_{(\pi,\sigma)}$ as the cumulative cost distribution of $\mathcal{G}$ under π and σ. In an SSPG, the agent seeks to minimise the expected cumulative cost, whilst the adversary seeks to maximise it:

$$\min_{\pi \in \Pi^{\mathcal{G}}} \max_{\sigma \in \Sigma^{\mathcal{G}}} \mathrm{E}\big[\mathcal{C}^{\mathcal{G}}_{(\pi,\sigma)}\big]. \tag{4}$$

In an SSPG, two assumptions are made to ensure that the value in (4) is finite: a) there exists a policy for the agent which is proper for all possible policies of the adversary, and b) for any states where π and σ are improper, the expected cost for the agent is infinite (Patek and Bertsekas 1999).

CVaR Optimisation in MDPs

Many existing works have addressed the problem of optimising the CVaR of $\mathcal{C}^{\mathcal{M}}_{\pi}$, defined as follows.

Problem 1. Let $\mathcal{M}$ be an MDP. Find the optimal CVaR of the cumulative cost at confidence level α:

$$\min_{\pi \in \Pi^{\mathcal{M}}_{\mathcal{H}}} \mathrm{CVaR}_\alpha(\mathcal{C}^{\mathcal{M}}_{\pi}). \tag{5}$$

Note that the optimal policy for Problem 1 may be history-dependent (Bäuerle and Ott 2011). Methods based on dynamic programming have been proposed to solve Problem 1 (Bäuerle and Ott 2011; Chow et al. 2015; Pflug and Pichler 2016). In particular, the current state of the art approach (Chow et al. 2015) formulates Problem 1 as the expected value in an SSPG against an adversary which modifies the transition probabilities. This formulation is based on the CVaR representation theorem in Eq. 2, where CVaR may be represented as the expected value under a perturbed probability distribution. The SSPG is defined so that the ability for the adversary to perturb the transition probabilities corresponds to the risk envelope given in Eq. 3.

Formally, the CVaR SSPG is defined by the tuple $\mathcal{G}^+ = (S^+, A^+, C^+, T^+, G^+, s_0^+)$. The state space $S^+ = S \times [0, 1] \times (A \cup \{\perp\})$ is the original MDP state space augmented with a continuous state factor, $y \in [0, 1]$, representing the "budget" of the adversary to perturb the probabilities; and a state factor $a \in A \cup \{\perp\}$ indicating the most recent agent action if it is the adversary's turn to choose an action, and $\perp$ if is the agent's turn. The action space is defined as $A^+ = A \cup \Xi$, i.e. the agent actions are the actions in the original MDP, and the adversary actions are a set Ξ that will be defined next.

The CVaR SSPG transition dynamics are as follows. Given an agent state, $(s, y, \perp) \in S^+_{agt}$, the agent applies an action, $a \in A$, and receives cost $C^+((s, y, \perp), a) = C(s, a)$. The state then transitions to the adversary state $(s, y, a) \in S^+_{adv}$. The adversary then chooses an action to perturb the original MDP transition probabilities from a continuous action space defined as:

$$\Xi(s,y,a) = \Big\{ \xi \in \mathrm{R}^{|S|} \mid 0 \leq \xi(s') \leq \eta \ \forall s' \\ \text{and} \sum_{s' \in S} \big[\xi(s') \cdot T(s,a,s')\big] = 1 \Big\}, \quad (6)$$

where $\eta = \infty$ if $y = 0$, and $\eta = 1/y$ otherwise; T is the original MDP transition function. Eq. 6 restricts the adversary actions so the probability of any history is increased by at most $1/y$, and the perturbed transition probabilities remain a valid probability distribution. After the adversary chooses the perturbation action $\xi \in \Xi(s,y,a)$, the game transitions back to an agent state $(s', y \cdot \xi(s'), \perp) \in S^+_{agt}$ according to the following transition function where the probability of each successor s' in the original MDP is perturbed by the factor $\xi(s')$:

$$T^+((s,y,a), \xi, (s', y\xi(s'), \perp)) = \xi(s')T(s,a,s'). \quad (7)$$

Finally, we define the initial augmented state as $s_0^+ = (s_0, \alpha, \perp) \in S^+_{agt}$, where the y state variable is set to the CVaR confidence level α. We also define G^+ as the set of goal states on the augmented state space corresponding to goal states in the original MDP.

Chow et al. (2015) showed that the minimax expected value equilibrium for $\mathcal{G}^+$ corresponds to the optimal CVaR.

Proposition 1. *(Chow et al. 2015) Let $\mathcal{G}^+$ be a CVaR SSPG corresponding to original MDP $\mathcal{M}$. Then:*

$$\min_{\pi \in \Pi^{\mathcal{M}}_{\mathcal{H}}} \mathrm{CVaR}_\alpha(\mathcal{C}^{\mathcal{M}}_\pi) = \min_{\pi \in \Pi^{\mathcal{G}^+}} \max_{\sigma \in \Sigma^{\mathcal{G}^+}} \mathrm{E}\big[\mathcal{C}^{\mathcal{G}^+}_{(\pi,\sigma)}\big]. \quad (8)$$

Proposition 1 holds because the y state variable keeps track of the total multiplicative perturbation to the probability of any history. Thus, the constraints in Eq. 6 ensure that the maximum perturbation to the probability of any history from the initial state is $1/\alpha$, and that the perturbed distribution over histories is a valid probability distribution. Therefore, the admissible adversarial perturbation actions in $\mathcal{G}^+$ correspond to the risk envelope in Eq. 3. According to Eq. 2, the CVaR is the expected value under the perturbations in the risk-envelope that maximise the expected cost.

