Risk-Sensitive Reinforcement Learning: 
A Constrained Optimization Viewpoint

Prashanth L.A.∗1 and Michael Fu†2

1Department of Computer Science and Engineering, Indian Institute of Technology Madras
2Robert H. Smith School of Business & Institute for Systems Research, University of Maryland

Abstract

The classic objective in a reinforcement learning (RL) problem is to find a policy that minimizes, in expectation, a long-run objective such as the infinite-horizon discounted or long-run average cost. In many practical applications, optimizing the expected value alone is not sufficient, and it may be necessary to include a risk measure in the optimization process, either as the objective or as a constraint. Various risk measures have been proposed in the literature, e.g., mean-variance tradeoff, exponential utility, the percentile performance, value at risk, conditional value at risk, prospect theory and its later enhancement, cumulative prospect theory. In this article, we focus on the combination of risk criteria and reinforcement learning in a constrained optimization framework, i.e., a setting where the goal is to find a policy that optimizes the usual objective of infinite-horizon discounted/average cost, while ensuring that an explicit risk constraint is satisfied. We introduce the risk-constrained RL framework, cover popular risk measures based on variance, conditional value-at-risk and cumulative prospect theory, and present a template for a risk-sensitive RL algorithm. We survey some of our recent work on this topic, covering problems encompassing discounted cost, average cost, and stochastic shortest path settings, together with the aforementioned risk measures in a constrained framework. This non-exhaustive survey is aimed at giving a flavor of the challenges involved in solving a risk-sensitive RL problem, and outlining some potential future research directions.

Markov decision processes (MDPs) [1] provide a general framework for modeling a wide range of problems involving sequential decision making under uncertainty, which arise in many areas of applications, such as transportation, computer/communication systems, manufacturing, and supply chain management. MDPs transition from state to state probabilistically over time due to chosen actions taken by the decision maker, incurring state/action-dependent costs/rewards at each instant. The goal is to find a policy (sequence of decision rules) for choosing actions that optimizes a long-run objective function, e.g., the cumulative sum of discounted costs or the long-run average cost.

The traditional MDP setting assumes that (i) the transition dynamics (probabilities) are fully specified, and (ii) the objective function and constraints involve standard expected value criteria.

∗prashla@cse.iitm.ac.in
†mfu@isr.umd.edu
However, in many settings of practical interest, one or both of these conditions does not hold, i.e., only samples of transitions (and costs/rewards) can be observed (e.g., in a black-box simulation model or an actual system), and/or performance measures that incorporate risk are more appropriate. In the case of the former, reinforcement learning (RL) techniques can be employed \[2, 3, 4\], and in the latter setting, risk-sensitive approaches are appropriate. Although there is abundant research on both of these settings dating back decades, the work combining both aspects is more recent. Furthermore, the two settings have been pursued predominantly independently by different research communities, with RL a focus of CS/AI researchers and risk-sensitive MDPs a focus of stochastic control (and to some extent, operations research/management science) researchers. The main purpose of this article is to review some recent work on policy gradient methods that considers problem formulations that combines both aspects, i.e., reinforcement learning with risk-sensitive criteria, and to outline some promising avenues for future research following the risk-sensitive RL framework. This entry both extends and complements the introductory RL survey Encyclopedia entry, “Reinforcement learning algorithms for MDPs” \[4\] in two ways by addressing risk and employing policy gradient search, neither of which is treated there (e.g., as stated in the last sentence of the Overview section, “Owing to the lack of space, the discussion of (direct) policy search methods is not included.”) In particular, we will focus on constrained formulations, where the traditional expected value performance measure is augmented with a risk constraint. Some well-known examples of risk constraints include the variance (or higher moments), probabilities (i.e., chance constraints), value at risk (VaR), and conditional value at risk (CVaR). We summarize some of our work \[5, 6, 7, 8, 9\] to give a flavor of the challenges involved in solving risk-sensitive MDPs, including some very recent work on using cumulative prospect theory (CPT) \[10\] with RL.

To be specific, in this article, we are primarily concerned with the combination of risk criteria and reinforcement learning in a constrained optimization setting, viz. the following optimization problem:

$$\min_{\theta \in \Theta} J(\theta) \triangleq \mathbb{E}[D(\theta)] \quad \text{subject to} \quad G(\theta) \leq \alpha, \quad (1)$$

where $\theta$ denotes the policy parameter, $\Theta$ represents the policy space, and $D(\theta)$ is an MDP long-run cost. Popular choices for $D(\theta)$ include the infinite-horizon cumulative cost in a discounted cost MDP, total cost in a stochastic shortest path problem, and the long-run average cost. The constraint(s) $G$ generally involves one of the risk measures. The classic “risk-neutral” approach is to find a policy that minimizes $J(\cdot)$, whereas a risk-sensitive approach will consider the risk-constrained formulation in (1). Furthermore, in the setting
above, it is assumed that $J$ and $G$ are unknown or cannot be calculated directly, but samples are available, so that we have at our disposal unbiased or consistent estimators. This contrasts with the traditional setting of risk-sensitive control, well studied in both the operations research (OR) and control communities, where in the context of MDPs, the (risk-sensitive) performance measures corresponding to $J$ and $G$ are analytically available. In our setting, the usual techniques cannot be applied, whereas RL algorithms suitably adapted should provide one avenue to attack such risk-sensitive MDPs, i.e., a setting when the MDP model is unknown and all the information about the system is obtained from samples resulting from the agent’s interaction with the environment.

We chose to focus on this particular constrained formulation, but there are many other possible approaches to incorporate risk, which we briefly review here. As alluded to earlier, various risk measures have been proposed, e.g., mean-variance tradeoff [11], exponential utility [12, 13], the percentile performance [14], the use of chance constraints [15], stochastic dominance constraints [16], value at risk (VaR), and conditional value at risk (CVaR) [17, 18, 19]. CVaR is a popular example of a coherent risk measure [20, 21] that has been extended to multi-period settings (cf. [22, 23, 18, 24, 25, 26]). Coherent risk measures are convex, making them amenable for optimization using stochastic programming techniques, while non-convex risk measures, such as VaR, are hard to optimize. Another non-convex risk measure, which is highly suitable for application in human decision-making systems is prospect theory (PT) [27] and its successor, cumulative prospect theory (CPT) [10]. Unlike expected value and other risk measures, CPT has been shown, through compelling experiments on humans, to model human risk preferences well [28, 10]. Some recent work adopting this approach includes [29, 30, 31, 32]; see also [24].

In the context of MDPs, a large body of literature involves the exponential utility formulation – cf. the classic formulation by Howard and Matheson [13] and some of the recent work in [33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43]. To illustrate the main idea behind the exponential utility formulation, suppose that $D(\theta)$ is the cumulative cost r.v., corresponding to policy $\theta$, in a discounted MDP (the exponential utility community tends to focus on costs rather than rewards, for a reason that becomes clearer shortly). Then, the risk-neutral objective is to minimize $\mathbb{E}[D(\theta)]$ over the set of admissible policies. In contrast, the risk-sensitive approach aims to minimize not just the expected value, but also the higher moments through an exponential cost function defined as $\frac{1}{\beta} \log \mathbb{E}[e^{\beta D(\theta)}]$, where $\beta$ is a parameter that controls risk sensitivity, which approaches the risk-neutral objective $\mathbb{E}[D(\theta)]$ as $\beta$ tends to zero. Moreover, using a Taylor’s series expansion, we can expand $\frac{1}{\beta} \log \mathbb{E}[e^{\beta D(\theta)}] = \mathbb{E}[D(\theta)] + \frac{\beta}{2} \text{Var}[D(\theta)] + O(\beta^2)$; hence, the risk-sensitive cost is closely related to mean-variance optimization. Note that using a constrained formulation, say with
a variance constraint in (1), eschews choosing $\beta$, although its value can be imputed from the choice of $\alpha$, where the latter in many practical applications has a more intuitive meaning. For instance, in a traffic control problem, it would correspond to an upper-bound on the variability of the acceptable delay, or alternatively, one could specify the probability of the delay exceeding an acceptable threshold, e.g., 99% likely that the delay will be less than an hour. For a survey of risk-sensitive RL under the exponential utility formulation, the reader is referred to [44].

**Challenges in risk-sensitive RL**

Risk-sensitive RL is generally more challenging than its risk-neutral counterpart. For instance, in [45], the author showed that, for a discounted MDP, there exists a Bellman equation for the variance of the return, but the underlying Bellman operator is not necessarily monotone, thus ruling out policy iteration as a solution approach for variance-constrained MDPs. In [46], the authors establish that finding a globally mean-variance optimal policy in a discounted MDP is NP-hard, even in the classic MDP setting where the transition model is known. In [47], the authors define, for an average-cost MDP, a risk measure that is not the plain variance of the average cost and instead is a variance of a quantity that measures the deviation of the single-stage cost from the average cost. As expected, solving an average-cost MDP under this notion of variance is NP-hard, as well, as shown in [47]. Finally, in comparison to variance/CVaR, CPT is a non-coherent and non-convex measure, ruling out the usual Bellman equation-based dynamic programming (DP) approaches when optimizing the MDP CPT-value.

The computational complexity results summarized in the previous paragraph imply that finding guaranteed global optima of the the risk-constrained MDP formulated in (1) is not computationally practical, motivating the need for algorithms that approximately solve such MDP formulations. In this article, we consider policy gradient-type learning algorithms where the policies are parameterized (in a continuous space), and an incremental search for a better policy occurs through a gradient-descent update. Actor-critic methods are a popular subclass of policy gradient methods and were among the earliest to be investigated in RL [48, 49]. They comprise of an *Actor* that improves the current policy via gradient descent (as in policy gradient schemes) and a *Critic* that incorporates feature-based representations to approximate the value function. The latter approximation is necessary to handle the curse of dimensionality. Regular policy gradient schemes usually rely on Monte Carlo methods for policy evaluation, an approach that suffers from high variance as compared to actor-critic schemes. On the other hand, function approximation introduces a bias in the policy evaluation. A policy gradient/actor-critic scheme with provable convergence to a locally
risk-optimal policy would require careful synthesis of techniques from stochastic approximation, stochastic gradient estimation approaches, and importance sampling. Due to space limitations, we summarize risk-sensitive policy gradient schemes in this article and omit the actor-critic extensions, which incorporate function approximation.

Several of the constituent solution pieces require significant research for various risk measures. To demonstrate this, consider the “policy evaluation” part of the overall algorithm in a risk-sensitive MDP. The requirement here is to estimate $J(\theta)$ and $G(\theta)$, given samples obtained by simulating the MDP with policy $\theta$. If $J(\theta)$ is one of the usual MDP optimization objectives such as discounted total cost, long-run average cost, or total cost (in a finite-horizon MDP), then estimating $J(\theta)$ can be performed using one of the existing algorithms. Temporal difference (TD) learning [50] is a well-known algorithm that can learn the objective value along a sample path for a given $\theta$. However, estimating $G(\theta)$ using TD-type learning algorithms is infeasible in many cases. For instance, consider variance as the risk measure in a discounted cost MDP. In this case, even though there is a Bellman equation, as shown in [45], the operator underlying this equation is not monotone, ruling out a TD-type learning algorithm. More recently, CVaR-constrained MDPs have been considered in [51, 52, 53], though a variance-reduced CVaR estimation algorithm is still needed.

Going beyond the prediction problem, designing policy gradient schemes is challenging for a risk-sensitive MDP, as it requires estimating the (policy) gradient of the risk measure considered - a non-trivial task in the RL context. For instance, applying a simultaneous perturbation method treating the MDP as a black box would ignore the underlying Markovian structure of the problem. This is the case with the existing actor schemes for mean-variance optimization in discounted MDPs [6] and for optimizing MDP CPT-value [8], where the authors apply SPSA to update the policy parameter in the descent direction.

We present four special cases of the constrained optimization problem posed in (1). The first and second cases concern variance as a risk measure in a discounted and average cost MDP, respectively, and these are based on [5, 6]. The third case involves CVaR as a risk measure, and is based on the work done in [7], while the fourth case considers CPT as the risk measure, and is based on [8, 54]. We provide a brief description of other works on risk-sensitive RL. In [55], the authors consider variance as risk in a stochastic shortest path context, and propose a policy gradient algorithm using the likelihood ratio method. In [56], the authors propose an algorithm that modifies the temporal differences, and show its connection to the exponential utility approach. In [57], the authors provide a general policy gradient algorithm that handles a class of risk measures that includes CVaR. An early work that considers a constrained MDP setup, similar to that in (1), is [58], where the objective is average cost, and the constraint is also an average cost function that is different.
from the one used in the objective. An extension of this setup to a discounted cost MDP, incorporating function approximation, was treated in [59]. CVaR optimization in a constrained MDP setup was also explored in [51], but the algorithm proposed there requires that the single-stage cost be separable. More recently, in [60], the authors consider optimization of risk measures that include CVaR, in an unconstrained MDP setting, and propose reinforcement learning algorithms with function approximation, for solving the problem.

The rest of the paper is organized as follows: Section 1 sets up the MDP notation and outlines the popular variants objectives: discounted/average cost MDPs and stochastic shortest path. Section 2 describes popular risk measures such as variance, CVaR, and CPT. Section 3 presents a template for the risk-sensitive policy gradient algorithm. In Section 4 we provide a brief introduction to temporal difference learning, and two popular gradient estimation techniques. In Section 5 we present a convergence analysis, without proofs, of the template algorithm. In Section 6 we present four special cases that were outlined earlier. Finally, Section 7 provides concluding remarks and identifies a few challenging future research directions.

