

# Variance-constrained actor-critic algorithms for discounted and average reward MDPs

L. A. Prashanth<sup>1</sup> · Mohammad Ghavamzadeh<sup>2,3</sup>

Received: 25 March 2014 / Accepted: 27 May 2016 / Published online: 5 August 2016  
© The Author(s) 2016

**Abstract** In many sequential decision-making problems we may want to manage risk by minimizing some measure of variability in rewards in addition to maximizing a standard criterion. Variance related risk measures are among the most common risk-sensitive criteria in finance and operations research. However, optimizing many such criteria is known to be a hard problem. In this paper, we consider both discounted and average reward Markov decision processes. For each formulation, we first define a measure of variability for a policy, which in turn gives us a set of risk-sensitive criteria to optimize. For each of these criteria, we derive a formula for computing its gradient. We then devise actor-critic algorithms that operate on three timescales—a TD critic on the fastest timescale, a policy gradient (actor) on the intermediate timescale, and a dual ascent for Lagrange multipliers on the slowest timescale. In the discounted setting, we point out the difficulty in estimating the gradient of the variance of the return and incorporate simultaneous perturbation approaches to alleviate this. The average setting, on the other hand, allows for an actor update using compatible features to estimate the gradient of the variance. We establish the convergence of our algorithms to locally risk-sensitive optimal policies. Finally, we demonstrate the usefulness of our algorithms in a traffic signal control application.

---

Editor: Prasad Tadepalli.

---

Mohammad Ghavamzadeh was at Adobe Research, USA, on leave of absence from INRIA Lille, France.

---

✉ L. A. Prashanth  
prashla@isr.umd.edu

Mohammad Ghavamzadeh  
mohammad.ghavamzadeh@inria.fr

<sup>1</sup> Institute for Systems Research, University of Maryland, College Park, USA

<sup>2</sup> Adobe Research, California, USA

<sup>3</sup> INRIA, Lille, France

**Keywords** Markov decision process (MDP) · Reinforcement learning (RL) · Risk sensitive RL · Actor-critic algorithms · Multi-time-scale stochastic approximation · Simultaneous perturbation stochastic approximation (SPSA) · Smoothed functional (SF)

## 1 Introduction

The usual optimization criteria for an infinite horizon Markov decision process (MDP) are the *expected sum of discounted rewards* and the *average reward* (Puterman 1994; Bertsekas 1995). Many algorithms have been developed to maximize these criteria both when the model of the system is known (planning) and unknown (learning) (Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998). These algorithms can be categorized to **value function-based** methods that are mainly based on the two celebrated dynamic programming algorithms *value iteration* and *policy iteration*; and **policy gradient** methods that are based on updating the policy parameters in the direction of the gradient of a performance measure, i.e., the value function of the initial state or the average reward. Policy gradient methods estimate the gradient of the performance measure either without using an explicit representation of the value function (e.g., Williams 1992; Marbach 1998; Baxter and Bartlett 2001) or using such a representation in which case they are referred to as *actor-critic* algorithms (e.g., Sutton et al. 2000; Konda and Tsitsiklis 2000; Peters et al. 2005; Bhatnagar et al. 2007, 2009a). Using an explicit representation for value function (e.g., linear function approximation) by actor-critic algorithms reduces the variance of the gradient estimate with the cost of adding it a bias.

Actor-critic methods were among the earliest to be investigated in RL (Barto et al. 1983; Sutton 1984). They comprise a family of reinforcement learning (RL) methods that maintain two distinct algorithmic components: An *Actor*, whose role is to maintain and update an action-selection policy; and a *Critic*, whose role is to estimate the value function associated with the actor's policy. Thus, the critic addresses a problem of *prediction*, whereas the actor is concerned with *control*. A common practice is to update the policy parameters using stochastic gradient ascent, and to estimate the value-function using some form of temporal difference (TD) learning (Sutton 1988).

However in many applications, we may prefer to minimize some measure of *risk* as well as maximizing a usual optimization criterion. In such cases, we would like to use a criterion that incorporates a penalty for the *variability* induced by a given policy. This variability can be due to two types of uncertainties: (i) uncertainties in the model parameters, which is the topic of *robust* MDPs (e.g., Nilim and Ghaoui 2005; Delage and Mannor 2010; Xu and Mannor 2012), and (ii) the inherent uncertainty related to the stochastic nature of the system, which is the topic of *risk-sensitive* MDPs (e.g., Howard and Matheson 1972; Sobel 1982; Filar et al. 1989).

In risk-sensitive sequential decision-making, the objective is to maximize a risk-sensitive criterion such as the expected exponential utility (Howard and Matheson 1972), a variance related measure (Sobel 1982; Filar et al. 1989), the percentile performance (Filar et al. 1995), or conditional value-at-risk (CVaR) (Ruszczyński 2010; Shen et al. 2013). Unfortunately, when we include a measure of risk in our optimality criteria, the corresponding optimal policy is usually no longer Markovian stationary (e.g., Filar et al. 1989) and/or computing it is not tractable (e.g., Filar et al. 1989; Mannor and Tsitsiklis 2011). In particular, (i) In Sobel (1982), the author analyzed variance constraints in the context of a discounted reward MDP and showed the existence of a Bellman equation for the variance of the return. However, it was established there that the operator underlying the aforementioned Bellman equation is

not necessarily monotone. The latter is a crucial requirement for employing popular dynamic programming procedures for solving MDPs. (ii) In [Mannor and Tsitsiklis \(2013\)](#), the authors provide hardness results for variance constrained MDPs and in particular show that finding a globally mean–variance optimal policy in a discounted MDP is NP-hard, even when the underlying transition dynamics are known. (iii) In [Filar et al. \(1989\)](#), the authors established hardness results for average reward MDP, with a variance constraint that differs significantly from its counterpart in the discounted setting. Nevertheless, the variance constraint is well motivated considering the objective is to optimize a long-run average reward. However, the mathematical difficulties in finding a globally mean variance optimal policy remains, even with this altered variance constraint.

Although risk-sensitive sequential decision-making has a long history in operations research and finance, it has only recently grabbed attention in the machine learning community. Most of the work on this topic (including those mentioned above) has been in the context of MDPs (when the model of the system is known) and much less work has been done within the reinforcement learning (RL) framework (when the model is unknown and all the information about the system is obtained from the samples resulted from the agent’s interaction with the environment). In risk-sensitive RL, we can mention the work by [Borkar \(2001, 2002, 2010\)](#) and [Basu et al. \(2008\)](#) who considered the expected exponential utility, the one by [Mihatsch and Neuneier \(2002\)](#) that formulated a new risk-sensitive control framework based on transforming the temporal difference errors that occur during learning, and the one by [Tamar et al. \(2012\)](#) on several variance related measures. [Tamar et al. \(2012\)](#) study stochastic shortest path problems, and in this context, propose a policy gradient algorithm [and in a more recent work ([Tamar and Mannor 2013](#)) an actor-critic algorithm] for maximizing several risk-sensitive criteria that involve both the expectation and variance of the return random variable (defined as the sum of the rewards that the agent obtains in an episode).

In this paper,<sup>1</sup> we develop actor-critic algorithms for optimizing variance-related risk measures in both discounted and average reward MDPs. In the following, we first summarize our contributions in the discounted reward setting and follow it with those in average reward setting.

*Discounted reward setting* Here we define the measure of variability as the *variance of the return* [similar to [Tamar et al. \(2012\)](#)]. We formulate the following constrained optimization problem with the aim of maximizing the mean of the return subject to its variance being bounded from above: For a given  $\alpha > 0$ ,

$$\max_{\theta} V^{\theta}(x^0) \quad \text{subject to} \quad \Lambda^{\theta}(x^0) \leq \alpha.$$

In the above,  $V^{\theta}(x^0)$  is the mean of the return, starting in state  $x^0$  for a policy identified by its parameter  $\theta$ , while  $\Lambda^{\theta}(x^0)$  is the variance of the return (see Sect. 3 for precise definitions). A standard approach to solve the above problem is to employ the Lagrangian relaxation procedure ([Bertsekas 1999](#)) and solve the following unconstrained problem:

$$\max_{\lambda} \min_{\theta} \left( L(\theta, \lambda) \triangleq -V^{\theta}(x^0) + \lambda(\Lambda^{\theta}(x^0) - \alpha) \right),$$

where  $\lambda$  is the Lagrange multiplier. For solving the above problem, it is required to derive a formula for the gradient of the Lagrangian  $L(\theta, \lambda)$ , both w.r.t.  $\theta$  and  $\lambda$ . While the gradient

<sup>1</sup> This paper is an extension of an earlier work by the authors ([Prashanth and Ghavamzadeh 2013](#)) and includes novel second order methods in the discounted setting, detailed proofs of all proposed algorithms, and additional experimental results.

w.r.t.  $\lambda$  is particularly simple since it is the constraint value, the other gradient, i.e., w.r.t.  $\theta$  is complicated. We derive this formula in Lemma 1 and show that  $\nabla_{\theta} L(\theta, \lambda)$  requires the gradient of the value function at every state of the MDP (see the discussion in Sects. 3, 4).

Note that we operate in a *simulation optimization* setting, i.e., we have access to reward samples from the underlying MDP. Thus, it is required to estimate the mean and variance of the return (we use a TD-critic for this purpose) and then use these estimates to compute gradient of the Lagrangian. The latter is used then used to descend in the policy parameter. We estimate the gradient of the Lagrangian using two simultaneous perturbation methods: *simultaneous perturbation stochastic approximation* (SPSA) (Spall 1992) and *smoothed functional* (SF) (Katkovnik and Kulchitsky 1972), resulting in two separate discounted reward actor-critic algorithms. In addition, we also propose second-order algorithms with a Newton step, using both SPSA and SF.

Simultaneous perturbation methods have been popular in the field of stochastic optimization and the reader is referred to Bhatnagar et al. (2013) for a textbook introduction. First introduced in Spall (1992), the idea of SPSA is to perturb each coordinate of a parameter vector uniformly using a Rademacher random variable, in the quest for finding the minimum of a function that is only observable via simulation. Traditional gradient schemes require  $2\kappa_1$  evaluations of the function, where  $\kappa_1$  is the parameter dimension. On the other hand, SPSA requires only two evaluations irrespective of the parameter dimension and hence is an efficient scheme, especially useful in high-dimensional settings. While a one-simulation variant of SPSA was proposed in Spall (1997), the original two-simulation SPSA algorithm is preferred as it is more efficient and also seen to work better than its one-simulation variant. Later enhancements to the original SPSA scheme include using deterministic perturbation using certain Hadamard matrices (Bhatnagar et al. 2003) and second-order methods that estimate Hessian using SPSA (Spall 2000; Bhatnagar 2005). The SF schemes are another class of simultaneous perturbation methods, which again perturb each coordinate of the parameter vector uniformly. However, unlike SPSA, Gaussian random variables are used here for the perturbation. Originally proposed in Katkovnik and Kulchitsky (1972), the SF schemes have been studied and enhanced in later works such as Styblinski and Opalski (1986) and Bhatnagar (2007). Further, Bhatnagar et al. (2011) proposes both SPSA and SF like schemes for constrained optimization.

*Average reward setting* Here we first define the measure of variability as the *long-run variance* of a policy as follows:

$$\Lambda(\theta) = \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{n=0}^{T-1} (R_n - \rho(\mu))^2 \middle| \theta \right],$$

where  $\rho(\theta)$  is the average reward under policy identified by its parameter  $\theta$  (see Sect. 5 for precise definitions). The aim here is to solve the following constrained optimization problem:

$$\max_{\theta} \rho(\theta) \quad \text{subject to} \quad \Lambda(\theta) \leq \alpha.$$

As in the discounted setting, we derive an expression for the gradient of the Lagrangian (see Lemma 3). Unlike the discounted setting, we do not require sophisticated simulation optimizations schemes, as the gradient expressions in Lemma 3 suggest a simpler alternative that employs *compatible features* (Sutton et al. 2000; Peters et al. 2005). Compatible features for linearly approximating the action-value function of policy  $\theta$  are of the form  $\nabla \log \mu(a|x)$ . These features are well-defined if the policy is differentiable w.r.t. its parameters  $\theta$ . Sutton et al. (2000) showed the advantages of using these features in approximating the

action-value function in actor-critic algorithms. In [Bhatnagar et al. \(2009a\)](#), the authors use compatible features to develop actor-critic algorithms for a risk-neutral setting. We extend this to variance-constrained setting and establish that the square value function itself serves as a good baseline level when calculating the gradient of the average square reward (see the discussion surrounding [Lemma 4](#)). This facilitates the usage of *compatible features* for obtaining unbiased estimates of both average reward as well as square reward. We then develop an actor-critic algorithm that employ these *compatible features* in order to descend in the policy parameter  $\theta$  and also identify the bias that arises due to function approximation (see [Lemma 5](#)).

*Proof of convergence* Using the ordinary differential equations (ODE) approach, we establish the asymptotic convergence of our algorithms to locally risk-sensitive optimal policies and in the light of hardness results from [Mannor and Tsitsiklis \(2013\)](#), this is the best one can hope to achieve. Our algorithms employ multi-timescale stochastic approximation, in both settings. The convergence proof proceeds by analysing each timescale separately. In essence, the iterates on a faster timescale view those on a slower timescale as quasi-static, while the slower timescale iterate views that on a faster timescale as equilibrated. Using this principle, we show that TD critic (on the fastest timescale in all the algorithms) converge to fixed points of the Bellman operator, for any fixed policy  $\theta$  and Lagrange multiplier  $\lambda$ . Next, for any given  $\lambda$ , the policy update tracks in the asymptotic limit and converges to the equilibria of the corresponding ODE. Finally,  $\lambda$  updates on slowest timescale converge and the overall convergence is to a local saddle point of the Lagrangian. Moreover, the limiting point is feasible for the constrained optimization problem mentioned above, i.e., the policy obtained upon convergence satisfies the constraint that the variance is upper-bounded by  $\alpha$ .