To compute the solution to Eq. 8, we denote the value function for the augmented state by $V_{\mathrm{CV}}(s,y,\perp) = \min_{\pi \in \Pi^{\mathcal{M}}_{\mathcal{H}}} \mathrm{CVaR}_y(\mathcal{C}^{\mathcal{M}}_\pi)$. This value function can be computed using minimax value iteration over $\mathcal{G}^+$ using the following Bellman equation:

$$V_{\mathrm{CV}}(s,y,\perp) = \min_{a \in A} \Bigg[C(s,a) + \\ \max_{\xi \in \Xi(s,y,a)} \sum_{s' \in S} \xi(s')T(s,a,s')V_{\mathrm{CV}}(s', y\xi(s'), \perp) \Bigg]. \quad (9)$$

We denote the policy corresponding to the value function obtained by solving Eq. 9 by π_{CV}. As the augmented state space contains a continuous variable and the action space is also continuous, Chow et al. (2015) use approximate dynamic programming with linear function approximation.

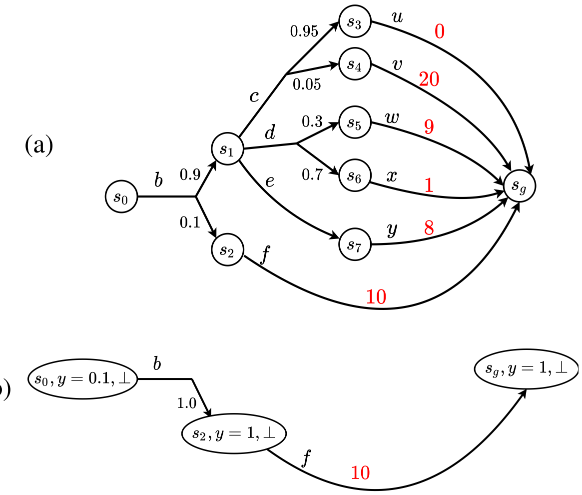

Figure 2: (a) Example Markov decision process with initial state s_0 and goal state s_g. Letters indicate actions. Costs are assumed to be zero unless indicated otherwise in red. (b) Corresponding Markov chain induced by the solution to Equation 8 when $\alpha = 0.1$.

Example 1. In Fig. 2 we present an illustrative example of the approach to CVaR optimisation from Chow et al. (2015). Fig. 2a depicts an example MDP. Fig. 2b shows the solution to Eq. 8 to optimise the CVaR when $\alpha = 0.1$, illustrated as a Markov chain (MC): the adversary perturbs the transition probabilities so that the probability of transitioning from s_0 to s_2 is increased from 0.1 to 1 and, conversely, the probability of transitioning from s_0 to s_1 is decreased from 0.9 to 0. The expected value in this MC is 10, and by Proposition 1 this is the optimal CVaR at $\alpha = 0.1$ in the original MDP.

Remark 1. State s_1 is reachable in the original MDP, but assigned zero probability by the adversary. At states assigned zero probability by the adversary, $y = 0$ in the CVaR SSPG. When $y = 0$, the adversary has unlimited power to perturb the transition probabilities (Eq. 6). Therefore, when $y = 0$ the minimax value iteration in Eq. 9 optimises for the minimum worst-case cost, as the adversary has the power to make the agent transition deterministically to the worst possible successor state for all future transitions. We will refer to the approach from Chow et al. (2015) as *CVaR-Worst-Case* because of this property that for histories assigned zero probability by the adversary, this method optimises for the minimum worst-case total cost. In the following section we show that this behaviour is unnecessarily conservative to optimise CVaR. We propose a method for finding an alternative policy to execute in such situations that optimises the expected value while maintaining the optimal CVaR.

Lexicographic Optimisation of CVaR

In this section we present the contributions of this paper. We begin with a formal problem statement.

Problem 2. Let $\mathcal{M}$ be an MDP. Find the policy π^* that optimises the expected cost subject to the constraint that π^*

obtains the optimal CVaR at confidence level α:

$$\begin{aligned}\pi^* = &\underset{\pi \in \Pi_{\mathcal{H}}^{\mathcal{M}}}{\arg\min}\ \mathrm{E}\big[\mathcal{C}_{\pi}^{\mathcal{M}}\big], \\ &\text{s.t.}\quad \mathrm{CVaR}_{\alpha}(\mathcal{C}_{\pi}^{\mathcal{M}}) = \min_{\pi' \in \Pi_{\mathcal{H}}^{\mathcal{M}}} \mathrm{CVaR}_{\alpha}(\mathcal{C}_{\pi'}^{\mathcal{M}}).\end{aligned} \tag{10}$$

Our approach to this problem extends the approach to CVaR optimisation from Chow et al. (2015), which we have outlined. We begin by emphasising that in the approach in Chow et al. (2015), some histories in the original MDP have zero probability under the adversarial perturbations obtained by solving Eq. 8. For example, all histories passing through s_1 in Fig. 2a are not reachable under the adversarial perturbations, as illustrated in Fig. 2b. Intuitively, this suggests that such histories do not contribute to the CVaR, as the CVaR can be computed as the expected value under the perturbed transition probabilities (Proposition 1).

In this paper, we investigate finding an alternative policy to execute when we reach such histories which are assigned zero probability by the adversary in the CVaR SSPG. By finding an appropriate alternative policy to execute in these situations, we are able to optimise the expected value whilst maintaining the optimal CVaR. We now state two properties of these histories which will be useful to ensure the policies obtained by our algorithm maintain the optimal CVaR. Full proofs are in the supplementary material.