1 Markov Decision Processes

We consider an MDP with state space $\mathcal{X}$ and action space $\mathcal{A}$, both assumed to be finite. Let $P(\cdot|x,a)$ denote the transition probability distribution and $k(x,a)$ denote the single-stage cost incurred in state $x$ under action $a$. A randomized policy $\mu$ maps any state $x \in \mathcal{X}$ onto a probability distribution over the actions $\mathcal{A}$, i.e., $\mu(\cdot|x)$ is a distribution over actions available in state $x$. We consider randomized policies $\mu^\theta$, where $\theta$ denotes the policy parameter, and assume that (i) the parameter $\theta$ belongs to a compact and convex subset $\Theta$ of $\mathbb{R}^d$; and (ii) any policy with underlying parameter $\theta \in \Theta$ is continuously differentiable. Since a policy is identifiable by its parameter $\theta$, we use them interchangeably. We now formally define risk-neutral objectives in three different MDP settings.

1.1 Stochastic Shortest Path

Here the state space $\mathcal{X}$ contains a cost-free absorbing state, say 0. We shall consider policies that ensure that the state 0 is recurrent and the remaining states transient in the underlying Markov chain. In [61], such policies are referred to as “proper”. An episode is a simulated sample path using a policy $\mu$ that starts in state $x^0 \in \mathcal{X}$, visits $\{x_1, \ldots, x_{\tau-1}\}$ before ending in the absorbing state 0, where $\tau$ is the first passage time to state 0. Let $D(\theta,x^0) = \sum_{m=0}^{\tau-1} k(x_m,a_m)$, denote the total cost from an episode, with the actions $a_m$ chosen
using policy \( \mu^\theta \), i.e., \( a_m \sim \mu^\theta(\cdot | x_m) \). The risk-neutral objective here is to minimize the expected total cost, i.e.,

\[
\min_{\theta \in \Theta} \left\{ J(\theta, x^0) \right\}.
\]

(2)

1.2 Discounted-cost MDP

Let \( 0 < \gamma < 1 \) denote the discount factor. For a given policy \( \theta \), let \( D(\theta, x^0) = \sum_{m=0}^\infty \gamma^m k(x_m, a_m) \) denote the infinite horizon cumulative discounted cost for an MDP trajectory starting in state \( x^0 \), with \( a_m \sim \mu^\theta(\cdot | x_m) \). The risk-neutral objective here is

\[
\min_{\theta \in \Theta} \left\{ J(\theta, x^0) \right\}.
\]

(3)

The expected cost \( J(\theta, x^0) \) is popularly referred to as the value function (cf. [3]) and satisfies the following fixed point equation:

\[
J(\theta, x) = \sum_a \mu^\theta(a|x)k(x, a) + \gamma \sum_{a,x'} \mu^\theta(a|x)P(x'|x, a)J(\theta, x').
\]

(4)

1.3 Average-cost MDP

The average cost under policy \( \theta \) is defined as

\[
J(\theta) \triangleq \lim_{T \to \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{m=0}^{T-1} k(x_m, a_m) \right],
\]

(5)

where the actions \( a_m \) above are chosen according to policy \( \theta \), i.e., \( a_m \sim \mu^\theta(\cdot | x_m) \). The goal in the standard (risk-neutral) average cost formulation is

\[
\min_{\theta \in \Theta} J(\theta).
\]

With every policy \( \mu^\theta \), we associate an expected differential value function, which is defined as follows:

\[
Q(\theta, x, a) = \sum_{n=0}^\infty \mathbb{E}[k(x_n, a_n) - J(\theta) \mid x_0 = x, a_0 = a, \theta],
\]

(6)

\[
V(\theta, x) = \sum_a \mu^\theta(a|x)Q(\theta, x, a).
\]

(7)

The functions defined above satisfy the following Poisson equations [11]:

\[
J(\theta) + V(\theta, x) = \sum_a \mu^\theta(a|x)k(x, a) + \sum_{x'} P(x'|x, a)V(\theta, x'),
\]

(8)

\[
J(\theta) + Q(\theta, x, a) = k(x, a) + \sum_{x'} P(x'|x, a)V(\theta, x').
\]

(9)
2 Risk Measures

2.1 Variance in discounted-cost MDPs

Sobel [45] introduced the following measure of variability in a discounted-cost MDP:

\[ G(\theta, x) \triangleq U(\theta, x) - J(\theta, x)^2, \quad \text{where} \quad U(\theta, x) \triangleq \mathbb{E}[D(\theta, x)^2]. \] (10)

The risk measure above is the overall variance of the cumulative discounted cost. An alternative is to consider per-period variance, i.e., the deviations of the single stage costs, and we shall introduce such a measure in the average cost MDP (see Section 2.2). Setting aside the question of which is the most appropriate notion of variability for the underlying MDP, we shall design algorithms for overall variance in the discounted case and per-period variance in the average cost case. The constituent pieces of these algorithms (see Sections 6.1–6.2) can be reused to handle the case where one considers per-period variance in a discounted cost MDP and overall variance in an average cost MDP.

A fixed point equation for \( G(\theta, x) \) is derived in [45]. However, as shown there, the operator underlying this equation for variance lacks the monotonicity property. Interestingly, one can derive a fixed point equation for the square value function \( U \), and the operator underlying this equation is a contraction mapping. The variance can then be estimated using \( U \) and \( J \), and estimation of the latter quantities is facilitated through TD-type learning algorithms. This approach was introduced in an SSP context in [62], and later extended to the discounted MDP context in [6]. The fixed point equation \( U(\theta, x) \) is given below.

\[
U(\theta, x) = \sum_a \mu(\theta|x)k(x, a)^2 + \gamma^2 \sum_{a, x'} \mu(\theta|x)P(x'|x, a)U(\theta, x') + 2\gamma \sum_{a, x'} \mu(\theta|x)P(x'|x, a)k(x, a)J(\theta, x').
\] (11)

Let \( T^\theta = [T^\theta_1; T^\theta_2] \) denote the operator underlying equations (4) and (11), when viewed together as a function over 2 \(|X|\) variables, where \( T^\theta_1 \) and \( T^\theta_2 \) denote the operators underlying the fixed point equations for the value and square-value functions, respectively, with underlying policy parameter \( \theta \). These operators are defined as: For any \((J, U) \in \mathbb{R}^{2|X|}\), where \( J \) and \( U \) denote the first and last \(|X|\) entries, respectively, define

\[
T^\theta(J, U) = [T^\theta_1(J, U); T^\theta_2(J, U)], \quad \text{where}
\]

\[
T^\theta_1(J, U) = k^\theta + \gamma P^\theta J, \quad \text{and} \quad T^\theta_2(J, U) = K^\theta k^\theta + 2\gamma K^\theta P^\theta J + \gamma^2 P^\theta U.
\] (12) (13)
In the above, $k^\theta$ is a $|X|$ vector of single-stage costs for each state, $K^\theta$ is a $|X| \times |X|$ matrix with the entries of $k^\theta$ along the diagonal and zeroes elsewhere, and $P^\theta$ is the transition probability matrix of the Markov chain underlying policy $\mu^\theta$. It can be shown that $T^\theta$ is a contraction mapping (cf. Lemma 2 in [6]).

For the risk-sensitive policy gradient algorithm presented later, we also require a counterpart of $U(\cdot)$ with initial state-action pair $(x,a)$ under policy $\theta$ and this quantity is defined as

$$W(\theta,x,a) \triangleq \mathbb{E}\left[ D(\theta,x,a)^2 \mid x_0 = x, a_0 = a, \theta \right],$$

where $D(\theta,x,a) = \sum_{n=0}^{\infty} \gamma^n k(x_n,a_n)$. As in the case of $U$, the function $W$ also satisfies a fixed point equation, which is given below.

$$W(\theta,x,a) = k(x,a)^2 + \gamma^2 \sum_{x'} P(x'|x,a)U(\theta,x') + 2\gamma k(x,a) \sum_{x'} P(x'|x,a)J(\theta,x').$$

### 2.2 Variance in average-cost MDPs

Unlike the discounted-cost MDP, the variance, as defined by [47], measures the deviations of single-stage cost from the average cost (and not the variance of the average cost itself). More precisely,

$$G(\theta) \triangleq \lim_{T \to \infty} \frac{1}{T} \mathbb{E}\left[ \sum_{n=0}^{T-1} (k(x_n,a_n) - J(\theta))^2 \mid \theta \right],$$

(14)

where the actions $a_n$ are governed by policy $\theta$. To see the rationale behind the definition above for variability, consider two stream of cost: a policy $\theta_1$ results in $(0,0,0,0,\ldots)$, while another policy $\theta_2$ gives $(100,-100,100,-100,\ldots)$. The average cost as well as the variance of the average cost is zero for both policies. On the other hand, from the point of the variance as defined by (14), policy $\theta_1$ is better than $\theta_2$.

A straightforward calculation yields

$$G(\theta) = \eta(\theta) - J(\theta)^2,$$

where $\eta(\theta) = \sum_{x,a} \pi(\theta,x,a)k(x,a)^2$ is the average square cost of policy $\theta$, with $\pi(\theta,x,a)$ denoting the stationary distribution of the state-action pair $(x,a)$ under policy $\mu^\theta$.

Along the lines of (8), we have the following Poisson equations for $U$ and $W$, which denote the respective square cost counterparts of $Q$ and $V$ in (6):

$$\eta(\theta) + U(\theta,x) = \sum_a \mu^\theta(a|x) [k(x,a)^2 + \sum_{x'} P(x'|x,a)U(\theta,x')]$$

$$\eta(\theta) + W(\theta,x,a) = k(x,a)^2 + \sum_{x'} P(x'|x,a)U(\theta,x').$$

(15)
2.3 Conditional Value-at-Risk

For any random variable $X$, the Value-at-Risk (VaR) at level $\beta \in (0, 1)$ is defined as

$$\text{VaR}_\beta(X) := \inf \{ \xi \mid P(X \leq \xi) \geq \beta \},$$

which mathematically is just a quantile, since if $F$ is the c.d.f. of $X$, VaR is equivalently defined as

$$\text{VaR}_\beta(X) := \inf \{ \xi \mid F(\xi) \geq \beta \} = F^{-1}(\beta).$$

VaR is a commonly used risk measure in the financial industry, where it represents a level of assets needed to cover a potential loss. VaR as a risk measure has several drawbacks, which precludes using standard stochastic optimization methods; most prominently, VaR is not a coherent risk measure [20]. A risk measure is coherent if it is sub-additive, monotone, positive homogeneous and translation invariant. On the other hand, another closely related risk measure also widely used in the financial industry called CVaR is coherent and thus lends itself to stochastic programming techniques [17]. CVaR is a conditional mean over the tail distribution as delineated by the VaR, defined as follows:

$$\text{CVaR}_\beta(X) := \mathbb{E}[X \mid X \geq \text{VaR}_\beta(X)].$$

In a stochastic shortest path problem, CVaR of a policy $\theta$ with underlying parameter $\theta$ is defined as:

$$G(\theta, x^0) \triangleq \text{CVaR}_\beta \left[ \sum_{m=0}^{\tau-1} k(x_m, a_m) \mid x_0 = x^0 \right], \tag{16}$$

where $\tau$ is the first visiting time to state 0 and $a_m \sim \mu^\theta(\cdot \mid x_m)$.

2.4 Cumulative prospect theory (CPT)

CPT is a risk measure that captures human attitudes towards risk. For any r.v. $X$, the CPT-value is defined as

$$C(X) \triangleq \int_0^\infty w^+ \left( \mathbb{P}(u^+(X) > z) \right) dz - \int_0^\infty w^- \left( \mathbb{P}(u^-(X) > z) \right) dz, \tag{17}$$

where $u^+, u^- : \mathbb{R} \rightarrow \mathbb{R}_+$, as shown in Figure 1(a), are the utility functions that are assumed to be continuous, with $u^+(x) = 0$ when $x \leq 0$ and increasing otherwise, and with $u^-(x) = 0$ when $x \geq 0$ and decreasing.
(a) An example of a utility function. A reference point on the $x$ axis serves as the point of separating gains and losses. For losses, the disutility $-u^-$ is typically convex and for gains, the utility $u^+$ is typically concave; both functions are non-decreasing and take the value of zero at the reference point.

(b) An example of a weight function. A typical CPT weight function inflates small, and deflates large probabilities, capturing the tendency of humans doing the same when faced with decisions of uncertain outcomes.

otherwise, $w^+, w^- : [0, 1] \to [0, 1]$, as illustrated in Figure 1(b) are weight functions assumed to be continuous, non-decreasing and satisfy $w^+(0) = w^-(0) = 0$ and $w^+(1) = w^-(1) = 1$.

CPT-value is a generalization of the classic expected value, and this can be seen as follows: Consider the case when $w^+, w^-$ are identity functions, $u^+(x) = x$ for $x \geq 0$ and 0 otherwise, and $u^-(x) = -x$ for $x \leq 0$ and 0 otherwise. Then, we have

$$C(X) = \int_0^\infty P(X > z) \, dz - \int_0^\infty P(-X > z) \, dz = E[(X)^+] - E[(X)^-],$$

where $(a)^+ = \max(a, 0)$, $(a)^- = \max(-a, 0)$.

The human preference to play safe with gains and take risks with losses is captured by a concave gain-utility $u^+$ and a convex disutility $-u^-$. The weight functions $w^+, w^-$ capture the idea that the value seen by a human subject is nonlinear in the underlying probabilities. In particular, humans deflate high probabilities
and inflate low probabilities. A risk measure based on CPT in a typical MDP setting could apply the CPT-functional to a risk-neutral objective. For instance, set the r.v. $X$ in (17) to either the total cost in a SSP problem or the infinite-horizon cumulative cost in a discounted MDP. However, to carry out dynamic programming would require a Bellman optimality equation, which seems formidable, given the non-convex structure of the CPT-value. A different approach is to use a nested formulation, together with CPT-style probability distortion, an approach adopted by Lin in his PhD dissertation; see also. Basically, the formulation is equivalent to optimizing the sum of CPT-value period costs rather than the CPT-value of the sum, and by doing so guarantees the existence of a Bellman optimality equation.