*Simulation experiments* We demonstrate the usefulness of our discounted and average reward risk-sensitive actor-critic algorithms in a traffic signal control application. On this high-dimensional system with state space  $\approx 10^{32}$ , the objective in our formulation is to minimize the total number of vehicles in the system, which indirectly minimizes the delay experienced by the system. The motivation behind using a risk-sensitive control strategy is to reduce the variations in the delay experienced by road users. From the results, we observe that the risk-sensitive algorithms proposed in this paper result in a long-term (discounted or average) cost that is higher than their risk-neutral variants. However, from the empirical variance of the cost (both discounted as well as average) perspective, the risk-sensitive algorithms outperform their risk-neutral variants. Moreover, the experiments in the discounted setting also show that our SPSA based actor-critic scheme outperforms the policy gradient algorithm proposed in [Tamar et al. \(2012\)](#), both from a mean–variance as well as gradient estimation standpoints. This observation justifies using the actor-critic approach for solving risk-sensitive MDPs, as it reduces the variance of the gradient estimated by the policy gradient approach with the cost of introducing a bias induced by the value function representation.

*Remark 1* It is important to note that both our discounted and average reward algorithms can be easily extended to other variance related risk criteria such as the Sharpe ratio, which is popular in financial decision-making ([Sharpe 1966](#)) (see [Remarks 3, 7](#) for more details).

*Remark 2* Another important point is that the *expected exponential utility* risk measure can be also considered as an approximation of the mean–variance tradeoff due to the following Taylor expansion [see e.g., Eq. 11 in [Mihatsch and Neuneier \(2002\)](#)]

$$-\frac{1}{\beta} \log \mathbb{E}[e^{-\beta X}] = \mathbb{E}[X] - \frac{\beta}{2} \text{Var}[X] + O(\beta^2),$$

and we know that it is much easier to design actor-critic or other reinforcement learning algorithms (Borkar 2001, 2002; Basu et al. 2008; Borkar 2010) for this risk measure than those that will be presented in this paper. However, this formulation is limited in the sense that it requires knowing the ideal tradeoff between the mean and variance, since it takes  $\beta$  as an input. On the other hand, the mean–variance formulations considered in this paper are more general because

- (i) we optimize for the Lagrange multiplier  $\lambda$ , which plays a similar role to  $\beta$ , as a tradeoff between the mean and variance, and
- (ii) it is usually more natural to know an upper-bound on the variance (as in the mean–variance formulations considered in this paper) than knowing the ideal tradeoff between the mean and variance (as considered in the expected exponential utility formulation).

Despite all these, we should not consider these formulations as replacement for each other or try to find a formulation that is the best for all problems, but instead should consider them as different formulations that each might be the right fit for a specific problem.

*Closely related works* In comparison to Tamar et al. (2012) and Tamar and Mannor (2013), which are the most closely related contributions, we would like to point out the following:

- (i) The authors develop policy gradient and actor-critic methods for stochastic shortest path problems in Tamar et al. (2012) and Tamar and Mannor (2013), respectively. On the other hand, we devise actor-critic algorithms for both discounted and average reward MDP settings; and
- (ii) More importantly, we note the difficulty in the discounted formulation that requires to estimate the gradient of the value function at every state of the MDP and also sample from two different distributions. This precludes us from using *compatible features*—a method that has been employed successfully in actor-critic algorithms in a risk-neutral setting (cf. Bhatnagar et al. 2009a) as well as more recently in Tamar and Mannor (2013) for a risk-sensitive stochastic shortest path setting. We alleviate the above mentioned problems for the discounted setting by employing simultaneous perturbation based schemes for estimating the gradient in the first order methods and Hessian in the second order methods, that we propose.
- (iii) Unlike (Tamar et al. 2012; Tamar and Mannor 2013) who consider a fixed  $\lambda$  in their constrained formulations, we perform dual ascent using sample variance constraints and optimize the Lagrange multiplier  $\lambda$ . In rigorous terms,  $\lambda_n$  in our algorithms is shown to converge to a local maxima of  $\nabla_{\lambda} L(\theta^{\lambda}, \lambda)$  (here  $\theta^{\lambda}$  is the limit of the  $\theta$  recursion for a given value of  $\lambda$ ) and the limit  $\lambda^*$  is such that the variance constraint is satisfied for the corresponding policy  $\theta^{\lambda^*}$ .

*Organization of the paper* The rest of the paper is organized as follows: In Sect. 2, we describe the RL setting. In Sect. 3, we describe the risk-sensitive MDP in the discounted setting and propose actor-critic algorithms for this setting in Sect. 4. In Sect. 5, we present the risk measure for the average setting and propose an actor-critic algorithm that optimizes this risk measure in Sect. 6. In Sects. 7 and 8, we present the convergence proofs for the algorithms in discounted and average reward settings, respectively. In Sect. 9, we describe the experimental setup and present the results in both average and discounted cost settings. Finally, in Sect. 10, we provide the concluding remarks and outline a few future research directions.

## 2 Preliminaries

We consider sequential decision-making tasks that can be formulated as a reinforcement learning (RL) problem. In RL, an agent interacts with a dynamic, stochastic, and incompletely known environment, with the goal of optimizing some measure of its *long-term* performance. This interaction is often modeled as a Markov decision process (MDP). A MDP is a tuple  $(\mathcal{X}, \mathcal{A}, R, P, x^0)$  where  $\mathcal{X}$  and  $\mathcal{A}$  are the state and action spaces;  $R(x, a), x \in \mathcal{X}, a \in \mathcal{A}$  is the reward random variable whose expectation is denoted by  $r(x, a) = \mathbb{E}[R(x, a)]$ ;  $P(\cdot|x, a)$  is the transition probability distribution; and  $x^0 \in \mathcal{X}$  is the initial state.<sup>2</sup> We assume that both state and action spaces are finite.

The rule according to which the agent acts in its environment (selects action at each state) is called a *policy*. A Markovian stationary policy  $\mu(\cdot|x)$  is a probability distribution over actions, conditioned on the current state  $x$ . The goal in a RL problem is to find a policy that optimizes the long-term performance measure of interest, e.g., maximizes the *expected discounted sum of rewards* or the *average reward*.

In policy gradient and actor-critic methods, we define a class of parameterized stochastic policies  $\{\mu(\cdot|x; \theta), x \in \mathcal{X}, \theta \in \Theta \subseteq \mathbb{R}^{k_1}\}$ , estimate the gradient of the performance measure w.r.t. the policy parameters  $\theta$  from the observed system trajectories, and then improve the policy by adjusting its parameters in the direction of the gradient. Here  $\Theta$  denotes a compact and convex subset of  $\mathbb{R}^{k_1}$ . Our algorithms projects the iterates onto  $\Theta$ , which ensures stability—a crucial requirement necessary for establishing convergence. Since in this setting a policy  $\mu$  is represented by its  $k_1$ -dimensional parameter vector  $\theta$ , policy dependent functions can be written as a function of  $\theta$  in place of  $\mu$ . So, we use  $\mu$  and  $\theta$  interchangeably in the paper.

We make the following assumptions on the policy, parameterized by  $\theta$ :

- (A1) For any state-action pair  $(x, a) \in \mathcal{X} \times \mathcal{A}$ , the policy  $\mu(a|x; \theta)$  is continuously differentiable in the parameter  $\theta$ .
- (A2) The Markov chain induced by any policy  $\theta$  is irreducible.

The above assumptions are standard requirements in policy gradient and actor-critic methods.

Finally, we denote by  $d^\mu(x)$  and  $\pi^\mu(x, a) = d^\mu(x)\mu(a|x)$ , the stationary distribution of state  $x$  and state-action pair  $(x, a)$  under policy  $\mu$ , respectively. The stationary distributions can be seen to exist because we consider a finite state-action space setting and irreducibility here implies positive recurrence. Similarly in the discounted formulation, we define the  $\gamma$ -discounted visiting distribution of state  $x$  and state-action pair  $(x, a)$  under policy  $\mu$  as  $d_\gamma^\mu(x|x^0) = (1 - \gamma) \sum_{n=0}^\infty \gamma^n \Pr(x_n = x|x_0 = x^0; \mu)$  and  $\pi_\gamma^\mu(x, a|x^0) = d_\gamma^\mu(x|x^0)\mu(a|x)$ .

## 3 Discounted reward setting

For a given policy  $\mu$ , we define the return of a state  $x$  (state-action pair  $(x, a)$ ) as the sum of discounted rewards encountered by the agent when it starts at state  $x$  (state-action pair  $(x, a)$ ) and then follows policy  $\mu$ , i.e.,

$$D^\mu(x) = \sum_{n=0}^\infty \gamma^n R(x_n, a_n) \mid x_0 = x, \mu,$$

<sup>2</sup> Our algorithms can be easily extended to a setting where the initial state is determined by a distribution.

$$D^\mu(x, a) = \sum_{n=0}^{\infty} \gamma^n R(x_n, a_n) \mid x_0 = x, a_0 = a, \mu.$$

The expected value of these two random variables are the value and action-value functions of policy  $\mu$ , i.e.,

$$V^\mu(x) = \mathbb{E}[D^\mu(x)] \quad \text{and} \quad Q^\mu(x, a) = \mathbb{E}[D^\mu(x, a)].$$

The goal in the standard (risk-neutral) discounted reward formulation is to find an optimal policy  $\mu^* = \arg \max_{\mu} V^\mu(x^0)$ , where  $x^0$  is the initial state of the system.

The most common measure of the *variability* in the stream of rewards is the *variance of the return*, defined by

$$\Lambda^\mu(x) \triangleq \mathbb{E}[D^\mu(x)^2] - V^\mu(x)^2 = U^\mu(x) - V^\mu(x)^2. \tag{1}$$

The above measure was first introduced by [Sobel \(1982\)](#). Note that

$$U^\mu(x) \triangleq \mathbb{E}[D^\mu(x)^2]$$

is the *square reward value function* of state  $x$  under policy  $\mu$ . On similar lines, we define the *square reward action-value function* of state-action pair  $(x, a)$  under policy  $\mu$  as

$$W^\mu(x, a) \triangleq \mathbb{E}[D^\mu(x, a)^2].$$

From the Bellman equation of  $\Lambda^\mu(x)$ , proposed by [Sobel \(1982\)](#), it is straightforward to derive the following Bellman equations for  $U^\mu(x)$  and  $W^\mu(x, a)$ :

$$U^\mu(x) = \sum_a \mu(a|x)r(x, a)^2 + \gamma^2 \sum_{a, x'} \mu(a|x)P(x'|x, a)U^\mu(x') + 2\gamma \sum_{a, x'} \mu(a|x)P(x'|x, a)r(x, a)V^\mu(x'),$$

$$W^\mu(x, a) = r(x, a)^2 + \gamma^2 \sum_{x'} P(x'|x, a)U^\mu(x') + 2\gamma r(x, a) \sum_{x'} P(x'|x, a)V^\mu(x'). \tag{2}$$

Although  $\Lambda^\mu$  of (1) satisfies a Bellman equation, unfortunately, it lacks the monotonicity property of dynamic programming (DP), and thus, it is not clear how the related risk measures can be optimized by standard DP algorithms ([Sobel 1982](#)). Policy gradient and actor-critic algorithms are good candidates to deal with this risk measure.

We consider the following risk-sensitive measure for discounted MDPs: For a given  $\alpha > 0$ ,

$$\max_{\theta} V^\theta(x^0) \quad \text{subject to} \quad \Lambda^\theta(x^0) \leq \alpha. \tag{3}$$

Assuming that there is at least one policy (in the class of parameterized policies that we consider) that satisfies the variance constraint above, it can be inferred from Theorem 3.8 of [Altman \(1999\)](#) that there exists an optimal policy that uses at most one randomization.

It is important to note that the algorithms proposed in this paper can be used for any risk-sensitive measure that is based on the variance of the return such as

1.  $\min_{\theta} \Lambda^\theta(x^0) \quad \text{subject to} \quad V^\theta(x^0) \geq \alpha,$
2.  $\max_{\theta} V^\theta(x^0) - \alpha \sqrt{\Lambda^\theta(x^0)},$
3. Maximizing the Sharpe ratio, i.e.,  $\max_{\theta} V^\theta(x^0) / \sqrt{\Lambda^\theta(x^0)}$ . Sharpe ratio (SR) is a popular risk measure in financial decision-making [Sharpe \(1966\)](#). Sect. 3 presents extensions of our proposed discounted reward algorithms to optimize the Sharpe ration.



To solve (3), we employ the Lagrangian relaxation procedure (Bertsekas 1999) to convert it to the following unconstrained problem:

$$\max_{\lambda} \min_{\theta} \left( L(\theta, \lambda) \triangleq -V^{\theta}(x^0) + \lambda(\Lambda^{\theta}(x^0) - \alpha) \right), \tag{4}$$

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. Sion 1958). 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 our setting, the Lagrangian  $L(\theta, \lambda)$  is not necessarily convex in  $\theta$ , which implies there may not be a 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 tuple  $(\theta^*, \lambda^*)$  which 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 have been shown to be NP-hard in Mannor and Tsitsiklis (2013) and the best one can hope is to find an approximately optimal policy.