Proposition 2. *Let π denote a CVaR-optimal policy for confidence level α, and let σ denote the corresponding adversary policy from Eq. 8. For some history, h, if $\mathcal{P}_{\pi}^{\mathcal{M}}(h) > 0$ and $\mathcal{P}_{(\pi,\sigma)}^{\mathcal{G}}(h) = 0$, there exists a policy $\pi' \in \Pi_{\mathcal{H}}^{\mathcal{M}}$ which may be executed from h onwards for which the total cost received over the run is guaranteed to be less than or equal to VaR$_{\alpha}(\mathcal{C}_{\pi}^{\mathcal{M}})$.*

Proof Sketch: We prove Proposition 2 by showing that the original policy, π, satisfies this condition. Denote by $\mathcal{H}_g^{\mathcal{M}}$ the set all of histories ending at a goal state, and let $h_g \in \mathcal{H}_g^{\mathcal{M}}$ denote any such history. Using the CVaR representation theorem in Equation 2, we can show that if h_g has non-zero probability under π in the original MDP, but is assigned zero probability by the adversary in the CVaR SSPG, then the total cost of h_g must be less than or equal to $\mathrm{VaR}_{\alpha}(\mathcal{C}_{\pi}^{\mathcal{M}})$:

$$\mathcal{P}_{\pi}^{\mathcal{M}}(h_g) > 0 \text{ and } \mathcal{P}_{(\pi,\sigma)}^{\mathcal{G}}(h_g) = 0 \implies \mathrm{cumul}_C(h_g) \leq \mathrm{VaR}_{\alpha}(\mathcal{C}_{\pi}^{\mathcal{M}}). \tag{11}$$

Now consider any history, h, which has not reached the goal. Assume that $\mathcal{P}_{\pi}^{\mathcal{M}}(h) > 0$ and $\mathcal{P}_{(\pi,\sigma)}^{\mathcal{G}}(h) = 0$. For all histories $h_g \in \mathcal{H}_g^{\mathcal{M}}$ reachable after h under π (i.e. for which $\mathcal{P}_{\pi}^{\mathcal{M}}(h_g) > 0$), we have that $\mathcal{P}_{(\pi,\sigma)}^{\mathcal{G}}(h_g) = 0$. Therefore, by Equation 11 all histories h_g reachable after h under π are guaranteed to have $\mathrm{cumul}_C(h_g) \leq \mathrm{VaR}_{\alpha}(\mathcal{C}_{\pi}^{\mathcal{M}})$. This proves that by continuing to execute π, the total cost received over the run is guaranteed to be less than or equal to $\mathrm{VaR}_{\alpha}(\mathcal{C}_{\pi}^{\mathcal{M}})$.

Proposition 2 shows that if a state is reached which is assigned zero probability by the adversary, then there must exist a policy which can be executed from that state onwards for which the total cost is guaranteed not to exceed the VaR. The following proposition states that by switching to any policy that guarantees that the cost will not exceed the VaR, the optimal CVaR is maintained.

Proposition 3. *During execution of policy π optimal for* $\mathrm{CVaR}_{\alpha}(\mathcal{C}_{\pi}^{\mathcal{M}})$*, we may switch to any policy π' and still attain the same CVaR, provided that π' is guaranteed to incur total cost of less than or equal to* $\mathrm{VaR}_{\alpha}(\mathcal{C}_{\pi}^{\mathcal{M}})$.

Proof Sketch: $\mathrm{CVaR}_{\alpha}(\mathcal{C}_{\pi}^{\mathcal{M}})$ is computed by integrating over the distribution of costs greater than or equal to $\mathrm{VaR}_{\alpha}(\mathcal{C}_{\pi}^{\mathcal{M}})$ (Equation 1). Because switching to π' does not result in any costs greater than $\mathrm{VaR}_{\alpha}(\mathcal{C}_{\pi}^{\mathcal{M}})$, the strategy of switching to π' cannot increase the CVaR. Switching to π' cannot decrease the CVaR as π already attains the optimal CVaR. Therefore, the strategy of switching to an appropriate π' must attain the same CVaR.

CVaR-Expected-Value

Proposition 3 establishes that during execution of π_{CV} we can switch to another policy, π', without influencing the CVaR, provided that the total cost of executing π' is guaranteed not to exceed $\mathrm{VaR}_{\alpha}(\mathcal{C}_{\pi_{\mathrm{CV}}}^{\mathcal{M}})$. Proposition 2 establishes that such a policy exists if the current history is assigned zero probability by the adversary in the CVaR SSPG. However, as we shall illustrate in the following example there may be multiple policies which satisfy this criterion. Of these policies, we would like to find the one which optimises our secondary objective of expected value as stated in Problem 2.

Example 2. Consider again the model in Fig. 2. We established in Example 1 that the optimal CVaR at $\alpha = 0.1$ is 10, which can be computed as the expected value under the Markov chain in Fig. 2b. The corresponding VaR at $\alpha = 0.1$ is also 10. All of the histories passing through s_1 are assigned zero probability by the adversarial perturbations in the CVaR SSPG. At s_1, we can continue to execute any policy for which all histories are guaranteed to reach the goal with total cost less than or equal to the VaR threshold of 10. Executing either d or e at s_1 satisfies this property, and maintains the optimal CVaR of 10. The approach of Chow et al. (2015), which we refer to as *CVaR-Worst-Case*, would choose action e as in this situation it optimises for the minimum worst-case cost (see Remark 1). However, d achieves better expected value and still attains the optimal CVaR of 10. Therefore, to solve Problem 2, the policy should choose d. On the other hand, executing c attains a sub-optimal CVaR of greater than 10, as some histories would receive a total cost of 20.