3 Risk-sensitive Policy Gradient: A Template

Recall that we are given the following problem:

$$\min_{\theta \in \Theta} J(\theta) \quad \text{subject to} \quad G(\theta) \leq \alpha,$$

where $J$ is the usual risk-neutral MDP objective, while the constraint $G$ involves one of the risk measures presented earlier. In an average cost formulation, the objective/constraint do not depend on the initial state, while they do in total cost formulations such as SSP and discounted problems. In either case, the template for solving the problem remains the same, and to keep the presentation simple, we have chosen to have only the policy parameter in $J$ and $G$. In the special cases of Section we shall include the initial state as necessary.

If there is a policy in $\Theta$ that satisfies the constraint in (18), then using Theorem 3.8 in [64], it can be inferred that there exists an optimal policy, which uses at most one randomization.

Using the Lagrangian approach, we consider the following relaxed MDP problem:

$$\max_{\lambda} \min_{\theta} \left( L(\theta, \lambda) \triangleq J(\theta) + \lambda (G(\theta) - \alpha) \right),$$

where $\lambda$ is the Lagrange multiplier. The goal here is to find the saddle point of $L(\theta, \lambda)$, i.e., a point $(\theta^*, \lambda^*)$ that satisfies

$$L(\theta, \lambda^*) \geq L(\theta^*, \lambda^*) \geq L(\theta^*, \lambda), \forall \theta \in \Theta, \forall \lambda > 0.$$ 

For a standard convex optimization problem where the objective $L(\theta, \lambda)$ is convex in $\theta$ and concave in $\lambda$, one can ensure the existence of a unique saddle point under mild regularity conditions (cf. [65]). Further,
convergence to this point can be achieved by descending in $\theta$ and ascending in $\lambda$ using $\nabla_\theta L(\theta, \lambda)$ and $\nabla_\lambda L(\theta, \lambda)$, respectively.

However, in the risk-sensitive RL setting, the Lagrangian $L(\theta, \lambda)$ is not necessarily convex in $\theta$, which implies there may not be an unique saddle point. The problem is further complicated by the fact that we operate in a simulation optimization setting, i.e., only sample estimates of the Lagrangian are obtained. Hence, performing primal descent and dual ascent, one can only get to a local saddle point, i.e., a point $(\theta^*, \lambda^*)$ that is a local minima w.r.t. $\theta$, and local maxima w.r.t $\lambda$ of the Lagrangian. As an aside, global mean-variance optimization of MDPs has been shown to be NP-hard in [46], so the best one can hope is to find an approximately optimal policy.

For the purpose of finding an optimal risk-sensitive policy, a standard procedure would update the policy parameter $\theta$ and Lagrange multiplier $\lambda$ in two nested loops: an inner loop that descends in $\theta$ using the gradient of the Lagrangian $L(\theta, \lambda)$ w.r.t. $\theta$, and an outer loop that ascends in $\lambda$ using the gradient of the Lagrangian $L(\theta, \lambda)$ w.r.t. $\lambda$.

We operate in a setting where we only observe simulated costs of the underlying MDP. Thus, it is required to estimate both $J$ and $G$ for a given $\theta$ and then use these estimates to compute an estimate of the gradient of the Lagrangian w.r.t. $\theta$ and $\lambda$. The gradient $\nabla_\lambda L(\theta, \lambda)$ has a particularly simple form of $(G(\theta) - \alpha)$, suggesting that a sample of the risk measure can be used to perform the dual ascent for Lagrange multiplier $\lambda$. On the other hand, the policy gradient $\nabla_\theta L(\theta, \lambda) = \nabla J(\theta) + \lambda \nabla G(\theta)$ is usually complicated and does not lend itself to stochastic programming techniques in a straightforward fashion. We shall address the topic of gradient estimation in the next section, but for presenting the template of the risk-sensitive policy gradient algorithm, suppose that we are given estimates $\hat{\nabla} J(\theta)$, $\hat{\nabla} G(\theta)$ and $\hat{G}(\theta)$ of $\nabla J(\theta)$, $\nabla G(\theta)$ and $G(\theta)$, respectively. Then, using two-timescale stochastic approximation [66, Chapter 6], the inner and outer loops mentioned above can run in parallel, as follows:

\begin{align}
\theta_{n+1} &= \Gamma \left[ \theta_n - \zeta_2(n) \left( \hat{\nabla} J(\theta_n) + \lambda_n \hat{\nabla} G(\theta_n) \right) \right], \\
\lambda_{n+1} &= \Gamma_\lambda \left[ \lambda_n + \zeta_1(n) \left( \hat{G}(\theta_n) - \alpha \right) \right].
\end{align}

In the above, $\Gamma$ is a projection operator that keeps the iterate $\theta_n$ stable by projecting onto a compact and convex set $\Theta := \prod_{i=1}^d [\theta_{\text{min}}^{(i)}, \theta_{\text{max}}^{(i)}]$. In particular, for any $\theta \in \mathbb{R}^d$, $\Gamma(\theta) = (\Gamma^{(1)}(\theta^{(1)}), \ldots, \Gamma^{(d)}(\theta^{(d)}))^T$, with $\Gamma^{(i)}(\theta^{(i)}) := \min(\max(\theta_{\text{min}}^{(i)}, \theta^{(i)}), \theta_{\text{max}}^{(i)})$; $\Gamma_\lambda$ is a projection operator that keeps the Lagrange multiplier $\lambda_n$ within the interval $[0, \lambda_{\text{max}}]$, for some large positive constant $\lambda_{\text{max}} < \infty$ and can be defined in an analogous
Using policy $\mu_n$, simulate the underlying MDP

Using policy $\mu_n$, simulate the underlying MDP

Estimate $\nabla_{\theta} J(\theta)$

Policy Gradient

Estimate $\nabla_{\theta} G(\theta)$

Risk Gradient

Estimate $G(\theta)$

Risk Estimation

Update $\theta_n$ using (20)

Update $\lambda_n$ using (21)

Policy Update

$\theta_n, \lambda_n$ $\rightarrow$ $\theta_n+1, \lambda_{n+1}$

Figure 1: Overall flow of risk-sensitive policy gradient algorithm.

fashion as $\Gamma$; and $\zeta_1(n), \zeta_2(n)$ are step-sizes selected such that the $\theta$ update is on the faster timescale, and the $\lambda$ update is on the slower timescale (see (A4) below).

Two timescale algorithms have been employed in a RL context in the context of actor-critic algorithms (cf. [67], [58], [68], [59]). Several simulation-based optimization algorithms (cf. the textbook [69] which has a detailed description of stochastic optimization algorithms using the simultaneous perturbation method) use the two timescale stochastic approximation approach as well. In both settings, the analysis of the various timescales conform to the view that the faster timescale recursions see the iterates on slower timescale as quasi-static, while the slower timescale recursions see the iterates on faster timescale as equilibrated, which in essence is equivalent to assuming slower timescale iterates as constant while analysing faster timescale recursions and using converged values of faster timescale iterates for analysis of slower timescale recursions.

As illustrated in Figure 1, the template for a risk-sensitive policy gradient algorithm would involve the following items:

1. a two-timescale update rule for the policy parameter and Lagrange multiplier;

2. estimates of the objective $J(\cdot)$ and the risk measure $G(\cdot)$, which can be obtained by sampling from the underlying MDP with the current policy parameter, and then using a suitable estimation scheme,
usually based on stochastic approximation (in RL, this would be equivalent to some form of TD-learning), or based on Monte Carlo averaging; the estimate of the objective would feed into estimating the policy gradient (see step below), while the estimate of the risk measure is necessary for the gradient of the Lagrangian, as well as for the dual ascent procedure; and,

3. estimates of the gradients $\nabla F(\cdot)$ and $\nabla G(\cdot)$ for primal descent - this may be quite challenging if the underlying risk measure has no structure that can be exploited in an MDP framework.

We postpone the convergence analysis of the policy update algorithm in (20)–(21) to Section 5 and cover the necessary background material in the next section, which is necessary for understanding the special cases presented in Section 6, as well as for the convergence analysis of the algorithm.

4 Background

For the four special cases considered in Section 6, TD-learning and gradient estimation using two popular approaches serve as building blocks. In this section, we briefly cover these topics, and the reader is referred to [2] and [61] for a textbook introduction.

4.1 Stochastic approximation (SA)

The goal of stochastic approximation (SA) is to solve the equation $F(\theta) = 0$ when an analytical form of $F$ is not known but noisy measurements $F(\theta_n) + \xi_n$ can be obtained, where $\theta_n, n \geq 0$ is the input parameter, and $\xi_n, n \geq 0$ is a zero-mean r.v. More generally, $\xi_n$ could be a martingale difference sequence.

The seminal Robbins-Monro algorithm [70] solved this problem by employing the following iterative update rule:

$$
\theta_{n+1} = \theta_n + \zeta(n)(F(\theta_n) + \xi_n),
$$

where $\{\zeta(n)\}$ is a step-size sequence that satisfies $\sum_n \zeta(n) = \infty$ and $\sum_n \zeta(n)^2 < \infty$. Let $\mathcal{F}_n = \sigma(\theta_m, m \leq n)$ denote the underlying sigma-field. Then, assuming $F$ is Lipschitz, iterates are stable, i.e., $\sup_n |\theta_n| < \infty$, and the noise satisfies $\mathbb{E}[|\xi_{n+1}|^2 | \mathcal{F}_n] \leq K(1 + |\theta_n|^2)$ for some $K > 0$, it can be shown that $\theta_n$ governed by (22) converges to the solution of $F(\theta) = 0$, e.g., Theorem 2 in Chapter 2 of [66]; see also [71].

SA is useful in solving several subproblems in risk-sensitive RL. For example, TD-learning is an instance of an SA algorithm that incorporates a fixed-point iteration. While regular TD-learning is useful in...
estimating $J(\theta)$, a variant will be useful in estimating variance indirectly (see Section 6.1). Moreover, VaR estimation is performed using an SA scheme that features a stochastic gradient descent-type update iteration, while CVaR estimation is a plain averaging rule that can be done through SA, as well.

4.2 Temporal-difference (TD) learning

A key algorithm for policy evaluation in RL is TD learning [50], where the objective is to estimate the value function $J(\theta, x)$ for a given policy $\mu^\theta$. For simplicity, we use the TD(0) algorithm and consider the discounted cost MDP setting. Starting with any $J_0$, the TD(0) algorithm updates an estimate $J_{n+1}$ at time instant $n + 1$ using the observed sample cost $k(x_n, a_n)$ ($a_n \sim \mu^\theta(\cdot \mid x_n)$) and previous estimate $J_n$ as follows:

$$J_{n+1}(x) = J_n(x) + \zeta(n) \nu(x, n) \mathbb{I}\{x_n = x\} (k(x_n, a_n) + \gamma J_n(x_{n+1}) - J_n(x_n)), \quad (23)$$

where $\nu(x, n) = \sum_{m=0}^{n} \mathbb{I}\{x_m = x\}$ and $x_{n+1}$ is a r.v. sampled from $P(\cdot \mid x_n, a_n)$, and $\zeta(\cdot)$ is a step-size. TD(0) can be shown to converge to the true value $J(\theta, x^0)$ almost surely if the step-size decreases according to the standard stochastic approximation conditions.

However, the approach above employs full state representations, i.e., to have a lookup table entry for each state $x \in \mathcal{X}$, and would be subject to the curse of dimensionality. The latter refers to the exponential blow up of the computation with an increase in size of the state space. A standard trick to overcome this problem is to employ feature-based representations and function approximation, by approximating the value function in a discounted cost MDP setting as follows: $J(\theta, x) \approx v^T \phi(x)$, where $\phi(x)$ is a $d$-dimensional feature (column) vector corresponding to the state $x$, with $d \ll |\mathcal{X}|$ and $v$ is a $d$-dimensional parameter that is updated as follows:

$$v_{n+1} = v_n + \zeta(n) \phi(x_n) (k(x_n, a_n) + \gamma v_n^T \phi(x_{n+1}) - v_n^T \phi(x_n)), \quad (24)$$

where $\theta_0$ is set arbitrarily, $a_n \sim \mu(\cdot \mid x_n)$ and $\{\zeta(n)\}$ is a step-size sequence satisfying standard SA conditions. In [72], the authors establish that $v_n$ asymptotically converges to the fixed point of the projected Bellman equation, given as follows: $\Phi v = \Pi T^\mu(\Phi v)$, where $T^\mu$ is the Bellman operator corresponding to the policy $\mu$, $\Pi$ is the orthogonal projection onto the linearly parameterized space $\{\Phi v \mid v \in \mathbb{R}^d\}$, with $\Phi$ denoting the feature matrix with rows $\phi(x)^T, \forall x \in \mathcal{X}$.
TD-learning can be employed to estimate the differential value function \( V(\theta, x) \) in an average cost MDP setting, with the Poisson equation in (8) as the basis \[73\]. Notice that the latter equation contains the average cost \( J(\theta) \), which has to be estimated from sample data, and then plugged into the TD-learning update rule for estimating \( V \). The TD(0) variant in this case, with underlying policy \( \mu^\theta \), would update as follows:

\[
V_{n+1}(x) = V_n(x) + \zeta(\nu(x, n)) \mathbb{1}\{x_n = x\} \left( k(x_n, a_n) - \hat{J}_n + V_n(x_{n+1}) - V_n(x_n) \right),
\]

(24)

where \( \hat{J}_n \) is the average of the sample single-stage costs seen up to time instant \( n \). The function approximation variant of TD in the average cost setting can be worked out using arguments similar to that in the discounted cost setting.