In our setting, the necessary gradients of the Lagrangian are as follows:

$$\nabla_{\theta} L(\theta, \lambda) = -\nabla_{\theta} V^{\theta}(x^0) + \lambda \nabla_{\theta} \Lambda^{\theta}(x^0) \quad \text{and} \quad \nabla_{\lambda} L(\theta, \lambda) = \Lambda^{\theta}(x^0) - \alpha.$$

Since  $\nabla_{\theta} \Lambda^{\theta}(x^0) = \nabla_{\theta} U^{\theta}(x^0) - 2V^{\theta}(x^0) \nabla_{\theta} V^{\theta}(x^0)$ , in order to compute  $\nabla_{\theta} \Lambda^{\theta}(x^0)$  it would be enough to calculate  $\nabla_{\theta} V^{\theta}(x^0)$  and  $\nabla_{\theta} U^{\theta}(x^0)$ . Using the above definitions, we are now ready to derive the expressions for the gradient of  $V^{\theta}(x^0)$  and  $U^{\theta}(x^0)$ , which in turn constitute the main ingredients in calculating  $\nabla_{\theta} L(\theta, \lambda)$ .<sup>3</sup>

**Lemma 1** *Under (A1) and (A2), we have*

$$\begin{aligned} (1 - \gamma) \nabla V^{\theta}(x^0) &= \sum_{x,a} \pi_{\gamma}^{\theta}(x, a|x^0) \nabla \log \mu(a|x; \theta) Q^{\theta}(x, a), \\ (1 - \gamma^2) \nabla U^{\theta}(x^0) &= \sum_{x,a} \tilde{\pi}_{\gamma}^{\theta}(x, a|x^0) \nabla \log \mu(a|x; \theta) W^{\theta}(x, a) \\ &\quad + 2\gamma \sum_{x,a,x'} \tilde{\pi}_{\gamma}^{\theta}(x, a|x^0) P(x'|x, a) r(x, a) \nabla V^{\theta}(x'), \end{aligned}$$

where  $\tilde{d}_{\gamma}^{\theta}(x|x^0)$  and  $\tilde{\pi}_{\gamma}^{\theta}(x, a|x^0)$  are the  $\gamma^2$ -discounted visiting distributions of state  $x$  and state-action pair  $(x, a)$  under policy  $\mu$ , respectively, and are defined as

$$\begin{aligned} \tilde{d}_{\gamma}^{\theta}(x|x^0) &= (1 - \gamma^2) \sum_{n=0}^{\infty} \gamma^{2n} \Pr(x_n = x|x_0 = x^0; \theta), \\ \tilde{\pi}_{\gamma}^{\theta}(x, a|x^0) &= \tilde{d}_{\gamma}^{\theta}(x|x^0) \mu(a|x). \end{aligned}$$

<sup>3</sup> Henceforth, we shall drop the subscript  $\theta$  and use  $\nabla L(\theta, \lambda)$  to denote the derivative w.r.t.  $\theta$ .

*Proof* The proof of  $\nabla V^\theta(x^0)$  is standard and can be found, for instance, in Peters et al. (2005). To prove  $\nabla U^\theta(x^0)$ , we start by the fact that from (2) we have  $U(x) = \sum_a \mu(x|a)W(x, a)$ . If we take the derivative w.r.t.  $\theta$  from both sides of this equation and obtain

$$\begin{aligned} \nabla U(x^0) &= \sum_a \nabla \mu(a|x^0)W(x^0, a) + \sum_a \mu(a|x^0)\nabla W(x^0, a) \\ &= \sum_a \nabla \mu(a|x^0)W(x^0, a) + \sum_a \mu(a|x^0)\nabla \left[ r(x^0, a)^2 + \gamma^2 \sum_{x'} P(x'|x^0, a)U(x') \right. \\ &\quad \left. + 2\gamma r(x^0, a) \sum_{x'} P(x'|x^0, a)V(x') \right] \\ &= \underbrace{\sum_a \nabla \mu(a|x^0)W(x^0, a) + 2\gamma \sum_{a,x'} \mu(a|x^0)r(x^0, a)P(x'|x^0, a)\nabla V(x')}_{h(x^0)} \\ &\quad + \gamma^2 \sum_{a,x'} \mu(a|x^0)P(x'|x^0, a)\nabla U(x') \\ &= h(x^0) + \gamma^2 \sum_{a,x'} \mu(a|x^0)P(x'|x^0, a)\nabla U(x') \\ &= h(x^0) + \gamma^2 \sum_{a,x'} \mu(a|x^0)P(x'|x^0, a)\nabla \left[ h(x') \right. \\ &\quad \left. + \gamma^2 \sum_{a',x''} \mu(a'|x')P(x''|x', a')\nabla U(x'') \right]. \end{aligned} \tag{5}$$

By unrolling the last equation using the definition of  $\nabla U(x)$  from (5), we obtain

$$\begin{aligned} \nabla U(x^0) &= \sum_{n=0}^\infty \gamma^{2n} \sum_x \Pr(x_n = x|x_0 = x^0)h(x) = \frac{1}{1-\gamma^2} \sum_x \tilde{d}_\gamma(x|x^0)h(x) \\ &= \frac{1}{1-\gamma^2} \left[ \sum_{x,a} \tilde{d}_\gamma(x|x^0)\mu(a|x)\nabla \log \mu(a|x)W(x, a) \right. \\ &\quad \left. + 2\gamma \sum_{x,a,x'} \tilde{d}_\gamma(x|x^0)\mu(a|x)r(x, a)P(x'|x, a)\nabla V(x') \right] \\ &= \frac{1}{1-\gamma^2} \left[ \sum_{x,a} \tilde{\pi}_\gamma(x, a|x^0)\nabla \log \mu(a|x)W(x, a) \right. \\ &\quad \left. + 2\gamma \sum_{x,a,x'} \tilde{\pi}_\gamma(x, a|x^0)r(x, a)P(x'|x, a)\nabla V(x') \right]. \end{aligned}$$

□

In Sutton et al. (1999), a policy gradient result analogous to Lemma 1 is provided for the value function in the case of full-state representations. In the average reward setting, a similar result helps in extension to incorporate function approximation—see the actor-critic algorithms in Bhatnagar et al. (2009a).<sup>4</sup> However, a similar approach is not viable for discounted setting and this motivates the use of stochastic optimization techniques like SPSA/SF (cf. Bhatnagar 2010). The problem is further complicated in the variance-constrained setting that we consider because:

<sup>4</sup> We extend this to the case of variance-constrained MDP in Sect. 6.

1. two different sampling distributions,  $\pi_y^\theta$  and  $\tilde{\pi}_y^\theta$ , are used for  $\nabla V^\theta(x^0)$  and  $\nabla U^\theta(x^0)$ , and
2.  $\nabla V^\theta(x')$  appears in the second sum of  $\nabla U^\theta(x^0)$  equation, which implies that we need to estimate the gradient of the value function  $V^\theta$  at every state of the MDP, and not just at the initial state  $x^0$ .

To alleviate the above mentioned problems, we borrow the principle of simultaneous perturbation for estimating the gradient  $\nabla L(\theta, \lambda)$  and develop novel risk-sensitive actor-critic algorithms in the following section.

### 4 Discounted reward risk-sensitive actor-critic algorithms

In this section, we present actor-critic algorithms for optimizing the risk-sensitive measure (3). These algorithms are based on two simultaneous perturbation methods: *simultaneous perturbation stochastic approximation* (SPSA) and *smoothed functional* (SF).

#### 4.1 Algorithm structure

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 as follows:

- 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.  $\lambda$ .

Using two-timescale stochastic approximation (Chapter 6, [Borkar 2008](#)), the two loops above can run in parallel, as follows:

$$\theta_{n+1} = \Gamma[\theta_n - \zeta_2(n)A_n^{-1}\nabla L(\theta_n, \lambda_n)], \tag{6}$$

$$\lambda_{n+1} = \Gamma_\lambda[\lambda_n + \zeta_1(n)\nabla_\lambda L(\theta_n, \lambda_n)], \tag{7}$$

In the above,

- $A_n$  is a positive definite matrix that fixes the order of the algorithm. For the first order methods,  $A_n = I$  ( $I$  is the identity matrix), while for the second order methods  $A_n \rightarrow \nabla_\theta^2 L(\theta_n, \lambda_n)$  as  $n \rightarrow \infty$ .
- $\Gamma$  is a projection operator that keeps the iterate  $\theta_n$  stable by projecting onto a compact and convex set  $\Theta := \prod_{i=1}^{\kappa_1} [\theta_{\min}^{(i)}, \theta_{\max}^{(i)}]$ . In particular, for any  $\theta \in \mathbb{R}_1^\kappa$ ,  $\Gamma(\theta) = (\Gamma^{(1)}(\theta^{(1)}), \dots, \Gamma^{(\kappa_1)}(\theta^{(\kappa_1)}))^T$ , with  $\Gamma^{(i)}(\theta^{(i)}) := \min(\max(\theta_{\min}^{(i)}, \theta^{(i)}), \theta_{\max}^{(i)})$ .
- $\Gamma_\lambda$  is a projection operator that keeps the Lagrange multiplier  $\lambda_n$  within the interval  $[0, \lambda_{\max}]$ , for some large positive constant  $\lambda_{\max} < \infty$  and can be defined in an analogous fashion as  $\Gamma$ .
- $\zeta_1(n), \zeta_2(n)$  are step-sizes selected such that  $\theta$  update is on the faster and  $\lambda$  update is on the slower timescale. Note that another timescale  $\zeta_3(n)$  that is the fastest is used for the TD-critic, which provides the estimate of the Lagrangian for a given  $(\theta, \lambda)$ .

*Simulation optimization* We operate in a setting where we only observe simulated rewards of the underlying MDP. Thus, it is required to estimate the mean and variance of the return (we use a TD-critic for this purpose) and then use these estimates to compute gradient of the Lagrangian. The gradient  $\nabla_\lambda L(\theta, \lambda)$  has a particularly simple form of  $(A^\theta(x^0) - \alpha)$ ,

suggesting the usage of sample variance constraints to perform the dual ascent for Lagrange multiplier  $\lambda$ . On the other hand, the expression for gradient of  $L(\theta, \lambda)$  w.r.t.  $\theta$  is complicated (see Lemma 1) and warrants the usage of a simulation optimization that can provide gradient estimates from sample observation. We employ simultaneous perturbation schemes for estimating the gradient (and in the case of second order methods, the Hessian) of the Lagrangian  $L(\theta, \lambda)$ . The idea in these methods is to estimate the gradients  $\nabla V^\theta(x^0)$  and  $\nabla U^\theta(x^0)$  [needed for estimating the gradient  $\nabla L(\theta, \lambda)$ ] using two simulated trajectories of the system corresponding to policies with parameters  $\theta_n$  and  $\theta_n^+ = \theta_n + p_n$ . Here  $p_n$  is a perturbation vector that is specific to the algorithm.

Based on the order, our algorithms can be classified as:

1. *First order* This corresponds to  $A_n = I$  in (6). The proposed algorithms here include RS-SPSA-G and RS-SF-G, where the former estimates the gradient using SPSA, while the latter uses SF. These algorithms use the following choice for the perturbation vector:  $p_n = \beta_n \Delta_n$ . Here  $\beta_n > 0$  is a positive constant and  $\Delta_n$  is a perturbation random variable, i.e., a  $\kappa_1$ -vector of independent Rademacher (for SPSA) and Gaussian  $\mathcal{N}(0, 1)$  (for SF) random variables.
2. *Second order* This corresponds to  $A_n$  which converges to  $\nabla^2 L(\theta_n, \lambda_n)$  as  $n \rightarrow \infty$ . The proposed algorithms here include RS-SPSA-N and RS-SF-N, where the former uses SPSA for gradient/Hessian estimates and the latter employs SF for the same. These algorithms use the following choice for perturbation vector: For RS-SPSA-N,  $p_n = \beta_n \Delta_n + \beta_n \widehat{\Delta}_n$ ,  $\beta_n > 0$  is a positive constant and  $\Delta_n$  and  $\widehat{\Delta}_n$  are perturbation parameters that are  $\kappa_1$ -vectors of independent Rademacher random variables, respectively. For RS-SF-N,  $p_n = \beta_n \Delta_n$ , where  $\Delta_n$  is a  $\kappa_1$  vector of Gaussian  $\mathcal{N}(0, 1)$  random variables.