We wish to develop a general approach to finding the policy, π', which optimises the expected value, subject to the constraint that the worst-case total cost must not exceed $\mathrm{VaR}_{\alpha}(\mathcal{C}_{\pi_{\mathrm{CV}}}^{\mathcal{M}})$. We know that such a policy exists for histories assigned zero probability by the adversary in the CVaR SSPG. Therefore, by switching from π_{CV} to π' when such histories occur, we can optimise the expected value while still attaining the optimal CVaR, thus solving Problem 2.

We first compute the minimum worst-case total cost at each state, $V_{worst}(s)$, using the following minimax Bellman equation which assumes that the agent always transitions deterministically to the worst-case successor state:

$$V_{worst}(s) = \min_{a \in A} \Big[C(s,a) + \max_{s' \in S} \Big(\mathbf{1}(T(s,a,s') > 0) \cdot V_{worst}(s') \Big) \Big], \tag{12}$$

where $\mathbf{1}$ denotes the indicator function. Let π_{worst} denote the policy corresponding to V_{worst}. We also compute the optimal worst-case Q-values:

$$Q_{worst}(s,a) = C(s,a) + \max_{s' \in S} \mathbf{1}(T(s,a,s') > 0) \cdot V_{worst}(s'). \tag{13}$$

Now, assume that the history so far is h, and that the cost so far in the history is $\text{cumul}_C(h)$. To maintain the optimal CVaR according to Proposition 3, we can allow an action a to be executed at h if

$$\text{cumul}_C(h) + Q^*_{worst}(s,a) \leq \text{VaR}_\alpha(\mathcal{C}^{\mathcal{M}}_{\pi_{\text{CV}}}). \tag{14}$$

For any action a which satisfies Eq. 14, after taking a we can then execute π_{worst} and guarantee that the total cost will not exceed $\text{VaR}_\alpha(\mathcal{C}^{\mathcal{M}}_{\pi_{\text{CV}}})$. Thus, by finding a policy which optimises the expected value subject to the constraint that actions must satisfy Eq. 14, we can find the policy with best expected value which is guaranteed to have total cost less than or equal to $\text{VaR}_\alpha(\mathcal{C}^{\mathcal{M}}_{\pi_{\text{CV}}})$ and maintain the optimal CVaR.

Note that which actions are allowed in Eq. 14 depends not only on the current state, but also on the cost received so far. To arrive at an offline solution we create an augmented MDP, $\mathcal{M}' = (S', A, T', C', G', s'_0)$, where we restrict which actions can be executed depending on the cost received so far. We start by augmenting the state space of the original MDP such that $S' = S \times [0, \text{VaR}_\alpha(\mathcal{C}^{\mathcal{M}}_{\pi_{\text{CV}}})]$. This augmented state keeps track of how much cost has been incurred. The action set A is the same as the original MDP. The transition function ensures that actions are only enabled if they satisfy Eq. 14, and propagates the accumulated cost appropriately:

$$T'((s,c),a,(s',c')) = \begin{cases} T(s,a,s') & \text{if } c + Q_{worst}(s,a) \leq \text{VaR}_\alpha(\mathcal{C}^{\mathcal{M}}_{\pi_{\text{CV}}}) \\ & \text{and } c' = c + C(s,a), \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

The cost function is $C'((s,c),a) = C(s,a)$; the goal states are $G' = G \times [0, \text{VaR}_\alpha(\mathcal{C}^{\mathcal{M}}_{\pi_{\text{CV}}})]$; and the initial state is $s'_0 = (s_0, 0)$. The value function corresponding to the following standard MDP Bellman equation is the optimal expected value subject to the constraint that $\text{VaR}_\alpha(\mathcal{C}^{\mathcal{M}}_{\pi_{\text{CV}}})$ will not be exceeded for any possible history.

$$V^{\mathcal{M}'}(s,c) = \min_{a \in A} \Big[C'(s,a) + \sum_{(s',c')} T'\big((s,c),a,(s',c')\big) \cdot V^{\mathcal{M}'}(s',c') \Big]. \tag{16}$$

Solving Eq. 16 is not straightforward due to the additional continuous state variable which prohibits standard discrete value iteration. Therefore, we instead apply value iteration

Algorithm 1: *CVaR-Expected-Value*

```
get π_CV and V_CV(s, y) by solving Equation 9
get π' by solving Equation 16
s⁺ ← s⁺_0
c ← 0
function ExecuteEpisode()
    do
        a ← π_CV(s⁺)
        ξ = arg max_{ξ∈Ξ(s,y,a)} Σ_{s'∈S} ξ(s') · T(s, a, s') ·
            V_CV(s', y · ξ(s'), ⊥)
        s⁺ ← (s', y · ξ(s'), ⊥), where s' is successor after
            executing a in the MDP
        c ← c + C(s, a)
        if s⁺ ∈ G⁺ :
            return
    while ξ(s') > 0

    while s ∉ G :
        a ← π'(s, c)
        s ← s', where s' is successor after executing a in
            the MDP
        c ← c + C(s, a)
    return
```

with linear interpolation (Bertsekas 2007) for the continuous variable in the same manner as (Chow et al. 2015).