### 4.3 Simultaneous perturbation stochastic approximation (SPSA)

Suppose we want to solve

\[
\min_\theta (F(\theta) = \mathbb{E}[f(\theta, \xi)]),
\]

where \( \xi \) denotes zero-mean i.i.d. noise. For this stochastic optimization problem, the Kiefer-Wolfowitz algorithm performs gradient descent as follows:

\[
\theta_{n+1} = \theta_n - \zeta(n) \hat{\nabla} f(\theta_n), \quad \text{where, for } i = 1, \ldots, d,
\]

\[
\hat{\nabla}_i f(\theta_n) = \left( \frac{f(\theta_n + \delta_n e_i, \xi_{n,i}^+)}{2\delta_n} - \frac{f(\theta_n - \delta_n e_i, \xi_{n,i}^-)}{2\delta_n} \right),
\]

(25)

where \( \{\zeta(n)\} \) is a step-size sequence satisfying standard SA conditions, \( \delta_n \) are positive perturbation constants, \( \xi_{n,i}^+, \xi_{n,i}^- \) are i.i.d. zero-mean noise components, and \( e_i \) is the unit vector with 1 in the \( i \)th place and 0’s elsewhere. It can be shown that \( \hat{\nabla}_i f(\theta) \) approaches \( \nabla_i f(\theta) \) if \( \delta_n \to 0 \) as \( n \to \infty \). However, the disadvantage with this approach is that at any update epoch \( n \), one requires \( 2N \) samples of the objective function \( f(\cdot, \cdot) \). On the other hand, the seminal SPSA algorithm \[74\] requires only two samples of the objective function in each iteration, for any dimension \( d \), and estimates the gradients as follows:

\[
\hat{\nabla}_i f(\theta_n) = \left( \frac{f(\theta_n + \delta_n \Delta(n), \xi_{n,i}^+)}{2\delta_n \Delta_i(n)} - f(\theta_n - \delta_n \Delta(n), \xi_{n,i}^-) \right),
\]

(25)

where \( \xi_{n,i}^+, \xi_{n,i}^- \) are i.i.d. noise as before and \( \Delta(n) = (\Delta_1(n), \ldots, \Delta_d(n))^T \) is a random perturbation vector, with each \( \Delta_i(n) \), chosen to be symmetric \( \pm 1 \)-valued Bernoulli r.v.s. It can be shown that the SPSA gradient estimate is only an order \( O(\delta_n^2) \) term away from the true gradient and hence, asymptotically unbiased if \( \delta_n \to 0 \). The convergence analysis of this algorithm can be seen, for instance, in \[74\],\[69, Theorem 5.1\].
A variant of (25) is to use an one-sided estimate, i.e., given sample observations at \(\theta_n + \delta_n \Delta(n)\) and \(\theta_n\), form the gradient estimate as follows:

\[
\hat{\nabla}_i f(\theta_n) = \left( \frac{f(\theta_n + \delta_n \Delta(n), \xi_n^+ - f(\theta_n, \xi_n))}{\delta_n \Delta_i(n)} \right),
\]

(26)

where \(\delta_n\) and \(\Delta(n)\) are as defined earlier. For solving constrained optimization problems, one-sided estimates are efficient, since a sample observation at the unperturbed value of the underlying parameter is necessary for performing the dual ascent on the Lagrange multiplier. The overall SPSA-based policy gradient algorithm would estimate the necessary gradient, as well as the risk measure, using two sample observations corresponding to \(\theta_n + \delta_n \Delta(n)\) and \(\theta_n\). On the other hand, using a balanced estimate, as defined in (25) would require an additional observation with the underlying parameter set to \(\theta_n - \delta_n \Delta(n)\).

### 4.4 Direct single-run gradient estimation using the likelihood ratio method

When the system is a complete black box, then SPSA is an effective way to carry out gradient-based policy optimization. However, in many settings, more is known about the system, and more efficient direct gradient estimation techniques may be applicable, where “direct” means that the gradient estimator is unbiased (as opposed to asymptotically unbiased when finite difference methods such as SPSA are used). The main approaches are perturbation analysis, the likelihood ratio method (also known as the score function method), and weak derivatives (also known as measure-valued differentiation); for overviews of such techniques, see [75, 76] and references therein.

Consider a Markov chain \(\{X_n\}\) with a single recurrent state 0, transient states 1, \ldots, \(r\), and (one-step) transition probability matrix \(P(\theta) := [p_{i,j}(\theta)]_{i,j=0}^r\), where \(p_{i,j}(\theta)\) denotes the probability of going from state \(X_n = i\) to \(X_{n+1} = j\) and is parameterized by \(\theta\). Let \(\tau\) denote the first passage time to the recurrent state 0 and \(X := (X_0, \ldots, X_{\tau-1})^T\) denote the corresponding sequence of states (sample path). Assuming \(\theta\) occurs only in the transition probabilities (as opposed to appearing directly in the sample performance \(f\) itself), i.e., \(F(\theta) := E_{\theta}[f(X)]\), an unbiased single-run sample path likelihood ratio gradient estimate for \(\nabla_\theta F(\theta)\) is given by

\[
f(X) \sum_{m=0}^{\tau-1} \frac{\nabla_\theta p_{X_m,X_{m+1}}(\theta)}{p_{X_m,X_{m+1}}(\theta)}.
\]

It should be noted that if \(\theta\) is a common parameter that appears in all of the probabilities, then this estimator will have variance that increases linearly with the sample path length.
5 Convergence Analysis

In this section, we analyze the convergence properties of the two timescale algorithm in (20)–(21), and can be skipped without loss of continuity.

We make the following assumptions for the sake of analysis of the two timescale recursions in (20)–(21). Let \( \mathcal{F}_n = \sigma(\theta_m, m < n) \) denote the underlying sigma-field.

(A1) The policy \( \mu^0(\cdot | x, a) \) is a continuously differentiable function of \( \theta \), for any \( x \in \mathcal{X} \) and \( a \in \mathcal{A} \).

(A2) The estimate of the risk measure \( \hat{G}(\cdot) \) satisfies \( \mathbb{E} \left( \frac{\hat{G}(\theta_n)}{ \mathcal{F}_n} \right) = G(\theta_n) \).

(A3) The gradient estimates \( \hat{\nabla}J(\theta_n) \) and \( \hat{\nabla}G(\theta_n) \) satisfy
\[
\mathbb{E} \left( \hat{\nabla}J(\theta_n) \mid \mathcal{F}_n \right) = \nabla J(\theta_n), \quad \mathbb{E} \left( \hat{\nabla}G(\theta_n) \mid \mathcal{F}_n \right) = \nabla G(\theta_n), \quad \text{and}
\mathbb{E} \left\| \hat{\nabla}J(\theta_n) \right\|^2 + \mathbb{E} \left\| \hat{\nabla}G(\theta_n) \right\|^2 < \infty.
\]

(A3') The gradient estimates \( \hat{\nabla}J(\theta_n) \) and \( \hat{\nabla}G(\theta_n) \) satisfy
\[
\mathbb{E} \left| \hat{\nabla}J_i(\theta_n) \mid \mathcal{F}_n \right| = O(\delta_n^2), \quad \text{and} \quad \mathbb{E} \left| \hat{\nabla}G_i(\theta_n) \mid \mathcal{F}_n \right| = O(\delta_n^2), \quad \text{for } i = 1, \ldots, |\mathcal{X}|,
\]
and
\[
\mathbb{E} \left( \left\| \hat{\nabla}J(\theta_n) \right\|^2 + \mathbb{E} \left\| \hat{\nabla}G(\theta_n) \right\|^2 \right) < \frac{C}{\delta_n^2},
\]
where \( \delta_n > 0 \) can be chosen arbitrarily.

(A4) The stepsize sequence \( \{\zeta_1(n), \zeta_2(n)\} \) satisfies
\[
\sum_n \zeta_1(n) = \sum_n \zeta_2(n) = \infty, \quad \sum_n (\zeta_1(n)^2 + \zeta_2(n)^2) < \infty, \quad \text{and} \quad \zeta_1(n) = o(\zeta_2(n)).
\]

(A4') \( \delta_n \to 0 \) as \( n \to \infty \), and the stepsize sequence \( \{\zeta_1(n), \zeta_2(n)\} \) satisfies
\[
\sum_n \zeta_1(n) = \sum_n \zeta_2(n) = \infty, \quad \sum_n \left( \zeta_1(n)^2 + \left( \frac{\zeta_2(n)}{\delta_n} \right)^2 \right) < \infty, \quad \text{and} \quad \zeta_1(n) = o(\zeta_2(n)).
\]

Assumption (A1) is a standard requirement in the analysis of policy gradient-type RL algorithms. Assumption (A2) is an unbiasedness requirement on the risk measure estimate that is used for dual ascent in (21). Assumption (A2)-(A2') are unbiasedness requirements on the estimates of the gradient of the objective and risk measure, and are necessary to ensure that the \( \theta \)-recursion in (20) is descending in the Lagrangian objective. Assumption (A3) requires that the gradient estimates are perfectly unbiased, and also that the variance of the gradient estimates are bounded - standard requirements in the analysis of stochastic gradient schemes. The likelihood ratio method, described in Section 4.4 satisfies (A3). Assumption (A3') is a relaxed variant of (A3), where the gradient estimates are asymptotically unbiased, i.e., the gradient estimates
have a parameter $\delta_n$ which can be used to control the bias-variance tradeoff - lower $\delta_n$ results in lower gradient estimation bias but higher variance, and vice-versa. The SPSA technique for gradient estimation, presented in Section 4.3 as well as the general class of simultaneous perturbation schemes, meet the conditions in (A3'). For convergence, $\delta_n$ has to vanish asymptotically, but not too fast, as outlined in the second part of (A4’). Note that the condition requiring bounded variance of the gradient estimates in (A3)/(A3’) is satisfied for both likelihood ratio and SPSA-based methods, if the underlying MDP has finite state and action spaces.

Assumption (A4) is required for the standard two-timescale view, i.e., the policy recursion in (20) views the Lagrange multiplier as quasi-static, while the Lagrange multiplier recursion views the policy parameter as almost equilibrated. Two timescale updates are convenient because both policy and Lagrange multiplier can be updated in parallel, albeit with varying step-sizes. The latter are chosen carefully so that one is able to mimic a two-loop behavior, with policy updates in the inner loop and Lagrange multiplier updates in the outer loop. The reader is referred to Chapter 6 of [66] for a detailed introduction to two timescale stochastic approximation. Assumption (A4’) is a variant of (A4) that has to be coupled with (A3’), in the sense that the gradient estimates have a $O(\delta_n^2)$ bias, but $\sum_n \left( \frac{\zeta_2(n)}{\delta_n} \right)^2 < \infty$.

We adopt the ordinary differential equation (ODE) approach for analyzing the template algorithm in (20)–(21). In particular, under the assumptions listed above, the ODE governing the policy update, for any given Lagrange multiplier $\lambda$, is given by

$$\dot{\theta}(t) = \tilde{\Gamma} \left( \nabla J(\theta(t)) + \lambda \nabla G(\theta(t)) \right), \quad (27)$$

where $\tilde{\Gamma}(\cdot)$ is a projection operator that ensures the evolution of $\theta$ via the ODE (27) stays within the set $\Theta$ and is defined as follows: For any bounded continuous function $f(\cdot)$,

$$\tilde{\Gamma}(f(\theta)) = \lim_{\tau \to 0} \frac{\Gamma(\theta + \tau f(\theta)) - \theta}{\tau}. \quad (28)$$

**Remark 1 (Two timescale view)** In describing the ODE governing the policy recursion, we have assumed that the Lagrange multiplier is constant, and this view can be justified as follows: First rewrite the $\lambda$-recursion as

$$\lambda_{n+1} = \Gamma\lambda \left[ \lambda_n + \zeta_2(n) \left( \frac{\zeta_1(n)}{\zeta_2(n)} (G(\theta_n) - \alpha + \varsigma_n) \right) \right],$$

where $\varsigma_n$ is a martingale difference sequence (a consequence of (A1)). Considering that we have a finite dimensional MDP setting, together with the fact that $\frac{\zeta_1(n)}{\zeta_2(n)} = o(1)$ (see (A4)), it is clear that the $\lambda$-recursion above tracks the ODE $\dot{\lambda}(t) = 0$. 

20
The claim that the \( \lambda \)-recursion views the policy parameter as almost equilibrated requires a more sophisticated argument, and the reader is referred to Theorem 2 in Chapter 6 of \cite{66}.

**Theorem 1** Assume that (A1), (A2), (A3) + (A4) or (A3') + (A4') hold. With \( \lambda_n \equiv \lambda \), for \( \theta_n \) governed by (20), we have

\[
\theta_n \rightarrow Z_\lambda \ a.s. \ as \ n \rightarrow \infty,
\]

where \( Z_\lambda = \{ \theta \in \Theta : \hat{\Gamma}(\nabla J(\theta(t)) + \lambda \nabla G(\theta(t))) = 0 \} \) is the set of limit points of the ODE (27).

The proof involves an application of the well-known Kushner-Clark lemma \cite{77} for projected stochastic approximation, in particular, verifying conditions A5.3.1-2, and A5.1.3-5, so that Theorem 5.3.1 there can be invoked.

We now turn to the analysis of \( \lambda \)-recursion in (21). The ODE underlying the Lagrange multiplier is given below.

\[
\dot{\lambda}(t) = \hat{\Gamma}_\lambda [G(\theta_{\lambda(t)}) - \alpha], \tag{29}
\]

where \( \theta_{\lambda(t)} \) is the converged value of the \( \theta \)-recursion, when the Lagrange multiplier is set to \( \lambda(t) \). The operator \( \hat{\Gamma}_\lambda \) is similar to \( \hat{\Gamma} \), and ensures that the \( \lambda \)-recursion stays within \([0, \lambda_{\text{max}}]\).