---

**Algorithm 1** Template of the Risk-Sensitive Discounted Reward Actor-Critic Algorithms

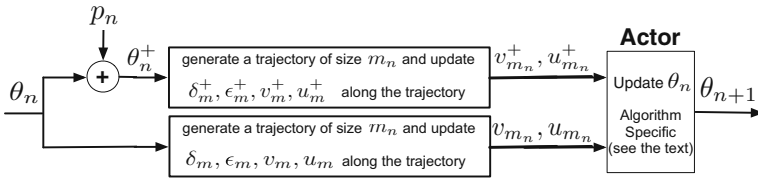
---

**Input:** parameterized policy  $\mu(\cdot|\cdot; \theta)$  and value function feature vectors  $\phi_v(\cdot)$  and  $\phi_u(\cdot)$   
**Initialization:** policy parameter  $\theta = \theta_0$ ; value function weight vectors  $v = v_0$  and  $v^+ = v_0^+$ ; square value function weight vectors  $u = u_0$  and  $u^+ = u_0^+$ ; initial state  $x_0 \sim P_0(x)$   
**for**  $n = 0, 1, 2, \dots$  **do**  
    **for**  $m = 0, 1, 2, \dots, m_n$  **do**  
        Draw action  $a_m \sim \mu(\cdot|x_m; \theta_n)$ , observe next state  $x_{m+1}$  and reward  $R(x_m, a_m)$   
        Draw action  $a_m^+ \sim \mu(\cdot|x_m^+; \theta_n^+)$ , observe next state  $x_{m+1}^+$  and reward  $R(x_m^+, a_m^+)$   
        **Critic Update:** see (13) and (15) in the text  
    **end for**  
    **Actor Update:** Algorithm-Specific  
    **Lagrange Multiplier Update:** see (21) in the text  
**end for**  
**return** policy and value function parameters  $\theta, \lambda, v, u$

---

The overall flow of our proposed actor-critic algorithms is illustrated in Fig. 1 and Algorithm 1. The overall operation involves the following two loops: At each time instant  $n$ , **Inner loop (critic update)** For a fixed policy (given as  $\theta_n$ ), simulate two system trajectories, each of length  $m_n$ , as follows:

- (1) *Unperturbed simulation* For  $m = 0, 1, \dots, m_n$ , take action  $a_m \sim \mu(\cdot|x_m; \theta_n)$ , observe the reward  $R(x_m, a_m)$ , and the next state  $x_{m+1}$  in the first trajectory.
- (2) *Perturbed simulation* For  $m = 0, 1, \dots, m_n$ , take action  $a_m^+ \sim \mu(\cdot|x_m^+; \theta_n^+)$ , observe the reward  $R(x_m^+, a_m^+)$ , and the next state  $x_{m+1}^+$  in the second trajectory.



**Fig. 1** The overall flow of our simultaneous perturbation based actor-critic algorithms

Using the method of temporal differences (TD) (Sutton 1984), estimate the value functions  $\widehat{V}^{\theta_n}(x^0)$  and  $\widehat{V}^{\theta_n^+}(x^0)$ , and square value functions  $\widehat{U}^{\theta_n}(x^0)$  and  $\widehat{U}^{\theta_n^+}(x^0)$ , corresponding to the policy parameter  $\theta_n$  and  $\theta_n^+$ .

**Outer loop (actor update)** Estimate the gradient/Hessian of  $\widehat{V}^\theta(x^0)$  and  $\widehat{U}^\theta(x^0)$ , and hence the gradient/Hessian of Lagrangian  $L(\theta, \lambda)$ , using either SPSA (17) or SF (18) methods. Using these estimates, update the policy parameter  $\theta$  in the descent direction using either a gradient or a Newton decrement, and the Lagrange multiplier  $\lambda$  in the ascent direction.

In the next section, we describe the TD-critic and subsequently, in Sects. 4.3 and 4.4, present the first and second order actor critic algorithms, respectively.

### 4.2 TD-critic

In our actor-critic algorithms, the critic uses linear approximation for the value and square value functions, i.e.,  $\widehat{V}(x) \approx v^\top \phi_v(x)$  and  $\widehat{U}(x) \approx u^\top \phi_u(x)$ , where the features  $\phi_v(\cdot)$  and  $\phi_u(\cdot)$  are from low-dimensional spaces  $\mathbb{R}^{\kappa_2}$  and  $\mathbb{R}^{\kappa_3}$ , respectively. Let  $\Phi_v$  and  $\Phi_u$  denote  $|\mathcal{X}| \times \kappa_2$  and  $|\mathcal{X}| \times \kappa_3$  dimensional matrices, whose  $i$ th columns are  $\phi_v^{(i)} = (\phi_v^{(i)}(x), x \in \mathcal{X})^\top$ ,  $i = 1, \dots, \kappa_2$  and  $\phi_u^{(i)} = (\phi_u^{(i)}(x), x \in \mathcal{X})^\top$ ,  $i = 1, \dots, \kappa_3$ . Let  $S_v := \{\Phi_v v \mid v \in \mathbb{R}_2^\kappa\}$  and  $S_u := \{\Phi_u u \mid u \in \mathbb{R}_3^\kappa\}$ , denote the subspaces within which we approximate the value and square value functions. We make the following standard assumption as in Bhatnagar et al. (2009a):

**(A3)** The basis functions  $\{\phi_v^{(i)}\}_{i=1}^{\kappa_2}$  and  $\{\phi_u^{(i)}\}_{i=1}^{\kappa_3}$  are linearly independent. In particular,  $\kappa_2, \kappa_3 \ll n$  and  $\Phi_v$  and  $\Phi_u$  are full rank. Moreover, for every  $v \in \mathbb{R}^{\kappa_2}$  and  $u \in \mathbb{R}^{\kappa_3}$ ,  $\Phi_v v \neq e$  and  $\Phi_u u \neq e$ , where  $e$  is the  $n$ -dimensional vector with all entries equal to one.

Let  $\Pi_u$  and  $\Pi_v$  be operators that project onto  $S_v$  and  $S_u$ , respectively and as a consequence of the above assumption, can be defined as follows:

$$\Pi_v = \Phi_v (\Phi_v^\top D^\theta \Phi_v)^{-1} \Phi_v^\top D^\theta \text{ and } \Pi_u = \Phi_u (\Phi_u^\top D^\theta \Phi_u)^{-1} \Phi_u^\top D^\theta, \tag{8}$$

where  $D^\theta$  is a diagonal  $|\mathcal{X}| \times |\mathcal{X}|$  matrix with entries  $d^\theta(x)$ , for each  $x \in \mathcal{X}$ . Recall that  $d^\theta(\cdot)$  denotes the stationary distribution of the Markov chain underlying policy  $\theta$ .

Let  $T^\theta = [T_v^\theta; T_u^\theta]$ , where  $T_v^\theta$  and  $T_u^\theta$  denote the Bellman operators for value and square value functions of the policy governed by parameter  $\theta$ , respectively. These operators are defined as: For any  $y \in \mathbb{R}^{2|\mathcal{X}|}$ , let  $y_v$  and  $y_u$  denote the first and last  $|\mathcal{X}|$  entries, respectively. Then

$$T^\theta y = [T_v^\theta y; T_u^\theta y], \text{ where} \tag{9}$$

$$T_v^\theta y = r^\theta + \gamma P^\theta y_v, \tag{10}$$

$$T_u^\theta y = R^\theta r^\theta + 2\gamma R^\theta P^\theta y_v + \gamma^2 P^\theta y_u, \tag{11}$$

where  $\mathbf{r}^\theta$  and  $\mathbf{P}^\theta$  are the reward vector and the transition probability matrix of policy  $\theta$ , and  $\mathbf{R}^\theta = \text{diag}(\mathbf{r}^\theta)$ .

Let  $\Pi = \begin{pmatrix} \Pi_v & 0 \\ 0 & \Pi_u \end{pmatrix}$ . Also, for any  $y \in \mathbb{R}^{2|\mathcal{X}|}$ , define its  $v$ -weighted norm as

$$\|y\|_v = v\|y_v\|_{\mathcal{D}^\theta} + (1 - v)\|y_u\|_{\mathcal{D}^\theta},$$

where  $\|z\|_{\mathcal{D}^\theta} = \sqrt{\sum_{i=1}^{|\mathcal{X}|} d^\theta(i)z_i^2}$  for any  $z \in \mathbb{R}^{|\mathcal{X}|}$ .

We now claim that the projected Bellman operator  $\Pi T$  is a contraction mapping w.r.t  $v$ -weighted norm, for any policy  $\theta$ .

**Lemma 2** *Under (A2) and (A3), there exists a  $v \in (0, 1)$  and  $\bar{\gamma} < 1$  such that*

$$\|\Pi T y - \Pi T \bar{y}\|_v \leq \bar{\gamma} \|y - \bar{y}\|_v, \forall y, \bar{y} \in \mathbb{R}^{2|\mathcal{X}|}.$$

*Proof* See Sect. 7.1. □

Let  $[\Phi_v \bar{v}; \Phi_u \bar{u}]$  denote the unique fixed-point of the projected Bellman operator  $\Pi T$ , i.e.,

$$\Phi_v \bar{v} = \Pi_v(T_v(\Phi_v \bar{v})), \text{ and } \Phi_u \bar{u} = \Pi_u(T_u(\Phi_u \bar{u})), \tag{12}$$

where  $\Pi_v$  and  $\Pi_u$  project into the linear spaces spanned by the columns of  $\Phi_v$  and  $\Phi_u$ , respectively.

We now describe the TD algorithm that updates the critic parameters corresponding to the value and square value functions (Note that we require critic estimates for both the unperturbed as well as the perturbed policy parameters). This algorithm is an extension of the algorithm proposed by Tamar et al. (2013b) to the discounted setting. Recall from Algorithm 1 that, at any instant  $n$ , the TD-critic runs two  $m_n$  length trajectories corresponding to policy parameters  $\theta_n$  and  $\theta_n + \delta \Delta_n$ .

**Critic update** Calculate the temporal difference (TD)-errors  $\delta_m, \delta_m^+$  for the value and  $\epsilon_m, \epsilon_m^+$  for the square value functions using (15), and update the critic parameters  $v_m, v_m^+$  for the value and  $u_m, u_m^+$  for the square value functions as follows:

**Unperturbed:**

$$v_{m+1} = v_m + \zeta_3(m)\delta_m\phi_v(x_m), \quad u_{m+1} = u_m + \zeta_3(m)\epsilon_m\phi_u(x_m), \tag{13}$$

**Perturbed:**

$$v_{m+1}^+ = v_m^+ + \zeta_3(m)\delta_m^+\phi_v(x_m^+), \quad u_{m+1}^+ = u_m^+ + \zeta_3(m)\epsilon_m^+\phi_u(x_m^+), \tag{14}$$

where the TD-errors  $\delta_m, \delta_m^+, \epsilon_m, \epsilon_m^+$  in (13) are computed as

**Unperturbed:**

$$\begin{aligned} \delta_m &= R(x_m, a_m) + \gamma v_m^\top \phi_v(x_{m+1}) - v_m^\top \phi_v(x_m), \\ \epsilon_m &= R(x_m, a_m)^2 + 2\gamma R(x_m, a_m)v_m^\top \phi_v(x_{m+1}) + \gamma^2 u_m^\top \phi_u(x_{m+1}) - u_m^\top \phi_u(x_m), \end{aligned} \tag{15}$$

**Perturbed:**

$$\begin{aligned} \delta_m^+ &= R(x_m^+, a_m^+) + \gamma v_m^{+\top} \phi_v(x_{m+1}^+) - v_m^{+\top} \phi_v(x_m^+), \\ \epsilon_m^+ &= R(x_m^+, a_m^+)^2 + 2\gamma R(x_m^+, a_m^+)v_m^{+\top} \phi_v(x_{m+1}^+) + \gamma^2 u_m^{+\top} \phi_u(x_{m+1}^+) \\ &\quad - u_m^{+\top} \phi_u(x_m^+). \end{aligned} \tag{16}$$

Note that the TD-error  $\epsilon$  for the square value function  $U$  comes directly from its Bellman Eq. (2). Theorem 2 in Sect. 7 establishes that the critic parameters  $(v_n, u_n)$  governed by (13) converge to the solutions  $(\bar{v}, \bar{u})$  of the fixed point Eq. (12).

### 4.2.1 Convergence rate

Let  $v_{\min} = \min(v_v, v_u)$ , where  $v_v$  and  $v_u$  are minimum eigenvalues of  $\Phi_v^T D^\theta \Phi_v$  and  $\Phi_u^T D^\theta \Phi_u$ , respectively. Recall that  $D^\theta$  denotes the stationary distribution of the underlying policy  $\theta$ . From (A2), (A3) and the fact that we consider finite state-spaces, we have that  $v_{\min} > 0$ .

From recent results in [Korda and Prashanth \(2015\)](#) that provide non-asymptotic bounds for TD(0) with function approximation, we know that the canonical  $O(m^{-1/2})$  rate can be achieved under the appropriate choice of the step-size  $\zeta_3(m)$ . The following rate result is crucial in setting the trajectory lengths  $m_n$  and relating them to perturbation constants  $\beta_n$  [see (A4) in the next section]:

**Theorem 1** *Under (A2)–(A3), choosing  $\zeta_3(m) = \frac{c_0 c}{(c+m)}$ , with  $c_0 < v_{\min}(1-\gamma)/(2(1+\gamma)^2)$  and  $c$  such that  $v_{\min}(1-\gamma)c_0 c > 1$ , we have,*

$$\mathbb{E} \|v_m - \bar{v}\|_2 \leq \frac{K_1(m)}{\sqrt{m+c}} \quad \text{and} \quad \mathbb{E} \|u_m - \bar{u}\|_2 \leq \frac{K_2(m)}{\sqrt{m+c}},$$

where  $K_1(m)$  and  $K_2(m)$  are  $O(1)$ .

*Proof* The first claim follows directly from Theorem 1 in [Korda and Prashanth \(2015\)](#), while the second claim can be proven in an analogous manner as the first.  $\square$

The above rate result holds only if the step-size is set using  $v_{\min}$  and the latter quantity is unknown in a typical RL setting. However, a standard trick to overcome this dependence while obtaining the same convergence rate is to employ iterate averaging, proposed independently by [Polyak and Juditsky \(1992\)](#) and [Ruppert \(1991\)](#). The latter approach involves using a larger step-size  $\Theta(1/n^{\zeta_1})$  with  $\zeta_1 \in (1/2, 1)$  and couple this with averaging of iterates. An iterate averaged variant of Theorem 1 can be claimed and we refer the reader to Theorem 2 of [Korda and Prashanth \(2015\)](#) for further details.