We write π' to denote the optimal policy corresponding to the value function $V^{\mathcal{M}'}$. By first executing π_{CV}, and then executing π' on histories assigned zero probability by the adversary in the CVaR SSPG, we switch to using the policy with best expected value subject to the constraint that the optimal CVaR is maintained. Therefore, this approach solves Problem 2. This approach, which we refer to as *CVaR-Expected-Value* is described in Algorithm 1.

In this approach, we decide which actions are pruned out based on $\text{VaR}_\alpha(\mathcal{C}^{\mathcal{M}}_{\pi_{\text{CV}}})$. To estimate $\text{VaR}_\alpha(\mathcal{C}^{\mathcal{M}}_{\pi_{\text{CV}}})$, one approach would be to compute the worst-possible history in the Markov chain induced by Eq. 8. However, this is computationally challenging as the number of reachable histories in the continuous state space may be very large. Therefore, we take the simple approach of first executing π_{CV} (*CVaR-Worst-Case*) for many episodes and compute a Monte Carlo estimate for $\text{VaR}_\alpha(\mathcal{C}^{\mathcal{M}}_{\pi_{\text{CV}}})$ (Hong, Hu, and Liu 2014), which we then use to restrict actions according to Eq. 15.

Experiments

The code and data used to run the experiments is included in the supplementary material and will be made publicly available. We experimentally evaluate the following three approaches: *CVaR-Worst-Case* (*CVaR-WC*), *CVaR-Expected-Value* (*CVaR-EV*), and *Expected Value* (*EV*). *Expected Value* is the policy which optimises the expected value only. All algorithms are implemented in C++ and Gurobi is used to solve linear programs where necessary. We use 30 interpolation points for α to solve Eq. 9, and 100 interpolation points for the cumulative cost to solve Eq. 16. The experiments used a 3.2 GHz Intel i7 processor with 64 GB of RAM.

We compare the approaches on the following four domains. The three synthetic domains are from the literature, and the fourth domain we introduce is based on real data.

Inventory Control (IC) We consider the stochastic inventory control problem (Puterman 2005; Ahmed et al. 2017), with 10 decision stages. The current number of units in the inventory is n, and the maximum number of units is $N = 20$. The action set is $a \in \{0, \ldots, N-n\}$, representing the amount of stock to purchase at each stage. There is an expense of $p_u = 1$ for purchasing each item of stock. Items are sold according to the stochastic demand for each stage, $d \in \{0, \ldots, N\}$. A revenue of $r = 3$ is received for each unit sold. For any items of stock which do not sell, there is a holding expense at each stage of $p_h = 1$ per unit. Thus, the profit received at each stage is

$$min\{d, n+a\} \cdot r - a \cdot p_u - max\{n+a-d, 0\} \cdot p_h$$

We assume that the demand is modelled as random walk. At each stage, $d = d_{prev} + \Delta d$, where d_{prev} is the demand at the previous stage, and Δd is uniformly distributed between ± 5. At the first stage, $d_{prev} = 10$. The demand at the previous stage is included in the state space.

We model all domains as SSP MDPs, where a positive cost function is minimised. To convert rewards to a positive cost function, we take the standard approach (Kolobov 2012) of defining the cost function to be the maximum possible reward minus the reward received. In Inventory Control, the maximum possible profit is: $\max(\text{profit}) = 400$. The cost for an episode is $\max(\text{profit})$ minus the cumulative profit received over all stages. Therefore, a cost of under 400 represents a net profit, and a cost over 400 represents a net loss.

Betting Game (BG) We adapt this domain from the literature on CVaR in MDPs (Bäuerle and Ott 2011). The state is represented by two factors: $(money, stage)$. The agent begins with $money = 5$. The amount of money that the agent can have is limited between 0, and $\max(money) = 100$. At each stage the agent may choose to place a bet from $bets = \{0, 1, 2, 3, 4, 5\}$ provided that sufficient money is available. If the agent wins the game at that stage, an amount of money equal to the bet placed is received. If the agent loses the stage, the money bet is lost. If the agent wins the jackpot, the agent receives $10\times$ the amount bet. For each stage, the probability of winning is 0.7, the probability of winning the jackpot is 0.05, and the probability of losing is 0.25. After 10 stages, the cost received is $\max(money) - money$.

Deep Sea Treasure (DST) This domain is adapted from the literature on multi-objective optimisation in MDPs (Vamplew et al. 2011). A submarine navigates a grid-world to collect one of many treasures, each of which is associated with a reward value, r (illustrated in the supplementary material). At each timestep, the agent chooses from 8 actions corresponding to directions of travel and moves to the corresponding square with probability 0.6 and each of the adjacent squares with probability 0.2. The episode ends when the agent reaches a treasure or the horizon of 15 steps is reached. The agent incurs a cost of 5 at each timestep, and a terminal cost of $500 - r$. There is a tradeoff between risk and expected cost as if the submarine travels further it may collect more valuable treasure, and therefore incur less cost, but risks running out of time before reaching any treasure.

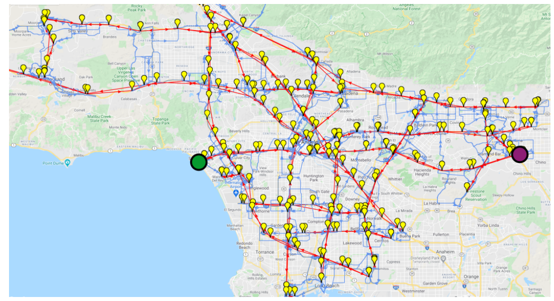

Figure 3: Autonomous Vehicle Navigation domain: MDP state space (yellow balloons) and transitions (red and blue edges) overlaid onto map of Los Angeles, USA. The green and purples circles indicate the start and goal locations.