**Theorem 2** Assume that (A1), (A2), (A3) + (A4) or (A3') + (A4') hold. Then, for \( \lambda_n \) governed by (21), we have

\[
\lambda_n \rightarrow Z \ a.s. \ as \ n \rightarrow \infty,
\]

where \( Z = \{ \lambda \in [0, \lambda_{\text{max}}] : \hat{\Gamma}_\lambda (G(\theta_\lambda) - \alpha) = 0, \theta_\lambda \in Z_\lambda \} \) is the set of limit points of the ODE (29).

The proof follows by using arguments similar to those employed in the proof of Theorem 2 in Chapter 2 of \cite{66}.

Thus, we have shown that \((\theta_n, \lambda_n)\) converges to \((\theta_{*, \lambda_*}, \lambda_{*, \lambda_*})\), for some \( \lambda_* \in Z \) and \( \theta_{*, \lambda_*} \in Z_{\lambda_{*, \lambda_*}} \). However, it is important to know if \((\theta_{*, \lambda_*}, \lambda_*^*)\) satisfies the constraint, and for this purpose, one invokes the envelope theorem of mathematical economics \cite{78}. In particular, the latter theorem leads to the conclusion that the limit \( \theta_{*, \lambda_*} \) satisfies \( G(\theta_{*, \lambda_*}) \leq \alpha \), and we omit the technical details of this theorem application. The reader is referred to \cite{58} or \cite{6} for further details.

### 6 Special Cases

Recall that the main ingredients in each iteration \( n \) of the risk-sensitive RL algorithm are as follows:
(I) Simulation of the underlying MDP. For the case of non-perturbation-based approaches, such as the likelihood ratio method, one simulation with policy \( \theta_n \) would suffice. On the other hand, for SPSA-based approaches, an additional simulation using a perturbed policy parameter would be necessary (see Section 4.3).

(II) Estimation of \( \nabla_{\theta} J(\theta_n) \) and \( \nabla_{\theta} G(\theta_n) \). These estimates are fed into the primal descent update for \( \theta_n \).

(III) Estimation of \( G(\theta_n) \) using sample data. This estimate is used for dual ascent.

(IV) Estimation of \( J(\theta_n) \) using sample data. This estimate is used for primal descent. Note that in the case of SPSA, we would require estimate of \( J \) for the perturbed policy as well, while additional function estimates are not necessary using the likelihood ratio method.

In the four special cases that we discuss in detail below, we shall address the items above in a variety of MDP contexts, under mean-variance, CVaR and CPT risk measures. In particular, we consider the following combinations for the objective \( J \) and risk measure \( G \) in (I):

| Case | MDP type                          | Objective \( J \)                        | Constraint \( G \)   |
|------|-----------------------------------|----------------------------------------|---------------------|
| 1    | Discounted cost                   | Cumulative cost (see (3))              | Overall variance (see (10)) |
| 2    | Average cost                      | Average cost (see (5))                 | Per-period variance (see (14)) |
| 3    | Stochastic shortest path          | Total cost (see (2))                   | CVaR (see (16))      |
| 4    | Discounted cost/ SSP              | Total/discounted cost                  | CPT-value (see (17)) |

The previous section presented the necessary background material on the TD algorithm and gradient estimation using two popular approaches, which serve as building blocks for the four special cases. Since we cover a lot of ground and also owing to space limitations, the presentation will be aimed at an advanced RL researcher who is familiar with the theory of risk-neutral RL and is seeking an understanding of the challenges in the risk-sensitive RL setting. The necessary references are provided for the interested reader to pursue any of the special cases in detail.

### 6.1 Case 1: Discounted-cost MDP + Variance as risk

We consider the following constrained problem: For a given \( \alpha > 0 \),

\[
\min_{\theta} J(\theta, x^0) \quad \text{subject to} \quad G(\theta, x^0) \leq \alpha,
\]
where $J(\theta, x^0)$ and $G(\theta, x^0)$ are the expectation and variance of the cumulative cost r.v., respectively (see Sections 1.2 and 2.1). Letting $L(\theta, \lambda) \triangleq J(\theta, x^0) + \lambda(G(\theta, x^0) - \alpha)$, the necessary gradients of the Lagrangian are given by

$$
\nabla_\theta L(\theta, \lambda) = \nabla_\theta J(\theta, x^0) + \lambda \nabla_\theta G(\theta, x^0) = \nabla_\theta J(\theta, x^0) + \lambda \left( \nabla_\theta U(\theta, x^0) - 2J(\theta, x^0)\nabla_\theta J(\theta, x^0) \right),
$$

$$
\nabla_\lambda L(\theta, \lambda) = G(\theta, x^0) - \alpha.
$$

From the expressions above, it is apparent that $\nabla_\theta J(\theta, x^0)$ and $\nabla_\theta U(\theta, x^0)$ are the necessary gradients and these quantities are given below.

$$
\nabla_\theta J(\theta, x^0) = \sum_{x,a} \pi_\theta(x,a|x^0) \nabla_\theta \log \mu^\theta(a|x) Q^\theta(x,a), \quad (30)
$$

$$
\nabla_\theta U(\theta, x^0) = \sum_{x,a} \tilde{\pi}_\theta(x,a|x^0) \nabla_\theta \log \mu^\theta(a|x) W^\theta(x,a) + 2\gamma \sum_{x,a,x'} \tilde{\pi}_\theta(x,a|x^0) P(x'|x,a) k(x,a) \nabla_\theta J(\theta, x'), \quad (31)
$$

where $Q^\theta(x,a) = \mathbb{E}[D^\theta(\theta, x, a) \mid x, a, \theta]$ is the Q-value function,

$\pi_\theta(x,a|x^0) = \sum_{n=0}^{\infty} \gamma^n \Pr(x_n = x|x_0 = x^0; \theta) \mu^\theta(a|x)$, and

$\tilde{\pi}_\theta(x,a|x^0) = \sum_{n=0}^{\infty} \gamma^{2n} \Pr(x_n = x|x_0 = x^0; \theta) \mu^\theta(a|x)$ are the $\gamma$ and $\gamma^2$-discounted visiting distributions, respectively, of the state-action pair $(x,a)$ under policy $\mu^\theta$.

Figure 2: Overall flow of SPSA-based risk-sensitive policy gradient algorithm in a discounted cost MDP setting.

Estimating $\nabla_\theta J(\theta, x^0)$ and $\nabla_\theta U(\theta, x^0)$ is challenging due to the following reasons:

I) Two different sampling distributions are used for $\nabla_\theta J(\theta, x^0)$ and $\nabla_\theta U(\theta, x^0)$. In particular, the distributions $\pi_\theta$ and $\tilde{\pi}_\theta$ involve factors $\gamma$ and $\gamma^2$, respectively.
\( \nabla J(\theta, x') \) appears in the second summation on the RHS of (31), and this makes the estimation task hard in practice, as one needs an estimate of the gradient of the value function \( J(\theta, x') \) at every state \( x' \) of the MDP, and not just at the initial state \( x^0 \).

To overcome these issues, the authors in [6] use SPSA to estimate \( \nabla J(\theta,x^0) \) and \( \nabla U(\theta,x^0) \). As illustrated in Figure 2, such an estimation scheme requires running two trajectories corresponding to policy parameters \( \theta_n + \delta_n \Delta(n) \) (where \( \delta_n \) and \( \Delta(n) \) are described in Section 4.3), and \( \theta_n \), respectively. The samples from the trajectories would be used to estimate \( J(\theta_n + \delta_n \Delta(n), x^0) \), \( J(\theta_n, x^0) \), \( U(\theta_n + \delta_n \Delta(n), x^0) \), and \( U(\theta_n, x^0) \), which in turn help in forming the estimates of the gradient of \( J(\theta, x^0) \) and \( U(\theta, x^0) \) as follows: For \( i = 1, \ldots, \|X\| \),

\[
\hat{\nabla}_i J(\theta_n, x^0) = \frac{\hat{J}(\theta_n + \delta_n \Delta(n), x^0) - \hat{J}(\theta_n, x^0)}{\delta_n \Delta_i(n)}, \text{ and } \hat{\nabla}_i U(\theta_n, x^0) = \frac{\hat{U}(\theta_n + \delta_n \Delta(n), x^0) - \hat{U}(\theta_n, x^0)}{\delta_n \Delta_i(n)}.
\]

(32)

In the above, \( \hat{J}(\theta, x^0) \) (resp. \( \hat{U}(\theta, x^0) \)) denotes an estimate of \( J(\theta, x^0) \) (resp. \( U(\theta, x^0) \)), for any \( \theta \in \Theta \). The task of estimating \( J \) is straightforward, and the regular TD algorithm, as described earlier in Section 4.2, can be employed.

Recall that \( U \) satisfies a fixed point equation (see (11)), and the operator underlying this equation is well-behaved in the sense that one gets a contraction mapping that is amenable for stochastic approximation, while estimating variance directly would not help, because the underlying operator is not monotone [45]. Thus, we have the following TD-type update for estimating \( U \):

\[
U_{n+1}(x) = U_n(x) + \zeta(\nu(x,n)) \mathbb{I}\{x_n = x\} \left( k(x_n, a_n)^2 + 2\gamma k(x_n, a_n) J_n(x_{n+1}) + \gamma^2 U_n(x_{n+1}) - U_n(x_n) \right),
\]

(33)

where \( \nu(x,n) = \sum_{m=0}^{n} \mathbb{I}\{x_m = x\} \) and \( x_{n+1} \) is a r.v. sampled from \( P(:|x_n, a_n) \). Notice that the factor \( J \) goes into the fixed point equation for \( U \), and hence, the TD algorithm for \( U \) has to employ the TD-based estimate of \( J \) for estimating \( U \).

Algorithm [1] presents the pseudocode for the risk-sensitive policy-gradient algorithm for the discounted cost setting. In a nutshell, this algorithm uses multi-timescale stochastic approximation [66, Chapter 6] to perform the following tasks: (i) run the TD algorithm on the fastest timescale to estimate both \( J \) and \( U \); (ii) use an SPSA-based gradient descent scheme on the intermediate timescale for solving the primal problem in (1); and (iii) perform dual ascent on the Lagrange multiplier using the sample variance constraint (using
estimate of $U$) on the slowest timescale. The latter two timescale updates follow the template provided in Section 3.

**Algorithm 1:** Policy gradient algorithm under variance as a risk measure in a discounted cost MDP setting

| Input | initial parameter $\theta_0 \in \Theta$, where $\Theta$ is a compact and convex subset of $\mathbb{R}^d$, perturbation constants $\delta_n > 0$, trajectory lengths $\{m_n\}$, step-sizes $\{\zeta_1(n), \zeta_2(n)\}$, projection operators $\Gamma$ and $\Gamma_\lambda$, number of iterations $M \gg 1$. |
|-------|-------------------------------------------------------------------------------------------------|
| 1     | for $n \leftarrow 0$ to $M - 1$ do                                                            |
| 2     | Generate $\Delta(n)$ using symmetric $\pm 1$-valued Bernoulli distribution;                     |
| 3     | for $m \leftarrow 0$ to $m_n - 1$ do                                                          |
| 4     | /* Unperturbed policy simulation */                                                            |
| 5     | Use the policy $\mu^{\theta_n}$ to generate the state $x_m$, draw action $a_m \sim \mu^{\theta_n}(\cdot | x_m)$; |
| 6     | Observe next state $x_{m+1}$ and cost $k(x_m, a_m)$;                                           |
| 7     | Use (23) and (33) to estimate $J(\theta_n)$ and $G(\theta_n)$;                               |
| 8     | /* Perturbed policy simulation */                                                             |
| 9     | Use the policy $\mu^{\theta_n+\delta_n \Delta(n)}$ to generate the state $x^+_m$, draw action $a^+_m \sim \mu^{\theta_n+\delta_n \Delta(n)}(\cdot | x^+_m)$; |
| 10    | Observe next state $x^+_{m+1}$ and cost $k(x^+_m, a^+_m)$;                                    |
| 11    | Use (23) and (33) to estimate $J(\theta_n + \delta_n \Delta(n))$ and $G(\theta_n + \delta_n \Delta(n))$; |
| 12    | end                                                                                           |
| 13    | /* Gradient estimates using SPSA */                                                             |
| 14    | Gradient of the objective: $\hat{\nabla}_i J(\theta_n, x^0) = \frac{\hat{J}(\theta_n + \delta_n \Delta(n), x^0) - \hat{J}(\theta_n, x^0)}{\delta_n \Delta_i(n)}$; |
| 15    | Gradient of the constraint: $\hat{\nabla}_i U(\theta_n, x^0) = \frac{\hat{U}(\theta_n + \delta_n \Delta(n), x^0) - \hat{U}(\theta_n, x^0)}{\delta_n \Delta_i(n)}$; |
| 16    | /* Policy update: Gradient descent using SPSA */                                               |
| 17    | $\theta_{n+1} = \Gamma [\theta_n - \zeta_2(n) \left( \hat{\nabla}_i J(\theta_n, x^0) + \lambda_n \left( \hat{\nabla}_\theta U(\theta_n, x^0) - 2 \hat{J}(\theta_n, x^0) \hat{\nabla}_\theta J(\theta_n, x^0) \right) \right)]$; |
| 18    | /* Lagrange multiplier update: Dual ascent using sample variance (formed using TD) */         |
| 19    | $\lambda_{n+1} = \Gamma_\lambda \left[ \lambda_n + \zeta_1(n) \left( \hat{U}(\theta_n, x^0) - 2 \hat{J}(\theta_n, x^0) - \alpha \right) \right]$; |
| 20    | Output: Policy $\theta_M$                                                                     |

**On the number of samples $m_n$ per iteration**

To understand the challenge in choosing an appropriate $m_n$, so that the overall algorithm converges, consider a simpler setting of optimizing a smooth function $f$, i.e.,

$$\text{find } \theta^* = \arg \min_{\theta \in \Theta} f(\theta),$$
where \( \Theta \) is a convex and compact subset of \( \mathbb{R}^d \). In many simulation optimization settings, as illustrated in Figure 3, one has an oracle that supplies noisy function measurements, but the noise is usually zero mean. On the other hand, in typical RL settings, the function \( f \) is to be estimated from sample observations. For instance, the function \( f \) could be the value function \( J \), and a popular estimation scheme in this setup is TD-learning. In this setting, one could simulate a longer trajectory of the underlying system to obtain more accurate estimates of \( J \). Thus, we have a setting where the error in function estimates can be controlled and made very low, albeit at the cost of additional simulations. Notice a similar consideration on the trajectory lengths to simulate for the purpose of CVaR estimation, as well as CPT-value estimation. In both cases, the estimation procedure (see (38) and (39)) is asymptotically unbiased, but one does not have the luxury of having a very long run of the policy evaluation procedure, considering that the outer loop of incremental policy update needs to perform policy evaluation often.