### 4.3 First-order algorithms: RS-SPSA-G and RS-SF-G

SPSA-based estimate for  $\nabla V^\theta(x^0)$ , and similarly for  $\nabla U^\theta(x^0)$ , is given by

$$\nabla_i \widehat{V}^{\theta_n}(x^0) \approx \frac{\widehat{V}^{\theta_n + \beta_n \Delta_n}(x^0) - \widehat{V}^{\theta_n}(x^0)}{\beta_n \Delta^{(i)}}, \quad i = 1, \dots, \kappa_1, \tag{17}$$

where  $\beta_n$  are perturbation constants that vanish asymptotically [see (A4) at the end of this section] and  $\Delta_n$  is a vector of independent Rademacher random variables, for all  $n = 1, 2, \dots$ . The advantage of this estimator is that it perturbs all directions at the same time (the numerator is identical in all  $\kappa_1$  components). So, the number of function measurements needed for this estimator is always two, independent of the dimension  $\kappa_1$ . However, unlike the SPSA estimates in [Spall \(1992\)](#) that use two-sided balanced estimates (simulations with parameters  $\theta_n - \beta_n \Delta_n$  and  $\theta + \beta \Delta$ ), our gradient estimates are one-sided (simulations with parameters  $\theta_n$  and  $\theta_n + \beta_n \Delta_n$ ) and resemble those in [Chen et al. \(1999\)](#). The use of one-sided estimates is primarily because the updates of the Lagrangian parameter require a simulation with the running parameter  $\theta_n$ . Using a balanced gradient estimate would therefore come at the cost of an additional simulation (the resulting procedure would then require three simulations), which we avoid by using one-sided gradient estimates.

SF-based method estimates not the gradient of a function  $H(\theta_n)$  itself, but rather the convolution of  $\nabla H(\theta_n)$  with the Gaussian density function  $\mathcal{N}(\mathbf{0}, \beta_n^2 \mathbf{I})$ , i.e.,

$$\begin{aligned} C_{\beta_n} H(\theta_n) &= \int \mathcal{G}_{\beta_n}(\theta_n - z) \nabla_z H(z) dz = \int \nabla_z \mathcal{G}_{\beta_n}(z) H(\theta_n - z) dz \\ &= \frac{1}{\beta_n} \int -z' \mathcal{G}_1(z') H(\theta_n - \beta_n z') dz', \end{aligned}$$

where  $\mathcal{G}_{\beta_n}$  is the  $\kappa_1$ -dimensional Gaussian p.d.f. The first equality above follows by using integration by parts and the second one by using the fact that  $\nabla_z \mathcal{G}_{\beta_n}(z) = \frac{-z}{\beta_n^2} \mathcal{G}_{\beta_n}(z)$  and by substituting  $z' = z/\beta_n$ . As  $\beta_n \rightarrow 0$ , it can be seen that  $C_{\beta_n} H(\theta_n)$  converges to  $\nabla H(\theta_n)$  [see Chapter 6 of Bhatnagar et al. (2013)]. Thus, a one-sided SF estimate of  $\nabla V^{\theta_n}(x^0)$  is given by

$$\nabla_i \widehat{V}^{\theta_n}(x^0) \approx \frac{\Delta_n^{(i)}}{\beta_n} (\widehat{V}^{\theta_n + \beta_n \Delta_n}(x^0) - \widehat{V}^{\theta_n}(x^0)), \quad i = 1, \dots, \kappa_1, \tag{18}$$

where  $\Delta_n$  is a vector of independent Gaussian  $\mathcal{N}(0, 1)$  random variables. The reasons for using the one-sided estimate in (18) are as follows: (i) the estimate in (18) has lower bias when compared to a one simulation estimate that does not use  $\widehat{V}^{\theta_n}(x^0)$  and (ii) for updating the Lagrange multiplier  $\lambda$ , we require a trajectory of the MDP corresponding to policy  $\theta_n$  and this trajectory can be used to estimate  $\widehat{V}^{\theta_n}(x^0)$ .

**Actor update** Estimate the gradients  $\nabla V^{\theta}(x^0)$  and  $\nabla U^{\theta}(x^0)$  using SPSA (17) or SF (18) and update the policy parameter  $\theta$  as follows<sup>5</sup>: For  $i = 1, \dots, \kappa_1$ ,

**RS-SPSA-G:**

$$\begin{aligned} \theta_{n+1}^{(i)} &= \Gamma_i \left[ \theta_n^{(i)} + \frac{\zeta_2(n)}{\beta_n \Delta_n^{(i)}} \left( (1 + 2\lambda_n v_n^T \phi_v(x^0)) (v_n^+ - v_n)^T \phi_v(x^0) \right. \right. \\ &\quad \left. \left. - \lambda_n (u_n^+ - u_n)^T \phi_u(x^0) \right) \right], \end{aligned} \tag{19}$$

**RS-SF-G:**

$$\begin{aligned} \theta_{n+1}^{(i)} &= \Gamma_i \left[ \theta_n^{(i)} + \frac{\zeta_2(n) \Delta_n^{(i)}}{\beta_n} \left( (1 + 2\lambda_n v_n^T \phi_v(x^0)) (v_n^+ - v_n)^T \phi_v(x^0) \right. \right. \\ &\quad \left. \left. - \lambda_n (u_n^+ - u_n)^T \phi_u(x^0) \right) \right]. \end{aligned} \tag{20}$$

For both SPSA and SF variants, the Lagrange multiplier  $\lambda$  is updated as follows:

$$\lambda_{n+1} = \Gamma_\lambda \left[ \lambda_n + \zeta_1(n) \left( u_n^T \phi_u(x^0) - (v_n^T \phi_v(x^0))^2 - \alpha \right) \right]. \tag{21}$$

In the above, note the following:

- (i)  $\beta_n \geq 0$  and vanish asymptotically [see (A4) below for the precise condition];
- (ii)  $\Delta_n^{(i)}$ 's are independent Rademacher and Gaussian  $\mathcal{N}(0, 1)$  random variables in SPSA and SF updates, respectively;
- (iii)  $\Gamma$  and  $\Gamma_\lambda$  are projection operators that keep the iterates  $(\theta_n, \lambda_n)$  stable and were defined in Sect. 4.1. These projection operators are necessary to keep the iterates stable and hence, ensure convergence of the algorithms.

<sup>5</sup> By an abuse of notation, we use  $v_n$  (resp.  $v_n^+, u_n, u_n^+$ ) to denote the critic parameter  $v_{m_n}$  (resp.  $v_{m_n}^+, u_{m_n}, u_{m_n}^+$ ) obtained at the end of a  $m_n$  length trajectory.



4.3.1 Choosing trajectory length  $m_n$ , perturbation constants  $\beta_n$  and step-sizes  $\zeta_3(n), \zeta_2(n), \zeta_1(n)$

We make the following assumption on the step-size schedules:

(A4) The step size schedules  $\{\zeta_2(n)\}$ , and  $\{\zeta_1(n)\}$  satisfy

$$\zeta_2(n), \beta_n \rightarrow 0, \frac{1}{\sqrt{m_n \beta_n}} \rightarrow 0, \tag{22}$$

$$\sum_n \zeta_1(n) = \sum_n \zeta_2(n) = \infty, \tag{23}$$

$$\sum_n \zeta_1(n)^2, \sum_n \frac{\zeta_2(n)^2}{\beta_n^2}, < \infty, \tag{24}$$

$$\zeta_1(n) = o(\zeta_2(n)). \tag{25}$$

Equations (23) and (24) are standard step-size conditions in stochastic approximation algorithms, and Equation (25) ensures that the policy parameter update is on the faster time-scale  $\{\zeta_2(n)\}$ , and the Lagrange multiplier update is on the slower time-scale  $\{\zeta_1(n)\}$ .

Equation (22) is motivated by a similar condition in Prashanth et al. (2016) and ensures that the bias from a finite length ( $m_n$ ) trajectory run of TD-critic can be ignored. A simple setting that ensures (22) is to have  $m_n = C_1 n^{\zeta_2}$  and  $\beta_n = C_2 n^{-\zeta_3}$ , where  $C_1, C_2$  are constants and  $\zeta_2, \zeta_3 > 0$  with  $\zeta_3 > \zeta_2/2$ . This ensures that the trajectories increase in length as a function of outer loop index  $n$ , at a rate that is sufficient to cancel the bias induced by the TD-critic. See Lemma 6 in Sect. 7 makes this claim precise, in particular justifying the need for (22) in (A4).

We provide a proof of convergence of the first-order SPSA and SF algorithms to a tuple  $(\theta^{\lambda^*}, \lambda^*)$ , which is a (local) saddle point of the risk-sensitive objective function  $\widehat{L}(\theta, \lambda) \triangleq -\widehat{V}^\theta(x^0) + \lambda(\widehat{A}^\theta(x^0) - \alpha)$ , where  $\widehat{V}^\theta(x^0) = \bar{v}^\top \phi_v(x^0)$  and  $\widehat{A}^\theta(x^0) = \bar{u}^\top \phi_u(x^0) - (\bar{v}^\top \phi_v(x^0))^2$  with  $\bar{v}$  and  $\bar{u}$  defined by (12). Further, the limit  $\theta^{\lambda^*}$  satisfies the variance constraint, i.e.,  $\widehat{A}^{\theta^{\lambda^*}}(x^0) \leq \alpha$ . See Theorems 3, 4, 5 and Proposition 1 in Sect. 7 for details.

**Remark 3 (Extension to Sharpe ratio optimization)** The gradient of Sharpe ratio (SR),  $S(\theta)$ , in the discounted setting is given by

$$\nabla S(\theta) = \frac{1}{\sqrt{A^\theta(x^0)}} \left( \nabla V^\theta(x^0) - \frac{V^\theta(x^0)}{2A^\theta(x^0)} \nabla A^\theta(x^0) \right).$$

The actor recursions for the variants of the RS-SPSA-G and RS-SF-G algorithms that optimize the SR objective are as follows:

**RS-SPSA-G**

$$\theta_{n+1}^{(i)} = \Gamma_i \left( \theta_n^{(i)} + \frac{\zeta_2(n)}{\sqrt{u_n^\top \phi_u(x^0) - (v_n^\top \phi_v(x^0))^2} \beta_n \Delta_n^{(i)}} \left( (v_n^+ - v_n)^\top \phi_v(x^0) - \frac{v_n^\top \phi_v(x^0) ((u_n^+ - u_n)^\top \phi_u(x^0) - 2v_n^\top \phi_v(x^0)(v_n^+ - v_n)^\top \phi_v(x^0))}{2(u_n^\top \phi_u(x^0) - (v_n^\top \phi_v(x^0))^2)} \right) \right). \tag{26}$$

**RS-SF-G**

$$\theta_{n+1}^{(i)} = \Gamma_i \left( \theta_n^{(i)} + \frac{\zeta_2(n)\Delta_n^{(i)}}{\beta_n \sqrt{u_n^\top \phi_u(x^0) - (v_n^\top \phi_v(x^0))^2}} \left( (v_n^+ - v_n)^\top \phi_v(x^0) - \frac{v_n^\top \phi_v(x^0) ((u_n^+ - u_n)^\top \phi_u(x^0) - 2v_n^\top \phi_v(x^0)(v_n^+ - v_n)^\top \phi_v(x^0))}{2 \left( u_n^\top \phi_u(x^0) - (v_n^\top \phi_v(x^0))^2 \right)} \right) \right). \quad (27)$$

Note that only the actor recursion changes for SR optimization, while the rest of the updates that include the critic recursions for nominal and perturbed parameters remain the same as before in the SPSA and SF based algorithms. Further, SR optimization does not involve the Lagrange parameter  $\lambda$ , and thus, the proposed actor-critic algorithms are two time-scale (instead of three time-scale as in the described algorithms) stochastic approximation algorithms in this case.

*Remark 4 (One-simulation SR variant)* For the SR objective, the proposed algorithms can be modified to work with only one simulated trajectory of the system. This is because in the SR case, we do not require the Lagrange multiplier  $\lambda$ , and thus, the simulated trajectory corresponding to the nominal policy parameter  $\theta$  is not necessary. In this implementation, the gradient is estimated as  $\nabla_i S(\theta) \approx S(\theta + \beta\Delta)/\beta\Delta^{(i)}$  for SPSA and as  $\nabla_i S(\theta) \approx (\Delta^{(i)}/\beta)S(\theta + \beta\Delta)$  for SF.

*Remark 5 (Monte-Carlo critic)* In the above algorithms, the critic uses a TD method to evaluate the policies. These algorithms can be implemented with a Monte-Carlo critic that at each time instant  $n$  computes a sample average of the total discounted rewards corresponding to the nominal  $\theta_n$  and perturbed  $\theta_n + \beta\Delta_n$  policy parameter. This implementation would be similar to that in Tamar et al. (2012), except here we use simultaneous perturbation methods to estimate the gradient.

**4.4 Second-order algorithms: RS-SPSA-N and RS-SF-N**

Recall from Sect. 4.1 that a second-order scheme updates the policy parameter in the following manner:

$$\theta_{n+1} = \Gamma[\theta_n - \zeta_2(n)\nabla_\theta^2 L(\theta, \lambda)^{-1}\nabla L(\theta, \lambda)]. \quad (28)$$

From the above, it is evident that for any second-order method, an estimate of the Hessian  $\nabla_\theta^2 L(\theta, \lambda)$  of the Lagrangian is necessary, in addition to an estimate of the gradient  $\nabla L(\theta, \lambda)$ . As in the case of the gradient based schemes outlined earlier, we employ the simultaneous perturbation technique to develop these estimates. The first algorithm, henceforth referred to as RS-SPSA-N, uses SPSA for the gradient/Hessian estimates. On the other hand, the second algorithm, henceforth referred to as RS-SF-N, uses a smoothed functional (SF) approach for the gradient/Hessian estimates. As confirmed by our numerical experiments, second order methods are in general more accurate, though at the cost of inverting the Hessian matrix in each step.