Autonomous Navigation (AN) An autonomous vehicle must plan routes across Los Angeles, USA between a start and goal location as illustrated in Fig. 3. We access real road traffic data collected by Caltrans Performance Measurement System (PeMS) by over 39,000 real-time traffic sensors deployed across the major metropolitan areas of California (Chen et al. 2001). We select a subset of 263 sensors along the major freeways, shown as yellow markers in Figure 3. We specify two types of transitions: i) *freeway* transitions (red) along a specified freeway where the transition time distribution is generated from historical PeMS traffic data (discrete with 10 bins), and ii) *between-freeway* transitions (blue) where each state is connected to its three nearest neighbours on other freeways and the transition time is normally distributed around the expected duration obtained from querying the Google Routes API. In this domain, the cost is the journey duration in minutes. To simulate rare traffic jams on the freeways, uniform noise in [0, 0.1] is added to the slowest *freeway* transitions (and probabilities renormalised). This is motivated by knowledge that rare but severe traffic can affect transition durations on freeways, and introduces a tradeoff between risk and expected cost.

Results

The results of our experiments are in Table **??**. The rows in the table indicate the method and the confidence level that the method is set to optimise. The columns indicate the performance of the policy for each objective over 20,000 evaluation episodes in each domain. We measure performance for $\text{CVaR}_{0.02}$ (i.e. the mean cost of the worst 2% of runs), $\text{CVaR}_{0.2}$ (worst 20%), and the expected value. We expect the *CVaR-EV* method to match the *CVaR-WC* performance on its CVaR measurement column. We also expect *CVaR-EV* to achieve lower cost than *CVaR-WC* in expectation.

In the Inventory Control domain, *EV* performs the best for expected value, but the worst for the CVaR objectives. For the $\text{CVaR}_{0.02}$ objective, *EV* obtains 416, representing an average net loss of 16 on the worst 2% of runs. In contrast, when optimising for $\text{CVaR}_{0.02}$ (i.e. $\alpha = 0.02$), both *CVaR-WC* and *CVaR-EV* obtain values of 386 (profit of 14) for the $\text{CVaR}_{0.02}$ objective. This illustrates that the risk-averse approaches avoid the possibility of losses on poor runs. Similarly, we see that *CVaR-EV* and *CVaR-WC* equally out-

	Inventory Control (IC)		Betting Game (BG)		Deep Sea Treasure (DST)		Autonomous Navigation (AN)	
Method	$\text{CVaR}_{0.02}$	Expected Value	$\text{CVaR}_{0.02}$	Expected Value	$\text{CVaR}_{0.02}$	Expected Value	$\text{CVaR}_{0.02}$	Expected Value
CVaR-WC ($\alpha = 0.02$)	386.49 (0.23)	286.18 (0.50)	**95.0** (0.0)	**95.0** (0.0)	502.25 (0.78)	402.74 (0.43)	210.20 (1.36)	167.35 (0.11))
CVaR-EV ($\alpha = 0.02$)	**386.92** (0.24)	**250.38** (0.66)	**95.0** (0.0)	**95.0** (0.0)	**502.98** (0.83)	**352.06** (0.57)	**211.16** (1.40)	**164.14** (0.12)
Expected Value	416.42 (0.60)	235.62 (0.70)	100.0 (0.0)	58.26 (0.22)	575.0 (0.0)	315.15 (0.66)	315.99 (2.12)	120.19 (0.38)
Method	$\text{CVaR}_{0.2}$	Expected Value	$\text{CVaR}_{0.2}$	Expected Value	$\text{CVaR}_{0.2}$	Expected Value	$\text{CVaR}_{0.2}$	Expected Value
CVaR-WC ($\alpha = 0.2$)	360.65 (0.31)	272.51 (0.48)	91.97 (0.08)	82.95 (0.06)	422.29 (0.62)	349.45 (0.42)	172.05 (0.80)	134.12 (0.21)
CVaR-EV ($\alpha = 0.2$)	**360.29** (0.31)	**250.08** (0.63)	**91.86** (0.08)	**75.63** (0.16)	**422.80** (0.63)	**340.12** (0.49)	**171.58** (0.80)	**132.77** (0.22)
Expected Value	370.54 (0.37)	235.62 (0.70)	97.36 (0.07)	58.26 (0.22)	453.07 (1.09)	315.15 (0.66)	217.86 (0.71)	120.19 (0.38)

Table 1: Results from evaluating each method for 20,000 episodes. Rows indicate the optimisation method. Columns indicate the performance for the CVaR and expected value objectives in each domain. The bolded results indicate the method with best performance for expected value given that the CVaR objective is optimal (i.e. Problem 2). Brackets indicate standard errors.

perform *EV* at optimising the $\text{CVaR}_{0.2}$ objective. For both $\alpha = 0.02$ and $\alpha = 0.2$, *CVaR-EV* significantly improves the expected value compared to *CVaR-WC*, meaning that the average profitability is improved while still avoiding risks.