A stochastic gradient-descent scheme to solve the problem above would update as follows:

\[
\theta_{n+1} = \Gamma \left( \theta_n - \gamma_n \hat{\nabla} f(\theta_n) \right),
\]

where \( \{\gamma_n\} \) is a step-size sequence that satisfies standard stochastic approximation conditions, \( \Gamma = (\Gamma_1, \ldots, \Gamma_d) \) is an operator that ensures that the update (34) stays bounded within the compact and convex set \( \Theta \), and \( \hat{\nabla} f(\theta_n) \) is an estimate of the gradient of \( f \) at \( \theta_n \). Suppose that the gradient estimate is formed using SPSA, as described in Section 4.3, i.e., \( \frac{\hat{f}(\theta_n + \delta_n \Delta(n)) - \hat{f}(\theta_n)}{\delta_n \Delta(n)^t} \), where \( \hat{f}(\theta) \) denotes the estimate of \( f \), when the underlying parameter is \( \theta \). Suppose that the estimation scheme returns \( \hat{f}(\theta_n) = f(\theta_n) + \psi_\theta^n \), where \( \psi_\theta^n \) denotes the error in estimating the objective \( f \) using \( m_n \) function measurements. These estimates could be formed using TD-learning, when the function \( f \) is the value or square value function, or using Monte Carlo averages, when \( f \) is the CPT-value or CVaR (see sections below). For the sake of this discussion, suppose that the estimation errors vanish at rate \( \frac{1}{\sqrt{m_n}} \).

We first rewrite the update rule in (34) as follows:

\[
\theta'_{n+1} = \Gamma_i \left( \theta'_n - \gamma_n \left( \frac{f(\theta_n + \delta_n \Delta(n)) - f(\theta_n)}{\delta_n \Delta(n)^t} + \kappa_n \right) \right),
\]
where $\kappa_n = \frac{(\psi_{\theta n}^{\alpha} + \delta_n \Delta(n)) - \psi_{\theta n}^{\alpha}}{\delta_n \Delta(n)}$. Let $\zeta_n = \sum_{t=0}^n \gamma_t \kappa_t$. Then, a critical requirement that allows us to ignore the estimation error term $\zeta_n$ is the following condition (see Lemma 1 in Chapter 2 of [66]):

$$\sup_{t \geq 0} (\zeta_{n+t} - \zeta_n) \to 0 \text{ as } n \to \infty.$$ Notice that the estimation error $\varphi^\theta$ is a function of number of samples $m_n$ used for estimating the objective value, and it is obviously necessary to increase the number of samples $m_n$ so that the bias vanishes asymptotically. In addition to the usual conditions on the step-size sequence and perturbation constant $\delta_n$, one possible choice for $m_n$ that ensures that the bias in the gradient estimate vanishes and the overall algorithm converges is the following: $\frac{1}{m_n^{1/2} \delta_n} \to 0$.

In the case of value and square value functions, using TD-learning with $m_n$ samples, the estimation error vanishes at rate $\frac{1}{\sqrt{m_n}}$. On the other hand, CPT-value estimation, presented later in Section 6.4, has an estimation error of order $\frac{1}{m_n^{\alpha/2}}$, for some $\alpha \in (0, 1]$, and the condition on $m_n$ for ensuring convergence of the overall stochastic gradient scheme would be $\frac{1}{m_n^{\alpha/2} \delta_n} \to 0$.

**Remark 2** (Extension to incorporate function approximation) In [6], the authors parameterize both $J$ and $U$ using linear function approximation and then employ TD-type schemes for policy evaluation. Notice that both $J(\cdot)$ and $U(\cdot)$ need to be evaluated for the perturbed policies. Let $J(x) \approx v^T \phi_v(x)$ and $U(x) \approx u^T \phi_u(x)$ be the linear approximations to $J$ and $U$, respectively, with features $\phi_v(\cdot)$ and $\phi_u(\cdot)$ from low-dimensional spaces. It can be shown that an appropriate operator can be defined for $U$ using the above equation and an operator that projects orthogonally onto the linear space $\{\Phi_u u \mid u \in \mathbb{R}^d\}$. Such a projected Bellman operator turns out to be a contraction mapping (see Lemma 2 in [6] for a proof) and hence, a TD-type scheme can be arrived at, along the lines of that for the regular cost $J$. Such an approach was proposed in [62] for an SSP setting and extended to include discounted problems in [6]. We omit the update rules, and the reader is referred to [6] for further details.

### 6.2 Case 2: Average cost MDP + Variance as risk

We consider the following constrained optimization problem for average cost MDPs:

$$\min_{\theta} J(\theta) \quad \text{subject to} \quad G(\theta) \leq \alpha,$$

where $J(\theta)$ is the long-run average cost and $G(\theta)$ is the variance, as defined in Sections 1.3 and 2.2. Letting $L(\theta, \lambda) \triangleq J(\theta) + \lambda (G(\theta) - \alpha)$, and noting that $\nabla G(\theta) = \nabla \eta(\theta) - 2 J(\theta) \nabla J(\theta)$, it is apparent that $\nabla J(\theta)$
and $\nabla \eta(\theta)$ are enough to calculate the necessary gradients of the Lagrangian. Let $U^\theta$ and $W^\theta$ denote the differential value and action-value functions associated with the square cost under policy $\mu^\theta$, respectively. These two quantities satisfy the following Poisson equations:

$$\eta(\theta) + U(\theta, x) = \sum_a \mu^\theta(a|x)[k(x, a)^2 + \sum_{x'} P(x'|x, a)U(\theta, x')],$$

$$\eta(\theta) + W(\theta, x, a) = k(x, a)^2 + \sum_{x'} P(x'|x, a)U(\theta, x').$$

As mentioned earlier, we consider finite state-action space MDPs, which together with an irreducibility assumption implies the existence of a stationary distribution for the Markov chain underlying any policy $\theta$. Denote by $d^\theta(x)$ and $\pi^\theta(x, a) = d^\mu(x)\mu(a|x)$, the stationary distribution of state $x$ and state-action pair $(x, a)$ under policy $\mu$, respectively. We now present the gradients of $J(\theta)$ and $\eta(\theta)$ below.

$$\nabla J(\theta) = \sum_{x, a} \pi^\theta(x, a) \nabla \log \mu^\theta(a|x)Q(\theta, x, a), \quad (35)$$

$$\nabla \eta(\theta) = \sum_{x, a} \pi^\theta(x, a) \nabla \log \mu^\theta(a|x)W(\theta, x, a), \quad (36)$$

where $Q(\theta, x, a) = \sum_{n=0}^{\infty} \mathbb{E}[k(x_n, a_n) - J(\theta) \mid x_0 = x, a_0 = a, \mu]$, with actions $a_n \sim \mu^\theta(\cdot \mid x_n)$.

The above relation holds owing to the fact that we parameterize the policies, and hence, the gradient of the transition probabilities can be estimated from the policy alone. This is the well-known policy gradient technique [79] that makes it amenable for estimating the gradient of a performance measure in MDPs, since the transition probabilities are not required and one can work with policies and simulated transitions from the MDP.

An important observation concerning $\nabla J(\theta)$ is that any function $b : \mathcal{X} \to \mathbb{R}$ can be added or subtracted to $Q(\theta, x, a)$ on the RHS of (35), and the resulting summation stays as $\nabla J(\theta)$. In a risk-neutral setting, a popular choice is to replace $Q(\theta, x, a)$ with the advantage function $A(\theta, x, a) = Q(\theta, x, a) - V(\theta, x)$. In a typical RL setting, $\nabla J(\theta)$ has to be estimated, and from the discussion before, this implies estimation of the advantage function using samples – TD-learning is a straightforward choice for this task. Using the expression on the RHS of (35), one can arrive at a decrement factor for the policy update as follows: substitute a TD-based empirical approximation to the advantage function, calculate the likelihood ratio $\nabla \log \mu^\theta(\cdot \mid x_n)$, and perform a gradient descent using the product of the advantage estimate with the likelihood ratio, and we arrive at an empirical approximation to the RHS of (35) with the advantage function $A$ instead of $Q$ there.
The idea described above, i.e., to use the advantage function in place of $Q$, can be used for the case of $\nabla \eta(\theta)$ as well, with the advantage function variant $B(\theta, x, a) = W(\theta, x, a) - U(x; \theta)$ on the RHS of (26). The resulting risk-sensitive policy update bears a close resemblance to the risk-neutral counterpart in the policy recursion. The pseudocode of the overall algorithm in the average reward setting is given in Algorithm 2.

Algorithm 2: Policy gradient algorithm under variance as a risk measure in an average cost MDP setting

```
Input: initial parameter $\theta_0 \in \Theta$, where $\Theta$ is a compact and convex subset of $\mathbb{R}^d$, step-sizes $\{\zeta_1(n), \zeta_2(n), \zeta_3(n), \zeta_4(n)\}$, projection operators $\Gamma$ and $\Gamma_\lambda$, number of iterations $M \gg 1$.

for $n \leftarrow 0$ to $M - 1$ do
    Draw action $a_m \sim \mu^\theta_{n+1}(\cdot|x_m)$, observe next state $x_{m+1}$ and cost $k(x_m, a_m)$;
    /* Estimate for average cost */
    $J_{n+1} = (1 - \zeta_4(n)) J_n + \zeta_4(n) k(x_n, a_n)$;
    /* Estimate for average square cost */
    $\eta_{n+1} = (1 - \zeta_4(n)) \eta_n + \zeta_4(n) k(x_n, a_n)^2$;
    /* TD critic for the value function */
    $\delta_n = k(x_n, a_n) - J_{n+1} + V_n(x_{n+1}) - V_n(x_n)$;
    $V_{n+1} = V_n + \zeta_3(n) \delta_n$;
    /* TD critic for the square value function */
    $\epsilon_n = k(x_n, a_n)^2 - \eta_{n+1} + U_n(x_{n+1}) - U_n(x_n)$;
    $U_{n+1} = U_n + \zeta_3(n) \epsilon_n$;
    Set $\psi_n = \nabla \log \mu^\theta_{n+1}(a_n|x_n)$; // Likelihood ratio
    /* Policy update */
    $\theta_{n+1} = \Gamma \left( \theta_n - \zeta_2(n) ( - \delta_n \psi_n + \lambda_n (\epsilon_n \psi_n - 2 J_{n+1} \delta_n \psi_n) ) \right)$;
    /* Lagrange multiplier update */
    $\lambda_{n+1} = \Gamma_\lambda \left( \lambda_n + \zeta_1(n) (\eta_{n+1} - J_{n+1}^2 - \alpha) \right)$;
end

Output: Policy $\theta_M$
```

In addition to the step-size requirements in (A4), we require that $\zeta_2(n) = o(\zeta_3(n))$ and $\zeta_4(n)$ is a constant multiple of $\zeta_3(n)$. Such choices ensure that the TD-critic and average cost updates are on the fastest timescale, the policy update is on an intermediate timescale, and the Lagrange multiplier update is on the slowest timescale.

Remark 3 The variance notion employed in this section involved measuring the deviations of the single-stage cost from its average. As we demonstrated in Algorithm 2 the per-period variance as a risk measure lends itself to policy gradient techniques well, since the likelihood ratio method can be employed to solve the risk-constrained problem. In contrast, the variance notion in the discounted cost setting involved the
variance of the cumulative discounted cost, i.e., the (overall) variance of the underlying r.v. and not the per-period one. Such a measure is hard to optimize (see discussion below (31)), though SPSA could be employed. The flip side to the latter approach is that we do not exploit the structure of the underlying problem in forming the gradient estimates, e.g., using the likelihood ratio method. More importantly, SPSA requires simulation of two independent trajectories (corresponding to unperturbed and perturbed policy parameters), and this may not be feasible in many practical applications.

One could consider swapping the risk measures of discounted and average cost settings, i.e., employ per-period variance in a discounted cost MDP, and overall variance in the average cost MDP. Leaving the question of which is the best risk measure for a given MDP aside, we believe that such a swap of risk measures would make solving the average cost problem difficult, and discounted cost problem easy in comparison.

**Remark 4** As in the discounted setting, incorporating function approximation for the functions \( J \) and \( U \) is straightforward, and we refer the reader to [6] for the case where a linear function approximation architecture is used.