*4.4.1 RS-SPSA-N algorithm*

The Hessian w.r.t.  $\theta$  of  $L(\theta, \lambda)$  can be written as follows:

$$\nabla_\theta^2 L(\theta, \lambda) = -\nabla_\theta^2 V^\theta(x^0) + \lambda \nabla_\theta^2 A^\theta(x^0)$$

$$= -\nabla^2 V^\theta(x^0) + \lambda (\nabla^2 U^\theta(x^0) - 2V^\theta(x^0)\nabla^2 V^\theta(x^0) - 2\nabla V^\theta(x^0)\nabla V^\theta(x^0)^\top). \tag{29}$$

**Critic update** As in the case of the gradient based schemes, we run two simulations. However, perturbed simulation here corresponds to the policy parameter  $\theta_n + \beta_n(\Delta_n + \widehat{\Delta}_n)$ , where  $\Delta_n$  and  $\widehat{\Delta}_n$  represent vectors of independent  $\kappa_1$ -dimensional Rademacher random variables. The critic parameters  $v_n, u_n$  from unperturbed simulation and  $v_n^+, u_n^+$  from perturbed simulation are updated as described earlier in Sect. 4.2.

**Gradient and Hessian estimates** Using an SPSA-based estimation technique [see Chapter 7 of [Bhatnagar et al. \(2013\)](#)], the gradient and Hessian of the value function  $V$ , and similarly of the square value function  $U$ , are estimated as follows: For  $i = 1, \dots, \kappa_1$ ,

$$\begin{aligned} \nabla_i \widehat{V}^\theta(x^0) &\approx \frac{\widehat{V}^{\theta+\beta_n(\Delta+\widehat{\Delta})}(x^0) - \widehat{V}^\theta(x^0)}{\beta_n \Delta^{(i)}} = \frac{(v_n^+ - v_n)^\top \phi_v(x^0)}{\beta_n \Delta^{(i)}}, \\ \nabla_{i,j}^2 \widehat{V}^\theta(x^0) &\approx \frac{\widehat{V}^{\theta+\beta_n(\Delta+\widehat{\Delta})}(x^0) - \widehat{V}^\theta(x^0)}{\beta_n^2 \Delta^{(i)} \widehat{\Delta}^{(j)}} = \frac{(v_n^+ - v_n)^\top \phi_v(x^0)}{\beta_n^2 \Delta^{(i)} \widehat{\Delta}^{(j)}}. \end{aligned}$$

As in the case of the first order algorithms, the TD-critic trajectory lengths are chosen such that there is no bias in the value estimates, when viewed from the actor-recursion. Next, using suitable Taylor expansions and observe that the bias terms vanish as  $\Delta_n, \widehat{\Delta}_n$ , being Rademacher, are zero-mean—see Lemma 7 in Sect. 7 for details. As in the case of RS-SPSA, this is an one-sided estimate with the unperturbed simulation required for updating the Lagrange multiplier.

**Hessian update** Using the critic values from the two simulations, we estimate the Hessian  $\nabla_\theta^2 L(\theta, \lambda)$  as follows: Let  $H_n^{(i,j)}$  denote the  $n$ th estimate of the  $(i, j)$ th element of the Hessian. Then, for  $i, j = 1, \dots, \kappa_1$ , with  $i \leq j$ , the update is

$$\begin{aligned} H_{n+1}^{(i,j)} &= H_n^{(i,j)} + \zeta'_2(n) \left[ \frac{(1 + \lambda_n(v_n + v_n^+)^\top \phi_v(x^0))(v_n - v_n^+)^\top \phi_v(x^0)}{\beta_n^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}} \right. \\ &\quad \left. + \frac{\lambda_n(u_n^+ - u_n)^\top \phi_u(x^0)}{\beta_n^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}} - H_n^{(i,j)} \right], \tag{30} \end{aligned}$$

and for  $i > j$ , we simply set  $H_{n+1}^{(i,j)} = H_{n+1}^{(j,i)}$ . In the above, the step-size  $\zeta'_2(n)$  satisfies

$$\sum_n \zeta'_2(n) = \infty; \sum_n \zeta'^2_2(n) < \infty, \frac{\zeta_2(n)}{\zeta'_2(n)} \rightarrow 0 \text{ as } n \rightarrow \infty.$$

The last condition above ensures that the Hessian update proceeds on a faster timescale in comparison to the  $\theta$ -recursion [see (31) below]. Finally, we set  $H_{n+1} = \Upsilon([H_{n+1}^{(i,j)}]_{i,j=1}^{\kappa_1})$ , where  $\Upsilon(\cdot)$  denotes an operator that projects a square matrix onto the set of symmetric and positive definite matrices. This projection is a standard requirement to ensure convergence of  $H_n$  to the Hessian  $\nabla_\theta^2 L(\theta, \lambda)$  and we state the following standard assumption (cf. [Bhatnagar et al. 2013](#), Chapter 7) on this operator:

**(A5)** For any sequence of matrices  $\{A_n\}$  and  $\{B_n\}$  in  $\mathcal{R}^{\kappa_1 \times \kappa_1}$  such that  $\lim_{n \rightarrow \infty} \|A_n - B_n\| = 0$ , the  $\Upsilon$  operator satisfies  $\lim_{n \rightarrow \infty} \|\Upsilon(A_n) - \Upsilon(B_n)\| = 0$ . Further, for any sequence of matrices  $\{C_n\}$  in  $\mathcal{R}^{\kappa_1 \times \kappa_1}$ , we have

$$\sup_n \|C_n\| < \infty \Rightarrow \sup_n \|\Upsilon(C_n)\| < \infty \text{ and } \sup_n \|\{ \Upsilon(C_n) \}^{-1}\| < \infty.$$

As suggested in Gill et al. (1981), a possible definition of  $\mathcal{Y}$  is to perform an eigen-decomposition of  $H_n$  and then make all eigenvalues positive. This avoids singularity of  $H_n$  and also satisfies the above assumption. In our experiments, we use this scheme for projecting  $H_n$ .

**Actor update** Let  $M_n \triangleq H_n^{-1}$  denote the inverse of the the Hessian estimate  $H_n$ . We incorporate a Newton decrement to update the policy parameter  $\theta$  as follows:

$$\theta_{n+1}^{(i)} = \Gamma_i \left[ \theta_n^{(i)} + \zeta_2(n) \sum_{j=1}^{\kappa_1} M_n^{(i,j)} \left( \frac{(1 + 2\lambda_n v_n^\top \phi_v(x^0))(v_n^+ - v_n)^\top \phi_v(x^0)}{\beta_n \Delta_n^{(j)}} - \frac{\lambda_n (u_n^+ - u_n)^\top \phi_u(x^0)}{\beta_n \Delta_n^{(j)}} \right) \right]. \tag{31}$$

In the long run,  $M_n$  converges to  $\nabla_\theta^2 L(\theta, \lambda)^{-1}$ , while the last term in the brackets in (31) converges to  $\nabla L(\theta, \lambda)$  and hence, the update (31) can be seen to descend in  $\theta$  using a Newton decrement. Note that the Lagrange multiplier update here is the same as that in RS-SPSA-G.

#### 4.4.2 RS-SF-N algorithm

**Gradient and Hessian Estimates** While the gradient estimate here is the same as that in the RS-SF-G algorithm, the Hessian is estimated as follows: Recall that  $\Delta_n = (\Delta_n^{(1)}, \dots, \Delta_n^{(\kappa_1)})^\top$  is a vector of mutually independent  $\mathcal{N}(0, 1)$  random variables. Let  $\tilde{H}(\Delta_n)$  be a  $\kappa_1 \times \kappa_1$  matrix defined as

$$\tilde{H}(\Delta_n) \triangleq \begin{bmatrix} (\Delta_n^{(1)^2} - 1) & \Delta_n^{(1)} \Delta_n^{(2)} & \dots & \Delta_n^{(1)} \Delta_n^{(\kappa_1)} \\ \Delta_n^{(2)} \Delta_n^{(1)} & (\Delta_n^{(2)^2} - 1) & \dots & \Delta_n^{(2)} \Delta_n^{(\kappa_1)} \\ \dots & \dots & \dots & \dots \\ \Delta_n^{(\kappa_1)} \Delta_n^{(1)} & \Delta_n^{(\kappa_1)} \Delta_n^{(2)} & \dots & (\Delta_n^{(\kappa_1)^2} - 1) \end{bmatrix}. \tag{32}$$

Then, the Hessian  $\nabla_\theta^2 L(\theta, \lambda)$  is approximated as

$$\nabla_\theta^2 L(\theta, \lambda) \approx \frac{1}{\beta_n^2} \left[ \tilde{H}(\Delta) (L(\theta + \beta \Delta, \lambda) - L(\theta, \lambda)) \right]. \tag{33}$$

The correctness of the above estimate in the limit as  $\beta_n \rightarrow 0$  can be seen from Lemma 8 in the Appendix. The main idea involves convolving the Hessian with a Gaussian density function (similar to RS-SF) and then performing integration by parts twice.

**Critic update** As in the case of the RS-SF-G algorithm, we run two simulations with unperturbed and perturbed policy parameters, respectively. Recall that the perturbed simulation corresponds to the policy parameter  $\theta_n + \beta_n \Delta_n$ , where  $\Delta_n$  represent a vector of independent  $\kappa_1$ -dimensional Gaussian  $\mathcal{N}(0, 1)$  random variables. The critic parameters for both these simulations are updated as described earlier in Sect. 4.2.

**Hessian update** As in RS-SPSA-N, let  $H_n^{(i,j)}$  denote the  $(i, j)$ th element of the Hessian estimate  $H_n$  at time step  $t$ . Using (33), we devise the following update rule for the Hessian estimate  $H_n$ : For  $i, j, k = 1, \dots, \kappa_1, j < k$ , the update is

$$H_{t+1}^{(i,i)} = H_n^{(i,i)} + \zeta'_2(n) \left[ \frac{(\Delta_n^{(i)^2} - 1)}{\beta_n^2} \left( (1 + \lambda_n (v_n + v_n^+)^\top \phi_v(x^0))(v_n - v_n^+)^\top \phi_v(x^0) + \lambda_n (u_n^+ - u_n)^\top \phi_u(x^0) \right) - H_n^{(i,i)} \right], \tag{34}$$

$$H_{t+1}^{(j,k)} = H_n^{(j,k)} + \zeta'_2(n) \left[ \frac{\Delta_n^{(j)} \Delta_n^{(k)}}{\beta_n^2} \left( (1 + \lambda_n(v_n + v_n^+)^\top \phi_v(x^0))(v_n - v_n^+)^\top \phi_v(x^0) + \lambda_n(u_n^+ - u_n)^\top \phi_u(x^0) \right) - H_n^{(j,k)} \right], \tag{35}$$

and for  $j > k$ , we set  $H_{n+1}^{(j,k)} = H_{n+1}^{(k,j)}$ . The step-size  $\zeta'_2(n)$  is as in RS-SPSA-N. Further, as in the latter algorithm, we set  $H_{n+1} = \mathcal{Y}([H_{n+1}^{(i,j)}]_{i,j=1}^{|\kappa_1|})$  and let  $M_{n+1} \triangleq H_{n+1}^{-1}$  denote its inverse.

**Actor update** Using the gradient and Hessian estimates from the above, we update the policy parameter  $\theta$  as follows:

$$\theta_{n+1}^{(i)} = \Gamma_i \left[ \theta_n^{(i)} + \zeta_2(n) \sum_{j=1}^{\kappa_1} M_n^{(i,j)} \frac{\Delta_n^{(j)}}{\beta_n} \left( (1 + 2\lambda_n v_n^\top \phi_v(x^0))(v_n^+ - v_n)^\top \phi_v(x^0) - \lambda_n(u_n^+ - u_n)^\top \phi_u(x^0) \right) \right]. \tag{36}$$

As in the case of RS-SPSA-N, it can be seen that the above update rule is equivalent to descent with a Newton decrement, since  $M_n$  converges to  $\nabla_\theta^2 L(\theta, \lambda)^{-1}$ , and the last term in the brackets in (36) converges to  $\nabla L(\theta, \lambda)$ . The Lagrange multiplier  $\lambda$  update here is the same as that in RS-SF-G.

*Remark 6* The second-order variants of the algorithms for SR optimization can be worked out along similar lines as outlined in Sect. 4.4 and the details are omitted here.