Histograms for the total costs received by *CVaR-WC* ($\alpha = 0.02$), *CVaR-EV* ($\alpha = 0.02$), and *EV* are shown in Figure 4 for the Inventory Control domain. Equivalent plots for the other domains are in the supplementary material. We see that while *EV* performs the best in expectation, it incurs the most runs where the cost is above 400, corresponding to net losses. Therefore, *EV* performs worse at $\text{CVaR}_{0.02}$. For *CVaR-WC* and *CVaR-EV*, the right tail of the distributions are equivalent, resulting in the same performance for $\text{CVaR}_{0.02}$. However, for *CVaR-EV*, the left side of the distribution is spread further left, improving the expected value.

Similarly, in the Betting Game domain *EV* performs the best for expected value, but worse for the CVaR objectives. For *CVaR-WC* and *CVaR-EV* with $\alpha = 0.02$, the optimal policy is never to bet, and this policy attains the best performance for the $\text{CVaR}_{0.02}$ objective. For $\alpha = 0.2$, both *CVaR-WC* and *CVaR-EV* achieve similar performance for $\text{CVaR}_{0.2}$. However, *CVaR-EV* attains significantly lower cost in expectation. This occurs because winning the jackpot is usually sufficient to guarantee that the VaR will not be exceeded. In these situations, *CVaR-WC* stops betting. On the other hand, *CVaR-EV* bets aggressively in these situations, as bets can safely be made without the risk of having a bad run which would influence the CVaR.

For both the Deep Sea Treasure and Autonomous Navigation domains, we make the same observation that both *CVaR-WC* and *CVaR-EV* achieve the same CVaR performance when optimising each of the $\text{CVaR}_{0.02}$ and $\text{CVaR}_{0.2}$ objectives. However, *CVaR-EV* obtains better expected value performance. In both domains, we also see that *EV* performs the best in expectation, but less well at the CVaR objectives.

The computation times in Table **??** indicate that for three out of four domains, the computation required for *CVaR-EV* is only a moderate (5%-30%) increase over the computation required for *CVaR-WC*. For Inventory Control, the increase in computation time is more substantial (150%).

Conclusion

In this paper, we have presented an approach to optimising the expected value in MDPs subject to the constraint that the CVaR is optimal. Our experimental evaluation on four domains has demonstrated that our approach is able to attain optimal CVaR while improving the expected performance compared to the current state of the art method. In future work, we wish to improve scalability by extending our approach to use labelled real-time dynamic programming (Bonet and Geffner 2003) rather than value iteration over the entire state space.

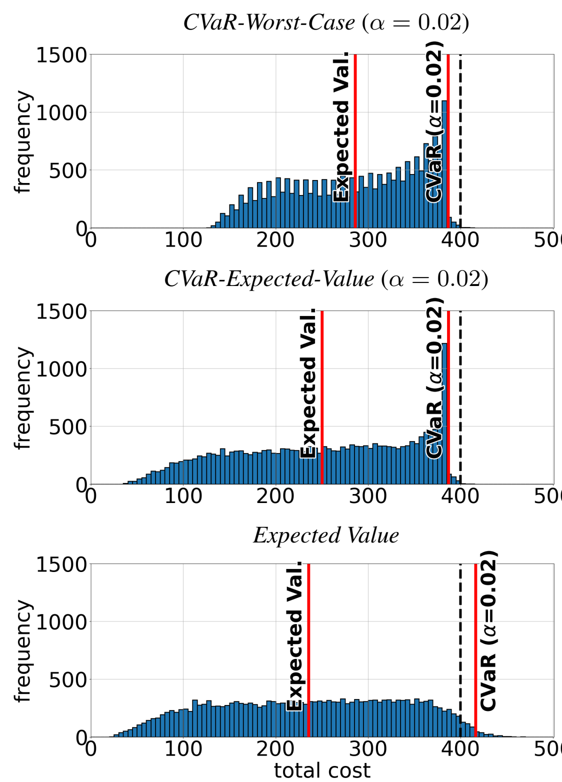

Figure 4: Histograms for the total cost received over 20000 evaluation episodes in the Inventory Control domain. Total costs of greater than 400 (dashed black line) represent losses, whereas total costs of less than 400 represent profit.

Method	IC	BG	DST	AN
CVaR-Worst-Case	19637	6215	8156	18327
CVaR-Expected-Value	48527	6526	10653	23020
Expected Value	8303	34.5	505	1876

Table 2: Computation times for each approach in seconds.

References

Ahiska, S. S.; Appaji, S. R.; King, R. E.; and Warsing Jr, D. P. 2013. A Markov decision process-based policy characterization approach for a stochastic inventory control problem with unreliable sourcing. *International Journal of Production Economics*, 144(2): 485–496.

Ahmadi-Javid, A. 2012. Entropic value-at-risk: A new coherent risk measure. *Journal of Optimization Theory and Applications*, 155(3): 1105–1123.

Ahmed, A.; Varakantham, P.; Lowalekar, M.; Adulyasak, Y.; and Jaillet, P. 2017. Sampling based approaches for minimizing regret in uncertain Markov decision processes (MDPs). *Journal of Artificial Intelligence Research*, 59: 229–264.

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

Artzner, P.; Delbaen, F.; Eber, J.-M.; and Heath, D. 1999. Coherent measures of risk. *Mathematical finance*, 9(3): 203–228.

Basel Committee on Banking Supervision. 2014. Fundamental review of the trading book: A revised market risk framework. *Bank for International Settlements*.

Bäuerle, N.; and Ott, J. 2011. Markov decision processes with average-value-at-risk criteria. *Mathematical Methods of Operations Research*, 74(3): 361–379.

Bertsekas, D. P. 2007. *Dynamic Programming and Optimal Control, Vol. II*. Athena Scientific, 3rd edition.