### 6.3 Case 3: Stochastic shortest path + CVaR as risk

We again consider the following constrained optimization problem:

\[
\min_{\theta} J(\theta, x^0) \quad \text{subject to} \quad G(\theta, x^0) \leq \alpha,
\]

where \( J(\theta, x^0) \) and \( G(\theta, x^0) \) are the expectation and CVaR of the total cost r.v. \( D^0(x^0) \), respectively (see Sections 1.1 and 2.3). With the Lagrangian \( L(\theta, \lambda) \triangleq J(\theta, x^0) + \lambda(G(\theta, x^0) - \alpha) \), the necessary gradients for solving the constrained problem above are \( \nabla_{\theta} J(\theta, x^0) \) and \( \nabla_{\theta} \text{CVaR}_\beta(D^0(x^0)) \). Using the likelihood ratio method (cf. [79]), the first gradient is obtained as follows:

\[
\nabla_{\theta} J(\theta, x^0) = \mathbb{E} \left[ \left( \sum_{n=0}^{\tau-1} k(x_n, a_n) \right) \sum_{m=0}^{\tau-1} \nabla \log \mu^\theta(a_m | x_m) \bigg| x^0 \right].
\]

For estimating the gradient of the CVaR of \( D^0(x^0) \) for a given policy parameter \( \theta \), we employ the following likelihood ratio gradient estimate proposed by [80]:
\[
\n\nabla_\theta \text{CVaR}_\beta(D^\theta(x^0)) = \mathbb{E} \left[ D^\theta(s^0) - \text{VaR}_\alpha(D^\theta(s^0)) \right] \sum_{m=0}^{\tau-1} \nabla \log \mu^\theta(a_m | x_m) \left| D^\theta(x^0) \geq \text{VaR}_\beta(D^\theta(x^0)) \right] .
\]

What remains to be specified is the technique employed for estimating \text{VaR} and \text{CVaR} for a given policy \( \theta \). Notice that \text{CVaR} estimation is required for dual ascent, since \( \nabla_\lambda L(\theta, \lambda) = \text{CVaR}_\beta(D^\theta(x^0)) - \alpha \). \text{VaR} is required for estimating \text{CVaR} and the \text{CVaR} gradient. A well-known result in [17] is that both \text{VaR} and \text{CVaR} can be obtained from the solution of a certain convex optimization problem. More precisely, for any r.v. \( X \), let

\[
\nu(\xi, X) := \xi + \frac{1}{1-\beta} (X - \xi)_+ \text{ and } V(\xi) = \mathbb{E} [\nu(\xi, X)] .
\]

Then, \( \text{VaR}_\beta(X) \) is the minimizer of \( V \), i.e., a point \( \xi^*_\beta \) that satisfies \( V'(\xi^*_\beta) = 0 \) and \( \text{CVaR}_\beta(X) = V(\xi^*_\beta) \).

Since \( \nu(\xi, \cdot) \) is continuous w.r.t. \( \xi \), \( V'(\xi) = \mathbb{E} \left( 1 - \frac{1}{1-\beta} \mathbb{I} \{ X \geq \xi \} \right) \). The minimizer \( \xi^* \) would be a \text{VaR}, and \( V(\xi^*) \) would be the \text{CVaR} of the r.v. \( X \). Observing that \( V \) is convex, a stochastic approximation-based procedure can be derived for estimating \text{VaR} and \text{CVaR}, an approach suggested in [81] and specialized to MDPs in [7]. In an SSP context, the r.v. is \( D^\theta(x^0) \). Suppose that we can obtain i.i.d. samples from the distribution of \( D^\theta(x^0) \), i.e., we can simulate the underlying SSP using the policy \( \theta \). Let \( D_k, k = 1, \ldots \) denote these samples. Then, \text{VaR} and \text{CVaR} can be estimated as follows:

\[
\text{VaR: } \xi_m = \xi_{m-1} - \zeta_3(m) \left( 1 - \frac{1}{1-\beta} \mathbb{I} \{ D_k \geq \xi_m \} \right) , \tag{37}
\]

\[
\text{CVaR: } C_m = C_{m-1} - \frac{1}{m} (C_{m-1} - \nu(\xi_{m-1}, C_{m-1})) . \tag{38}
\]

In the above, (37) can be seen as a gradient descent rule, while (38) can be seen as a plain averaging update. The step-size sequence \( \{\zeta_3(m)\} \) is required to satisfy standard stochastic approximation conditions, i.e., \( \sum_m \zeta_3(m) = \infty \), and \( \sum_m \zeta_3(m)^2 < \infty \).

The complete algorithm along with the update rules for various parameters is presented in Algorithm 3.
Algorithm 3: Policy gradient algorithm under CVaR as a risk measure in an SSP setting

**Input**: initial parameter \( \theta_0 \in \Theta \), where \( \Theta \) is a compact and convex subset of \( \mathbb{R}^d \), \( \beta \in (0, 1) \), trajectory lengths \( \{m_n\} \), step-sizes \( \{\zeta_1(n), \zeta_2(n), \zeta_3(n)\} \), projection operators \( \Gamma \) and \( \Gamma_\lambda \), number of iterations \( M \gg 1 \).

1. for \( n \leftarrow 0 \) to \( M - 1 \) do
2.   for \( m \leftarrow 0 \) to \( m_n - 1 \) do
3.     Simulate the SSP for an episode to generate the state sequence \( \{X_{n,j}\} \) using actions \( \{a_{n,j} \sim \mu^{\theta_n}(\cdot | X_{n,j})\} \). Let \( \tau_m \) denote the time instant when state 0 was visited in this episode;
4.     Observe total cost \( D_{n,m} = \sum_{j=0}^{\tau_m-1} k(X_{n,j}, a_{n,j}) \);
5.     Calculate likelihood ratio: \( \psi_{n,m} = \sum_{j=0}^{\tau_m-1} \nabla \log \mu^{\theta_n}(x_{n,j}, a_{n,j}) \);
6.   end
7. /* Policy evaluation */
8. Use the scheme in (37)-(38) to obtain the VaR estimate \( \xi_n \) and CVaR\( \beta \)-estimate \( C_n \);
9. Total cost estimate: \( \bar{D}_n = \frac{1}{m_n} \sum_{j=1}^{m_n} D_{n,j} \);
10. Likelihood ratio: \( \bar{\psi}_n = \frac{1}{m_n} \sum_{j=1}^{m_n} \psi_{n,j} \);
11. /* Gradient of the objective */
12. \( \hat{\nabla} J(\theta_n) = \bar{D}_n \bar{\psi}_n \);
13. /* Gradient of the risk measure */
14. \( \hat{\nabla} \text{CVaR}_\beta(\theta_n) = (C_n - \xi_n) \bar{\psi}_n \mathbb{1}\{C_n \geq \xi_n\} \);
15. /* Policy and Lagrange Multiplier Update */
16. \( \theta_{n+1} = \Gamma \left( \theta_n - \zeta_2(n) \left( \hat{\nabla} J(\theta_n) + \lambda_n \hat{\nabla} \text{CVaR}_\beta(\theta_n) \right) \right) \);
17. \( \lambda_{n+1} = \Gamma_\lambda \left( \lambda_n + \zeta_1(n)(C_n - \alpha) \right) \);
18. end

**Output**: Policy \( \theta_M \)

6.4 Case 4: Discounted cost/SSP + CPT as risk

The last case consider CPT as a risk measure in the constrained optimization problem (1). From the discussion in the previous sections, it is apparent that the technical challenges in handling any risk measure are:

(i) estimation of the risk measure from samples; and
(ii) gradient estimation for the policy update iteration.

For the sake of brevity, we provide the necessary details for handling (i) and (ii), and the rest of the pieces of the resulting actor-critic scheme follows in a manner similar to that for variance or CVaR.

To handle (i), suppose that we are given \( n \) i.i.d. samples from the distribution of \( X \), and the goal
is to estimate the CPT-value $C(X)$. Estimating the CPT-value is challenging, because the environment provides samples from the distribution of the r.v. $X$, while the integrals in $\{17\}$ involve a weight-distorted distribution. Thus, unlike the case of expected value estimation, it is necessary to have an estimate of the entire distribution to estimate $C(X)$, and a natural candidate to estimate the distribution is the empirical distribution function (EDF). Using the latter, we estimate $C(X)$ by

$$
\overline{C}_n = \int_0^\infty w^+ \left( 1 - \hat{F}_n^+ (x) \right) dx - \int_0^\infty w^- \left( 1 - \hat{F}_n^- (x) \right) dx.
$$

(39)

The first and the second integral on the RHS above (denoted by $\overline{C}_n^+$ and $\overline{C}_n^-$, respectively) can be computed in a straightforward fashion using the order statistics $X_{[1]} \leq X_{[2]} \leq \ldots \leq X_{[n]}$ as follows:

\[
\begin{align*}
\overline{C}_n^+ &:= \sum_{i=1}^n u^+ (X_{[i]}) \left( w^+ \left( \frac{n + 1 - i}{n} \right) - w^+ \left( \frac{n - i}{n} \right) \right), \\
\overline{C}_n^- &:= \sum_{i=1}^n u^- (X_{[i]}) \left( w^- \left( \frac{i}{n} \right) - w^- \left( \frac{i - 1}{n} \right) \right).
\end{align*}
\]

As far as handling point (ii) concerning the policy gradient for CPT, the authors in $[8]$ use SPSA, since the CPT-value does not admit a Bellman equation, ruling out a procedure based on the likelihood ratio method. The SPSA-based estimate of $\nabla C(X_\theta^n)$ with policy $\theta_n$, is given as follows: For any $i = 1, \ldots, d$,

$$
\hat{\nabla}_i C(X_\theta^n) = \frac{\overline{C}_{\theta_n + \delta_n \Delta(n)} - \overline{C}_{\theta_n}}{\delta_n \Delta_i(n)},
$$

(40)

where $\delta_n$ and $\Delta(n)$ are as described in Section $4.3$ and $\overline{C}_{\theta_n + \delta_n \Delta(n)}$ (resp. $\overline{C}_{\theta_n}$) denotes the CPT-value estimate that uses $m_n$ samples of the r.v. $X_{\theta_n + \delta_n \Delta(n)}$ (resp. $X_{\theta_n}$).

The complete algorithm with CPT-value as the risk measure and the usual value function as the objective
Algorithm 4: Policy gradient algorithm under CPT as a risk measure

**Input**: initial parameter \( \theta_0 \in \Theta \), where \( \Theta \) is a compact and convex subset of \( \mathbb{R}^d \), perturbation constants \( \delta_n > 0 \), trajectory lengths \( \{ m_n \} \), step-sizes \( \{ \zeta_1(n), \zeta_2(n) \} \), projection operators \( \Gamma \) and \( \Gamma_\lambda \), number of iterations \( M \gg 1 \).

1. for \( n \leftarrow 0 \) to \( M - 1 \) do
   2. for \( m \leftarrow 0 \) to \( m_n - 1 \) do
      3. /* Unperturbed policy simulation */
      4. Use the policy \( \mu_{\theta_n} \) to generate the state \( x_m \), draw action \( a_m \sim \mu_{\theta_n} (\cdot | x_m) \);
      5. Observe next state \( x_m+1 \) and cost \( k(x_m, a_m) \);
      6. Use (23) and (33) to estimate \( J(\theta_n) \) and \( G(\theta_n) \); // TD-learning
      7. /* Perturbed policy simulation */
      8. Use the policy \( \mu_{\theta_n + \delta_n \Delta(n)} \) to generate the state \( x_m^+ \), draw action \( a_m^+ \sim \mu_{\theta_n + \delta_n \Delta(n)} (\cdot | x_m^+) \);
      9. Observe next state \( x_m^+ + 1 \) and cost \( k(x_m^+, a_m^+) \);
   10. end
   11. /* Monte Carlo policy evaluation */
   12. Use the scheme in (39) to obtain \( \overline{C}_{\theta_n + \delta_n \Delta(n)} \) and \( \overline{C}_{\theta_n} \) - the estimates of the CPT-values \( \overline{C}(X_{\theta_n + \delta_n \Delta(n)}) \) and \( \overline{C}(X_{\theta_n}) \), respectively;
   13. Use (23) to estimate \( J(\theta_n + \delta_n \Delta(n)) \) and \( J(\theta_n) \); // TD-learning
   14. /* Gradient estimates using SPSA */
   15. Gradient of the objective: \( \hat{\nabla}_i J(\theta_n, x^0) = \frac{\hat{J}(\theta_n + \delta_n \Delta(n), x^0) - \hat{J}(\theta_n, x^0)}{\delta_n \Delta_i(n)} \);
   16. Gradient of the constraint: \( \hat{\nabla}_i C(X_{\theta_n}) = \frac{\overline{C}_{\theta_n + \delta_n \Delta(n)} - \overline{C}_{\theta_n}}{\delta_n \Delta_i(n)} \);
   17. /* Policy update: Gradient descent using SPSA */
   18. \( \theta_{n+1} = \Gamma \left[ \theta_n - \zeta_2(n) \left( \hat{\nabla}_i J(\theta_n, x^0) + \lambda_n \left( \hat{\nabla}_i C(X_{\theta_n}) - 2 \hat{J}(\theta_n, x^0) \hat{\nabla}_i J(\theta_n, x^0) \right) \right] \); 
   19. /* Lagrange multiplier update: Dual ascent using sample variance (formed using TD) */
   20. \( \lambda_{n+1} = \Gamma \left[ \lambda_n + \zeta_1(n) \left( \overline{C}_{\theta_n} - \alpha \right) \right] \);
   21. end

**Output**: Policy \( \theta_M \)

### 7 Conclusions and Future Challenges

In this article, we considered MDP problems that incorporate a variety of risk measures. The problems investigated encompassed discounted, average, and SSP settings. In the first two settings, the risk measure was variance (for the total and per period, respectively), whereas the SSP setting considered CVaR. CPT was the final risk measure that applied to all three settings. The algorithms had to grapple with a few challenges:
(i) lack of structure, leading to failure of classic DP methods (e.g., policy iteration for variance-constrained MDPs); (ii) lack of gradient information for the risk measures; and (iii) in the case of CVaR, challenges in estimation.