### 5 Average reward setting

The average reward under policy  $\mu$  is defined as

$$\rho(\mu) = \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{n=0}^{T-1} R_n \mid \mu \right] = \sum_{x,a} d^\mu(x) \mu(a|x) r(x, a) = \sum_{x,a} \pi^\mu(x, a) r(x, a),$$

where  $d^\mu$  and  $\pi^\mu$  are the stationary distributions of policy  $\mu$  over states and state-action pairs, respectively (see Sect. 2). The goal in the standard (risk-neutral) average reward formulation is to find an *average optimal* policy, i.e.,  $\mu^* = \arg \max_\mu \rho(\mu)$ . For all states  $x \in \mathcal{X}$  and actions  $a \in \mathcal{A}$ , the *differential* action-value and value functions of policy  $\mu$  are defined respectively as

$$Q^\mu(x, a) = \sum_{n=0}^{\infty} \mathbb{E}[R_n - \rho(\mu) \mid x_0 = x, a_0 = a, \mu],$$

$$V^\mu(x) = \sum_a \mu(a|x) Q^\mu(x, a).$$

These functions satisfy the following Poisson equations (Puterman 1994)

$$\rho(\mu) + V^\mu(x) = \sum_a \mu(a|x) \left[ r(x, a) + \sum_{x'} P(x'|x, a) V^\mu(x') \right], \tag{37}$$

$$\rho(\mu) + Q^\mu(x, a) = r(x, a) + \sum_{x'} P(x'|x, a) V^\mu(x'). \tag{38}$$

In the context of risk-sensitive MDPs, different criteria have been proposed to define a measure of *variability* in the average reward setting, among which we consider the *long-run variance* of  $\mu$  (Filar et al. 1989) defined as

$$\Lambda(\mu) = \sum_{x,a} \pi^\mu(x, a) [r(x, a) - \rho(\mu)]^2 = \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{n=0}^{T-1} (R_n - \rho(\mu))^2 \middle| \mu \right]. \tag{39}$$

This notion of variability is based on the observation that it is the frequency of occurrence of state-action pairs that determine the variability in the average reward. It is easy to show that

$$\Lambda(\mu) = \eta(\mu) - \rho(\mu)^2, \quad \text{where} \quad \eta(\mu) = \sum_{x,a} \pi^\mu(x, a) r(x, a)^2.$$

We consider the following risk-sensitive measure for average reward MDPs in this paper:

$$\max_{\theta} \rho(\theta) \quad \text{subject to} \quad \Lambda(\theta) \leq \alpha, \tag{40}$$

for a given  $\alpha > 0$ .<sup>6</sup> As in the discounted setting, we employ the Lagrangian relaxation procedure to convert (40) to the unconstrained problem

$$\max_{\lambda} \min_{\theta} \left( L(\theta, \lambda) \triangleq -\rho(\theta) + \lambda(\Lambda(\theta) - \alpha) \right).$$

As in the discounted setting, we descend in  $\theta$  using  $\nabla L(\theta, \lambda) = -\nabla \rho(\theta) + \lambda \nabla \Lambda(\theta)$  and ascend in  $\lambda$  using  $\nabla_{\lambda} L(\theta, \lambda) = \Lambda(\theta) - \alpha$ , to find the saddle point of  $L(\theta, \lambda)$ . Since  $\nabla \Lambda(\theta) = \nabla \eta(\theta) - 2\rho(\theta) \nabla \rho(\theta)$ , in order to compute  $\nabla \Lambda(\theta)$  it would be enough to calculate  $\nabla \rho(\theta)$  and  $\nabla \eta(\theta)$ . Let  $U^\mu$  and  $W^\mu$  denote the differential value and action-value functions associated with the square reward under policy  $\mu$ , respectively. These two quantities satisfy the following Poisson equations:

$$\begin{aligned} \eta(\mu) + U^\mu(x) &= \sum_a \mu(a|x) [r(x, a)^2 + \sum_{x'} P(x'|x, a) U^\mu(x')], \\ \rho(\mu) + W^\mu(x, a) &= r(x, a)^2 + \sum_{x'} P(x'|x, a) U^\mu(x'). \end{aligned} \tag{41}$$

The gradients of  $\rho(\theta)$  and  $\eta(\theta)$  are given by the following lemma:

**Lemma 3** *Under (A1) and (A2), we have*

$$\nabla \rho(\theta) = \sum_{x,a} \pi^\theta(x, a) \nabla \log \mu(a|x; \theta) Q(x, a; \theta), \tag{42}$$

$$\nabla \eta(\theta) = \sum_{x,a} \pi^\theta(x, a) \nabla \log \mu(a|x; \theta) W(x, a; \theta). \tag{43}$$

*Proof* The proof of  $\nabla \rho(\theta)$  can be found in the literature (e.g., Sutton et al. 2000; Konda and Tsitsiklis 2000). To prove  $\nabla \eta(\theta)$ , we start by the fact that from (41), we have  $U(x) = \sum_a \mu(x|a) W(x, a)$ . If we take the derivative w.r.t.  $\theta$  from both sides of this equation, we obtain

<sup>6</sup> Similar to the discounted setting, the risk-sensitive average reward algorithm proposed in this paper can be easily extended to other risk measures based on the long-term variance of  $\mu$ , including the Sharpe ratio (SR), i.e.,  $\max_{\theta} \rho(\theta) / \sqrt{\Lambda(\theta)}$ . The extension to SR will be described in more details in Sect. 3.

$$\begin{aligned}
 \nabla U(x) &= \sum_a \nabla \mu(x|a)W(x, a) + \sum_a \mu(x|a)\nabla W(x, a) \\
 &= \sum_a \nabla \mu(x|a)W(x, a) + \sum_a \mu(x|a)\nabla(r(x, a)^2 - \eta + \sum_{x'} P(x'|x, a)U(x')) \\
 &= \sum_a \nabla \mu(x|a)W(x, a) - \nabla \eta + \sum_{a,x'} \mu(a|x)P(x'|x, a)\nabla U(x'). \tag{44}
 \end{aligned}$$

The second equality is by replacing  $W(x, a)$  from (41). Now if we take the weighted sum, weighted by  $d^\mu(x) = \mathbf{D}^\theta(x)$ , from both sides of (44), we have

$$\begin{aligned}
 \sum_x d^\mu(x)\nabla U(x) &= \sum_{x,a} d^\mu(x)\nabla \mu(a|x)W(x, a) - \nabla \eta \\
 &\quad + \sum_{x,a,x'} d^\mu(x)\mu(a|x)P(x'|x, a)\nabla U(x'). \tag{45}
 \end{aligned}$$

The claim follows from the fact that the last sum on the RHS of (45) is equal to  $\sum_x d^\mu(x)\nabla U(x)$ . □

Note that (43) for calculating  $\nabla \eta(\theta)$  has close resemblance to (42) for  $\nabla \rho(\theta)$ , and thus, similar to what we have for (42), any function  $b : \mathcal{X} \rightarrow \mathbb{R}$  can be added or subtracted to  $W(x, a; \theta)$  on the RHS of (43) without changing the result of the integral (see e.g., Bhatnagar et al. 2009a). So, we can replace  $W(x, a; \theta)$  with the square reward advantage function  $B(x, a; \theta) = W(x, a; \theta) - U(x; \theta)$  on the RHS of (43) in the same manner as we can replace  $Q(x, a; \theta)$  with the advantage function  $A(x, a; \theta) = Q(x, a; \theta) - V(x; \theta)$  on the RHS of (42) without changing the result of the integral. We define the temporal difference (TD) errors  $\delta_n$  and  $\epsilon_n$  for the differential value and square value functions as

$$\begin{aligned}
 \delta_n &= R(x_n, a_n) - \widehat{\rho}_{n+1} + \widehat{V}(x_{n+1}) - \widehat{V}(x_n), \\
 \epsilon_n &= R(x_n, a_n)^2 - \widehat{\eta}_{n+1} + \widehat{U}(x_{n+1}) - \widehat{U}(x_n).
 \end{aligned}$$

If  $\widehat{V}$ ,  $\widehat{U}$ ,  $\widehat{\rho}$ , and  $\widehat{\eta}$  are unbiased estimators of  $V^\mu$ ,  $U^\mu$ ,  $\rho(\mu)$ , and  $\eta(\mu)$ , respectively, then we show in Lemma 4 that  $\delta_n$  and  $\epsilon_n$  are unbiased estimates of the advantage functions  $A^\mu$  and  $B^\mu$ , i.e.,  $\mathbb{E}[\delta_n | x_n, a_n, \mu] = A^\mu(x_n, a_n)$  and  $\mathbb{E}[\epsilon_n | x_n, a_n, \mu] = B^\mu(x_n, a_n)$ .

**Lemma 4** *For any given policy  $\mu$ , we have*

$$\mathbb{E}[\delta_n | x_n, a_n, \mu] = A^\mu(x_n, a_n), \quad \mathbb{E}[\epsilon_n | x_n, a_n, \mu] = B^\mu(x_n, a_n).$$

*Proof* The first statement  $\mathbb{E}[\delta_n | x_n, a_n, \mu] = A^\mu(x_n, a_n)$  has been proved in Lemma 3 of Bhatnagar et al. (2009a), so here we only prove the second statement  $\mathbb{E}[\epsilon_n | x_n, a_n, \mu] = B^\mu(x_n, a_n)$ . we may write

$$\begin{aligned}
 \mathbb{E}[\epsilon_n | x_n, a_n, \mu] &= \mathbb{E}[R(x_n, a_n)^2 - \widehat{\eta}_{n+1} + \widehat{U}(x_{n+1}) - \widehat{U}(x_n) | x_n, a_n, \mu] \\
 &= r(x_n, a_n)^2 - \eta(\mu) + \mathbb{E}[\widehat{U}(x_{n+1}) | x_n, a_n, \mu] - U^\mu(x_n) \\
 &= r(x_n, a_n)^2 - \eta(\mu) + \mathbb{E}[\mathbb{E}[\widehat{U}(x_{n+1}) | x_{n+1}, \mu] | x_n, a_n] - U^\mu(x_n) \\
 &= r(x_n, a_n)^2 - \eta(\mu) + \mathbb{E}[\widehat{U}(x_{n+1}) | x_n, a_n] - U^\mu(x_n)
 \end{aligned}$$

$$\begin{aligned}
 &= r(x_n, a_n)^2 - \eta(\mu) + \underbrace{\sum_{x_{n+1} \in \mathcal{X}} P(x_{n+1}|x_n, a_n) U^\mu(x_{n+1}) - U^\mu(x_n)}_{W^\mu(x, a)} \\
 &= B^\mu(x, a).
 \end{aligned}$$

□

From Lemma 4, we notice that  $\delta_n \psi_n$  and  $\epsilon_n \psi_n$  are unbiased estimates of  $\nabla \rho(\mu)$  and  $\nabla \eta(\mu)$ , respectively, where  $\psi_n = \psi(x_n, a_n) = \nabla \log \mu(a_n|x_n)$  is the *compatible* feature (see e.g., Sutton et al. 2000; Peters et al. 2005).

### 6 Average reward risk-sensitive actor-critic algorithm

We now present our risk-sensitive actor-critic algorithm for average reward MDPs. Algorithm 2 presents the complete structure of the algorithm along with the update rules for the average rewards  $\widehat{\rho}_n, \widehat{\eta}_n$ ; TD errors  $\delta_n, \epsilon_n$ ; critic  $v_n, u_n$ ; and actor  $\theta_n, \lambda_n$  parameters. The projection operators  $\Gamma$  and  $\Gamma_\lambda$  are as defined in Sect. 4, and similar to the discounted setting, are necessary for the convergence proof of the algorithm. The step-size schedules satisfy (A3) defined in Sect. 4, plus the step size schedule  $\{\zeta_4(n)\}$  satisfies  $\zeta_4(n) = k\zeta_3(n)$ , for some positive constant  $k$ . This is to ensure that the average and critic updates are on the (same) fastest time-scale  $\{\zeta_4(n)\}$  and  $\{\zeta_3(n)\}$ , the policy parameter update is on the intermediate time-scale  $\{\zeta_2(n)\}$ , and the Lagrange multiplier update is on the slowest time-scale  $\{\zeta_1(n)\}$ . This results in a three time-scale stochastic approximation algorithm.

---

#### Algorithm 2 Template of the Average Reward Risk-Sensitive Actor-Critic Algorithm

---

**Input:** parameterized policy  $\mu(\cdot|\cdot; \theta)$  and value function feature vectors  $\phi_v(\cdot)$  and  $\phi_u(\cdot)$

**Initialization:** policy parameters  $\theta = \theta_0$ ; value function weight vectors  $v = v_0$  and  $u = u_0$ ; initial state  $x_0 \sim P_0(x)$

**for**  $t = 0, 1, 2, \dots$  **do**

Draw action  $a_n \sim \mu(\cdot|x_n; \theta_n)$  and observe the next state  $x_{n+1} \sim P(\cdot|x_n, a_n)$  and the reward  $R(x_n, a_n)$

**Average Updates:**  $\widehat{\rho}_{n+1} = (1 - \zeta_4(n))\widehat{\rho}_n + \zeta_4(n)R(x_n, a_n),$  (46)

$\widehat{\eta}_{n+1} = (1 - \zeta_4(n))\widehat{\eta}_n + \zeta_4(n)R(x_n, a_n)^2$

**TD Errors:**  $\delta_n = R(x_n, a_n) - \widehat{\rho}_{n+1} + v_n^\top \phi_v(x_{n+1}) - v_n^\top \phi_v(x_n)$

$\epsilon_n = R(x_n, a_n)^2 - \widehat{\eta}_{n+1} + u_n^\top \phi_u(x_{n+1}) - u_n^\top \phi_u(x_n)$

**Critic Update:**  $v_{n+1} = v_n + \zeta_3(n)\delta_n \phi_v(x_n), \quad u_{n+1} = u_n + \zeta_3(n)\epsilon_n \phi_u(x_n)$

**Actor Update:**  $\theta_{n+1} = \Gamma\left(\theta_n - \zeta_2(n)\left(-\delta_n \psi_n + \lambda_n(\epsilon_n \psi_n - 2\widehat{\rho}_{n+1}\delta_n \psi_n)\right)\right)$  (47)

$\lambda_{n+1} = \Gamma_\lambda\left(\lambda_n + \zeta_1(n)(\widehat{\eta}_{n+1} - \widehat{\rho}_{n+1}^2 - \alpha)\right)$  (48)

**end for**

**return** policy and value function parameters  $\theta, \lambda, v, u$

---

As in the discounted setting, the critic uses linear approximation for the differential value and square value functions, i.e.,  $\widehat{V}(x) = v^\top \phi_v(x)$  and  $\widehat{U}(x) = u^\top \phi_u(x)$ , where  $\phi_v(\cdot)$  and  $\phi_u(\cdot)$  are feature vectors of size  $\kappa_2$  and  $\kappa_3$ , respectively.