Bonet, B.; and Geffner, H. 2003. Labeled RTDP: Improving the convergence of real-time dynamic programming. In *International Conference on Automated Planning and Scheduling*, 12–21.

Borkar, V.; and Jain, R. 2010. Risk-constrained Markov decision processes. In *49th IEEE Conference on Decision and Control (CDC)*, 2664–2669. IEEE.

Chen, C.; Petty, K.; Skabardonis, A.; Varaiya, P.; and Jia, Z. 2001. Freeway performance measurement system: mining loop detector data. *Transportation Research Record*, 1748(1): 96–102.

Chow, Y.; and Ghavamzadeh, M. 2014. Algorithms for CVaR optimization in MDPs. In *Advances in Neural Information Processing Systems*, 3509–3517.

Chow, Y.; Ghavamzadeh, M.; Janson, L.; and Pavone, M. 2017. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1): 6070–6120.

Chow, Y.; Tamar, A.; Mannor, S.; and Pavone, M. 2015. Risk-sensitive and robust decision-making: a CVaR optimization approach. In *Advances in Neural Information Processing Systems*, 1522–1530.

Filar, J. A.; Krass, D.; and Ross, K. W. 1995. Percentile performance criteria for limiting average Markov decision processes. *IEEE Transactions on Automatic Control*, 40(1): 2–10.

Hong, L. J.; Hu, Z.; and Liu, G. 2014. Monte Carlo methods for value-at-risk and conditional value-at-risk: a review. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 1–37.

Howard, R. A.; and Matheson, J. E. 1972. Risk-sensitive Markov decision processes. *Management science*, 18(7): 356–369.

Keramati, R.; Dann, C.; Tamkin, A.; and Brunskill, E. 2020. Being optimistic to be conservative: Quickly learning a CVaR policy. In *AAAI Conference on Artificial Intelligence*, volume 34, 4436–4443.

Kolobov, A. 2012. Planning with Markov decision processes: An AI perspective. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1): 1–210.

Lacerda, B.; Faruq, F.; Parker, D.; and Hawes, N. 2019. Probabilistic planning with formal performance guarantees for mobile service robots. *The International Journal of Robotics Research*, 38(9): 1098–1123.

Lacerda, B.; Parker, D.; and Hawes, N. 2015. Nested Value Iteration for Partially Satisfiable Co-Safe LTL Specifications. In *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents (SDMIA)*.

Majumdar, A.; and Pavone, M. 2020. How should a robot assess risk? Towards an axiomatic theory of risk in robotics. In *Robotics Research*, 75–84. Springer.

Mouaddib, A.-I. 2004. Multi-objective decision-theoretic path planning. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings.*, volume 3, 2814–2819.

Patek, S. D.; and Bertsekas, D. P. 1999. Stochastic shortest path games. *SIAM Journal on Control and Optimization*, 804–824.

Petrik, M.; and Subramanian, D. 2012. An Approximate Solution Method for Large Risk-Averse Markov Decision Processes. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 805–814.

Pflug, G. C.; and Pichler, A. 2016. Time-consistent decisions and temporal decomposition of coherent risk functionals. *Mathematics of Operations Research*, 41(2): 682–699.

Prashanth, L. 2014. Policy gradients for CVaR-constrained MDPs. In *International Conference on Algorithmic Learning Theory*, 155–169. Springer.

Puterman, M. L. 2005. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rigter, M.; Lacerda, B.; and Hawes, N. 2021. Risk-averse Bayes-adaptive reinforcement learning. In *Advances in Neural Information Processing Systems*.

Rockafellar, R. T.; Uryasev, S.; et al. 2000. Optimization of conditional value-at-risk. *Journal of risk*, 2: 21–42.

Ruszczyński, A. 2010. Risk-averse dynamic programming for Markov decision processes. *Mathematical programming*, 125(2): 235–261.

Santana, P.; Thiébaux, S.; and Williams, B. 2016. RAO*: An algorithm for chance-constrained POMDP's. *AAAI Conference on Artificial Intelligence*.

Shapiro, A.; Dentcheva, D.; and Ruszczyński, A. 2014. *Lectures on stochastic programming: modeling and theory*. SIAM.

Sobel, M. J. 1982. The variance of discounted Markov decision processes. *Journal of Applied Probability*, 19(4): 794–802.

Tamar, A.; Chow, Y.; Ghavamzadeh, M.; and Mannor, S. 2015. Policy gradient for coherent risk measures. In *Advances in Neural Information Processing Systems*, 1468–1476.

Tamar, A.; Chow, Y.; Ghavamzadeh, M.; and Mannor, S. 2016. Sequential decision making with coherent risk. *IEEE Transactions on Automatic Control*, 62(7): 3323–3338.

Tamar, A.; Glassner, Y.; and Mannor, S. 2015. Optimizing the CVaR via Sampling. In *AAAI Conference on Artificial Intelligence*, 2993–2999.

Tang, Y. C.; Zhang, J.; and Salakhutdinov, R. 2020. Worst cases policy gradients. *Conference on Robot Learning*.

Vamplew, P.; Dazeley, R.; Berry, A.; Issabekov, R.; and Dekker, E. 2011. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine learning*, 84(1): 51–80.

Wang, S. S. 2000. A class of distortion operators for pricing financial and insurance risks. *Journal of risk and insurance*, 15–36.

Wray, K.; Zilberstein, S.; and Mouaddib, A.-I. 2015. Multi-objective MDPs with conditional lexicographic reward preferences. In *AAAI Conference on Artificial Intelligence*, volume 29.