We briefly summarize some possible future research directions for risk-sensitive MDPs:

- For the discounted MDP setting, the current algorithm employs SPSA only, because a direct gradient estimate cannot be easily obtained for the risk measure. However, a likelihood ratio gradient estimator is available for the cost function, so combining SPSA with direct gradient-based search in a hybrid algorithm might improve the computational efficiency of the algorithm; see [82] for work along this line (but in the general SA setting, not specific to MDPs). Even more critical is that SPSA requires simulation of two system trajectories, which might be infeasible in real-time or online settings, so developing a risk-sensitive algorithm that uses only a single trajectory is of practical interest.

- For CVaR-constrained MDPs, variance reduction techniques such as importance sampling are essential for keeping CVaR estimation variance at reasonable levels, and as far as we are aware, there is no such provably convergent CVaR-estimation algorithm in an RL context. Another important need is incorporating function approximation to handle the curse of dimensionality for large state spaces.

- A critical challenge is to obtain non-asymptotic bounds for the risk-sensitive RL algorithms, which usually operate on multiple timescales. To the best of our knowledge, there are no non-asymptotic bounds available for multi-timescale stochastic approximation schemes, and hence, for actor-critic algorithms, even in the risk-neutral RL setting.

- The exploration-exploitation tradeoff has been studied in various contexts, with multi-armed bandit models offering one approach to this problem. It would be interesting to develop novel solution approaches involving risk measures such as CPT for CVaR in a bandit setting. Preliminary work on combining CPT with bandits is available in [9], but a lot of avenues remain open, e.g., finding the best CPT-optimal arm in a simple regret setting that does not involve the exploration-exploitation dilemma, and regret minimization in MDPs with risk criteria.

- Finally, CPT-value of the return of an MDP does not have a Bellman equation. Here, we proposed treating the CPT-value MDP problem as a black-box stochastic optimization problem, but certainly other approaches, especially ones that exploit special structure such as the Markovian property of an MDP, might be more computationally efficient in certain contexts.
References

[1] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[2] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[3] R. S. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[4] C. Szepesvári. Reinforcement learning algorithms for MDPs. *Wiley Encyclopedia of Operations Research and Management Science*, 2011.

[5] L. A. Prashanth and M. Ghavamzadeh. Actor-critic algorithms for risk-sensitive MDPs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 252–260, 2013.

[6] L. A. Prashanth and M. Ghavamzadeh. Variance-constrained actor-critic algorithms for discounted and average reward MDPs. *Machine Learning*, 105(3):367–417, dec 2016.

[7] L. A. Prashanth. Policy gradients for CVaR-constrained MDPs. In *Algorithmic Learning Theory (ALT)*, pages 155–169, 2014.

[8] L. A. Prashanth, C. Jie, M. C Fu, S. I. Marcus, and C. Szepesvári. Cumulative prospect theory meets reinforcement learning: prediction and control. In *International Conference on Machine Learning*, pages 1406–1415, 2016.

[9] A. Gopalan, L. A. Prashanth, M. C. Fu, and S. I. Marcus. Weighted bandits or: How bandits learn distorted values that are not expected. In *AAAI Conference on Artificial Intelligence*, pages 1941–1947, 2017.

[10] A. Tversky and D. Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, 5(4):297–323, 1992.

[11] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[12] K. J. Arrow. *Essays in the Theory of Risk Bearing*. Markham, Chicago, IL, 1971.

[13] R. A. Howard and J. E. Matheson. Risk-sensitive Markov decision processes. *Management Science*, 18:356–369, 1972.
[14] J. Filar, D. Krass, and K. Ross. Percentile performance criteria for limiting average Markov decision processes. *IEEE Transaction of Automatic Control*, 40(1):2–10, 1995.

[15] A. Prekopa. Probabilistic programming. In A. Ruszczyński and Shapiro, editors, *Stochastic Programming*. Elsevier, Amsterdam, 2003.

[16] D. Dentcheva and A. Ruszczyński. Optimization with stochastic dominance constraints. *SIAM Journal on Optimization*, 14(2):548–566, 2003.

[17] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

[18] A. Ruszczyński. Risk-averse dynamic programming for Markov decision processes. *Mathematical Programming*, 125:235–261, 2010.

[19] Y. Shen, W. Stannat, and K. Obermayer. Risk-sensitive Markov control processes. *SIAM Journal on Control and Optimization*, 51(5):3652–3672, 2013.

[20] P. Artzner, F. Delbaen, J. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[21] H. Föllmer and A. Schied. *Stochastic Finance: An Introduction in Discrete Time*. de Gruyter, 2004.

[22] F. Riedel. Dynamic coherent risk measures. *Stochastic Processes and Their Applications*, 112:185–200, 2004.

[23] A. Ruszczyński and A. Shapiro. Conditional risk mappings. *Mathematics of Operations Research*, 31:544–561, 2006.

[24] O. Cavus and A. Ruszczyński. Risk-averse control of undiscounted transient Markov models. *SIAM Journal on Control and Optimization*, 52(6):3935–3966, 2014.

[25] Y. Le Tallec. *Robust, Risk-sensitive, and Data-driven Control of Markov decision processes*. PhD thesis, MIT, 2007.

[26] S. Choi. *Risk-averse Newsvendor Models*. PhD thesis, Rutgers, 2009.

[27] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, pages 263–291, 1979.
[28] C. Starmer. Developments in non-expected utility theory: The hunt for a descriptive theory of choice under risk. *Journal of Economic Literature*, pages 332–382, 2000.

[29] K. Lin. *Stochastic Systems with Cumulative Prospect Theory*. PhD thesis, University of Maryland, College Park, 2013.

[30] K. Lin and S. I. Marcus. Dynamic programming with non-convex risk-sensitive measures. In *Proceedings of the 2013 American Control Conference*, Washington, D.C., June 2013.

[31] K. Lin and S. I. Marcus. Cumulative weighting optimization: The discrete case. In *Proceedings of the 2013 Winter Simulation Conference*, Washington, D.C., December 2013. Institute of Electrical and Electronic Engineers, Inc.

[32] K. Lin, C. Jie, and S. I. Marcus. Probabilistically distorted risk-sensitive infinite-horizon dynamic programming. *Automatica*, 2018.

[33] P. Whittle. *Risk-sensitive Optimal Control*. Wiley-Interscience series in systems and optimization. Wiley, 1990.

[34] V. S. Borkar and S. P. Meyn. Risk-sensitive optimal control for Markov decision processes with monotone cost. *Mathematics of Operations Research*, 27(1):192–209, 2002.

[35] W. H. Fleming and W. M. McEneaney. Risk-sensitive control on an infinite time horizon. *SIAM Journal on Control and Optimization*, 33(6):1881–1915, 1995.

[36] E. Fernández-Gaucherand and S. I. Marcus. Risk-sensitive optimal control of hidden Markov models: Structural results. *IEEE Transactions on Automatic Control*, 42:1418–1422, 1997.

[37] D. Hernández-Hernández and S. I. Marcus. Risk-sensitive control of Markov processes in countable state space. *Systems and Control Letters*, 34:105–106, 1998.

[38] D. Hernández-Hernández and S. I. Marcus. Existence of risk sensitive optimal stationary policies for controlled Markov processes. *Applied Mathematics and Optimization*, 40:273–285, 1999.

[39] S. P. Coraluppi and S. I. Marcus. Mixed risk-neutral/minimax control of discrete-time, finite-state Markov decision processes. *IEEE Transactions on Automatic Control*, 45:528–532, 2000.
[40] S. P. Coraluppi and S. I. Marcus. Risk-sensitive and minimax control of discrete-time, finite-state Markov decision processes. *Automatica*, 35:301–309, 1999.

[41] S. P. Coraluppi and S. I. Marcus. Risk-sensitive, minimax, and mixed risk neutral/minimax control of Markov decision processes. In *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W. H. Fleming*, pages 21–40. Boston: Birkhauser, 1999.

[42] S. I. Marcus, E. Fernández-Gaucherand, S. Coraluppi D. Hernández-Hernández, and P. Fard. Risk sensitive Markov decision processes. In C. I. Byrnes, editor, *Systems and Control in the Twenty-First Century*, pages 263–279. Boston: Birkhauser, 1997.

[43] N. Bäuerle and U. Rieder. More risk-sensitive markov decision processes. *Mathematics of Operations Research*, 39(1):105–120, 2014.

[44] V. S. Borkar. Learning algorithms for risk-sensitive control. In *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems–MTNS*, volume 5, 2010.

[45] M. Sobel. The variance of discounted Markov decision processes. *Journal of Applied Probability*, pages 794–802, 1982.

[46] S. Mannor and J. N. Tsitsiklis. Algorithmic aspects of mean–variance optimization in Markov decision processes. *European Journal of Operational Research*, 231(3):645–653, 2013.

[47] J. Filar, L. Kallenberg, and H. Lee. Variance-penalized Markov decision processes. *Mathematics of Operations Research*, 14(1):147–161, 1989.

[48] A. Barto, R. S. Sutton, and C. Anderson. Neuron-like elements that can solve difficult learning control problems. *IEEE Transaction on Systems, Man and Cybernetics*, 13:835–846, 1983.

[49] R. S. Sutton. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst, 1984.

[50] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

[51] V. S. Borkar and R. Jain. Risk-constrained Markov decision processes. In *IEEE Conference on Decision and Control*, pages 2664–2669, 2010.
[52] L. A. Prashanth. Policy gradients for CVaR-constrained MDPs. In Algorithmic Learning Theory, pages 155–169, 2014.

[53] A. Tamar, Y. Glassner, and S. Mannor. Optimizing the CVaR via sampling. arXiv preprint arXiv:1404.3862, 2014.

[54] C. Jie, L. A. Prashanth, M. C. Fu, S. I. Marcus, and C. Szepesvári. Stochastic optimization in a cumulative prospect theory framework. IEEE Transactions on Automatic Control, 63(9):2867–2882, 2018.

[55] A. Tamar, D. D. Castro, and S. Mannor. Policy gradients with variance related risk criteria. In Proceedings of the Twenty-Ninth International Conference on Machine Learning, pages 387–396, 2012.

[56] O. Mihatsch and R. Neuneier. Risk-sensitive reinforcement learning. Machine Learning, 49(2):267–290, 2002.

[57] A. Tamar, Y. Chow, M. Ghavamzadeh, and S. Mannor. Policy gradient for coherent risk measures. CoRR, abs/1502.03919, 2015.

[58] V. S. Borkar. An actor-critic algorithm for constrained Markov decision processes. Systems & Control Letters, 54(3):207–213, 2005.

[59] S. Bhatnagar. An actor–critic algorithm with function approximation for discounted cost constrained Markov decision processes. Systems & Control Letters, 59(12):760–766, 2010.

[60] D. R. Jiang and W. B. Powell. Risk-averse approximate dynamic programming with quantile-based risk measures. Mathematics of Operations Research, 43(2):554–579, 2017.

[61] D. P. Bertsekas. Dynamic Programming and Optimal Control, Vol. II, 4th edition. Athena Scientific, 2012.

[62] A. Tamar, D. Di Castro, and S. Mannor. Temporal difference methods for the variance of the reward to go. In International Conference on Machine Learning, pages 495–503, 2013.

[63] N. C. Barberis. Thirty years of prospect theory in economics: A review and assessment. Journal of Economic Perspectives, pages 173–196, 2013.

[64] E. Altman. Constrained Markov Decision Processes, volume 7. CRC Press, 1999.
[65] M. Sion. On general minimax theorems. Pacific J. Math, 8(1):171–176, 1958.

[66] V. S. Borkar. Stochastic Approximation: A Dynamical Systems Viewpoint. Cambridge University Press, 2008.

[67] V. R. Konda and V. S. Borkar. Actor-critic–type learning algorithms for Markov decision processes. SIAM Journal on Control and Optimization, 38(1):94–123, 1999.

[68] S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee. Natural actor-critic algorithms. Automatica, 45(11):2471–2482, 2009.

[69] S. Bhatnagar, H. L. Prasad, and L. A. Prashanth. Stochastic Recursive Algorithms for Optimization, volume 434. Springer, 2013.

[70] H. Robbins and S. Monro. A stochastic approximation method. The Annals of Mathematical Statistics, pages 400–407, 1951.

[71] H. J Kushner and G. G. Yin. Stochastic Approximation Algorithms and Applications. Springer-Verlag, 2nd edition, 2003.

[72] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. IEEE Transactions on Automatic Control, 42(5):674–690, 1997.

[73] J. N. Tsitsiklis and B. Van Roy. Average cost temporal-difference learning. Automatica, 35(11):1799–1808, 1999.

[74] J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions on Automatic Control, 37(3):332–341, 1992.

[75] M. C. Fu. Gradient estimation. In S. G. Henderson and B. L. Nelson, editors, Handbooks in Operations Research and Management Science: Simulation, chapter 19, pages 575–616. Elsevier, 2006.

[76] M. C. Fu. Stochastic gradient estimation. In Michael C. Fu, editor, Handbook on Simulation Optimization, chapter 5. Springer, 2015.

[77] H. Kushner and D. Clark. Stochastic Approximation Methods for Constrained and Unconstrained Systems. Springer-Verlag, 1978.
[78] A. Mas-Colell, M. Whinston, and J. Green. *Microeconomic theory*. Oxford University Press, 1995.

[79] Peter L Bartlett and Jonathan Baxter. Infinite-horizon policy-gradient estimation. *arXiv preprint arXiv:1106.0665*, 2011.

[80] A. Tamar, Y. Glassner, and S. Mannor. Policy gradients beyond expectations: Conditional Value-at-Risk. *arXiv preprint arXiv:1404.3862*, 2014.

[81] O. Bardou, N. Frikha, and G. Pages. Computing VaR and CVaR using stochastic approximation and adaptive unconstrained importance sampling. *Monte Carlo Methods and Applications*, 15(3):173–210, 2009.

[82] M. Chau, H. Qu, and M. Fu. A new hybrid stochastic approximation algorithm. In *Proceedings of the 12th International Workshop on Discrete Event Systems*, 2014.