Although our estimates of  $\rho(\theta)$  and  $\eta(\theta)$  are unbiased, since we use biased estimates for  $V^\theta$  and  $U^\theta$  (linear approximations in the critic), our gradient estimates  $\nabla\rho(\theta)$  and  $\nabla\eta(\theta)$ , and as a result  $\nabla L(\theta, \lambda)$ , are biased. The following lemma shows the bias in our estimate of  $\nabla L(\theta, \lambda)$ .

**Lemma 5** *The bias of our actor-critic algorithm in estimating  $\nabla L(\theta, \lambda)$  for fixed  $\theta$  and  $\lambda$  is*

$$\mathcal{B}(\theta, \lambda) = \sum_x \mathbf{D}^\theta(x) \left( - (1 + 2\lambda\rho(\theta)) [\nabla\bar{V}^\theta(x) - \nabla v^{\theta\top}\phi_v(x)] + \lambda [\nabla\bar{U}^\theta(x) - \nabla u^{\theta\top}\phi_u(x)] \right),$$

where  $v^{\theta\top}\phi_v(\cdot)$  and  $u^{\theta\top}\phi_u(\cdot)$  are estimates of  $V^\theta(\cdot)$  and  $U^\theta(\cdot)$  upon convergence of the TD recursion, and

$$\begin{aligned} \bar{V}^\theta(x) &= \sum_a \mu(a|x) [r(x, a) - \rho(\theta) + \sum_{x'} P(x'|x, a) v^{\theta\top}\phi_v(x')], \\ \bar{U}^\theta(x) &= \sum_a \mu(a|x) [r(x, a)^2 - \eta(\theta) + \sum_{x'} P(x'|x, a) u^{\theta\top}\phi_u(x')]. \end{aligned}$$

*Proof* The bias in estimating  $\nabla L(\theta, \lambda)$  consists of the bias in estimating  $\nabla\rho(\theta)$  and  $\nabla\eta(\theta)$ . Lemma 4 in [Bhatnagar et al. \(2009a\)](#) shows the bias in estimating  $\nabla\rho(\theta)$  as

$$\mathbb{E}[\delta_n^\theta \psi_n | \theta] = \nabla\rho(\theta) + \sum_{x \in \mathcal{X}} \mathbf{D}^\theta(x) [\nabla\bar{V}^\theta(x) - \nabla v^{\theta\top}\phi_v(x)],$$

where  $\delta_n^\theta = R(x_n, a_n) - \hat{\rho}_{n+1} + v^{\theta\top}\phi_v(x_{n+1}) - v^{\theta\top}\phi_v(x_n)$ . Similarly we can prove that the bias in estimating  $\nabla\eta(\theta)$  is

$$\mathbb{E}[\epsilon_n^\theta \psi_n | \theta] = \nabla\eta(\theta) + \sum_{x \in \mathcal{X}} \mathbf{D}^\theta(x) [\nabla\bar{U}^\theta(x) - \nabla u^{\theta\top}\phi_u(x)],$$

where  $\epsilon_n^\theta = R(x_n, a_n) - \hat{\eta}_{n+1} + u^{\theta\top}\phi_u(x_{n+1}) - u^{\theta\top}\phi_u(x_n)$ . The claim follows by putting these two results together and given the fact that  $\nabla\Lambda(\theta) = \nabla\eta(\theta) - 2\rho(\theta)\nabla\rho(\theta)$  and  $\nabla L(\theta, \lambda) = -\nabla\rho(\theta) + \lambda\nabla\Lambda(\theta)$ . Note that the following fact holds for the bias in estimating  $\nabla\rho(\theta)$  and  $\nabla\eta(\theta)$ :

$$\sum_x \mathbf{D}^\theta(x) [\bar{V}^\theta(x) - v^{\theta\top}\phi_v(x)] = 0, \quad \sum_x \mathbf{D}^\theta(x) [\bar{U}^\theta(x) - u^{\theta\top}\phi_u(x)] = 0.$$

□

**Remark 7 (Extension to Sharpe ratio optimization)** The gradient of the Sharpe ratio (SR) in the average setting is given by

$$\nabla S(\theta) = \frac{1}{\sqrt{\Lambda(\theta)}} \left( \nabla\rho(\theta) - \frac{\rho(\theta)}{2\Lambda(\theta)} \nabla\Lambda(\theta) \right),$$

and thus, the actor recursion for the SR-variant of our average reward risk-sensitive actor-critic algorithm is as follows:

$$\theta_{n+1} = \Gamma \left( \theta_n + \frac{\zeta_2(n)}{\sqrt{\hat{\eta}_{n+1} - \hat{\rho}_{n+1}^2}} \left( \delta_n \psi_n - \frac{\hat{\rho}_{n+1} (\epsilon_n \psi_n - 2\hat{\rho}_{n+1} \delta_n \psi_n)}{2(\hat{\eta}_{n+1} - \hat{\rho}_{n+1}^2)} \right) \right). \quad (49)$$

Note that the rest of the updates, including the average reward, TD errors, and critic recursions are as in the risk-sensitive actor-critic algorithm presented in Algorithm 2. Similar to the

discounted setting, since there is no Lagrange multiplier in the SR optimization, the resulting actor-critic algorithm is a two time-scale stochastic approximation algorithm.

*Remark 8* In the discounted setting, another popular variability measure is the *discounted normalized variance* Filar et al. (1989)

$$A(\mu) = \mathbb{E} \left[ \sum_{n=0}^{\infty} \gamma^n (R_n - \rho_\gamma(\mu))^2 \right], \tag{50}$$

where  $\rho_\gamma(\mu) = \sum_{x,a} d_\gamma^\mu(x|x^0)\mu(a|x)r(x, a)$  and  $d_\gamma^\mu(x|x^0)$  is the  $\gamma$ -discounted visiting distribution of state  $x$  under policy  $\mu$ , defined in Sect. 2. The variability measure (50) has close resemblance to the average reward variability measure (39), and thus, any (discounted) risk measure based on (50) can be optimized similar to the corresponding average reward risk measure (39).

*Remark 9 (Simultaneous perturbation analogues)* In the average reward setting, a simultaneous perturbation algorithm would estimate the average reward  $\rho$  and the square reward  $\eta$  on the faster timescale and use these to estimate the gradient of the performance objective. However, a drawback with this approach, compared to the algorithm proposed above is the necessity for having two simulated trajectories (instead of one) for each policy update.

In the following section, we establish the convergence of our average reward actor-critic algorithm to a (local) saddle point of the risk-sensitive objective function  $L(\theta, \lambda)$ .

## 7 Convergence analysis of the discounted reward risk-sensitive actor-critic algorithms

Our proposed actor-critic algorithms use multi-timescale stochastic approximation and we use the ordinary differential equation (ODE) approach (see Chapter 6 of Borkar (2008)) to analyze their convergence. We first provide the analysis for the SPSA based first-order algorithm RS-SPSA-G in Sect. 7.1 and later provide the necessary modifications to the proof of SF based first-order algorithm and SPSA/SF based second-order algorithms.

### 7.1 Convergence of the first-order algorithm: RS-SPSA-G

Recall that RS-SPSA-G is a two-loop scheme where the inner loop is a TD critic that evaluates the value/square value functions for both unperturbed as well as perturbed policy parameter. On the other hand, the outer loop is a two-timescale stochastic approximation algorithm, where the faster timescale updates policy parameter  $\theta$  in the descent direction using SPSA estimates of the gradient of the Lagrangian and the slower timescale performs dual ascent for the Lagrange multiplier  $\lambda$  using sample constraint values. The faster timescale  $\theta$ -recursion sees the  $\lambda$ -updates on the slower timescales as quasi-static, while the slower timescale  $\lambda$ -recursion sees the  $\theta$ -updates as equilibrated.

The proof of convergence of the RS-SPSA-G algorithm to a (local) saddle point of the risk-sensitive objective function  $\widehat{L}(\theta, \lambda) \triangleq -\widehat{V}^\theta(x^0) + \lambda(\widehat{\Lambda}^\theta(x^0) - \alpha) = -\widehat{V}^\theta(x^0) + \lambda(\widehat{U}^\theta(x^0) - \widehat{V}^\theta(x^0)^2 - \alpha)$  contains the following three main steps:

**Step 1: Critic’s convergence** We establish that, for any given values of  $\theta$  and  $\lambda$  that are updated on slower timescales, the TD critic converges to a fixed point of the projected Bellman operator for value and square value functions.

**Step 2: Convergence of  $\theta$ -recursion** We utilize the fact that owing to projection, the  $\theta$  parameter is stable. Using a Lyapunov argument, we show that the  $\theta$ -recursion tracks the ODE (55) in the asymptotic limit, for any given value of  $\lambda$  on the slowest timescale.

**Step 3: Convergence of  $\lambda$ -recursion** This step is similar to earlier analysis for constrained MDPs. In particular, we show that  $\lambda$ -recursion in (19) converges and the overall convergence of  $(\theta_n, \lambda_n)$  is to a local saddle point  $(\theta^{\lambda^*}, \lambda^*)$  of  $\widehat{L}(\theta, \lambda)$ , with  $\theta^{\lambda^*}$  satisfying the variance constraint in (3).

**Step 1: (Critic’s convergence)** Since the critic’s update is in the inner loop, we can assume in this analysis that  $\theta$  and  $\lambda$  are time-invariant quantities. The following theorem shows that the TD critic estimates for the value and square value function converge to the fixed point given by (12), for any given policy  $\theta$ .

**Theorem 2** *Under (A1)–(A4), for any given policy parameter  $\theta$  and Lagrange multiplier  $\lambda$ , the critic parameters  $\{v_m\}$  and  $\{u_m\}$  governed by the recursions of (13) converge almost surely, i.e.,*

$$\text{As } m \rightarrow \infty, v_m \rightarrow \bar{v} \text{ and } u_m \rightarrow \bar{u} \text{ a.s.}$$

*In the above  $\bar{v}$  and  $\bar{u}$  are the solutions to the TD fixed point equations for policy  $\theta$  [see (12) in Sect. 4.2].*

*Remark 10* It is easy to conclude from the above theorem that the TD critic parameters for the perturbed policy parameter also converge almost surely, i.e.,  $v_m^+ \rightarrow \bar{v}^+$  and  $u_m^+ \rightarrow \bar{u}^+$  a.s., where  $\bar{v}^+$  and  $\bar{u}^+$  are the unique solutions to TD fixed point relations for perturbed policy  $\theta_n + \beta_n \Delta_n$ , where  $\theta_n, \beta_n$  and  $\Delta_n$  correspond to the policy parameter, perturbation constant and perturbation random variable. The latter quantities are updated in the outer loop—see Algorithm 1.

We first provide a proof of Lemma 2 (see Sect. 4.2), which claimed that the operator  $\Pi T$  for the value/square value functions is a contraction mapping. The result in Lemma 2 is essential in establishing the convergence result in Theorem 2.

*Proof (Lemma 2)* We employ the technique from Tamar et al. (2013a) to prove this result. First, it is well-known that  $\Pi_v T_v^\theta$  is a contraction mapping [cf. Lemma 6 in Tsitsiklis and Roy (1997)]. This can be inferred as follows: For any  $y, \bar{y} \in \mathbb{R}^{2|\mathcal{X}|}$ ,

$$\|T_v^\theta y - T_v^\theta \bar{y}\|_{\mathcal{D}^\theta} = \gamma \|y_v - \bar{y}_v\|_{\mathcal{D}^\theta}.$$

We have used the fact that  $\|P^\theta v\|_{\mathcal{D}^\theta} \leq \|v\|_{\mathcal{D}^\theta}$  for any  $v \in \mathbb{R}^{|\mathcal{X}|}$  [For a proof, see Lemma 1 in Tsitsiklis and Roy (1997)]. The claim that  $\Pi_v T_v^\theta$  is a contraction mapping now follows from the fact that the projection operator  $\Pi_v$  is non-expansive under  $\|\cdot\|_{\mathcal{D}^\theta}$  norm.

Now, for any  $y, \bar{y} \in \mathbb{R}^{2|\mathcal{X}|}$ , we have

$$\begin{aligned} & \|\Pi_u T_u^\theta y - \Pi_u T_u^\theta \bar{y}\|_{\mathcal{D}^\theta} \\ &= \|2\gamma \Pi_u R^\theta P^\theta y_v - 2\gamma \Pi_u R^\theta P^\theta \bar{y}_v + \gamma^2 \Pi_u P^\theta y_u - \gamma^2 \Pi_u P^\theta \bar{y}_u\|_{\mathcal{D}^\theta} \\ &\leq 2\gamma \|\Pi_u R^\theta P^\theta y_v - \Pi_u R^\theta P^\theta \bar{y}_v\|_{\mathcal{D}^\theta} + \gamma^2 \|y_u - \bar{y}_u\|_{\mathcal{D}^\theta} \\ &\leq \gamma C_1 \|y_v - \bar{y}_v\|_{\mathcal{D}^\theta} + \gamma^2 \|y_u - \bar{y}_u\|_{\mathcal{D}^\theta}, \end{aligned} \tag{51}$$

for some  $C_1 < \infty$ . The first inequality above follows from the aforementioned facts that  $P^\theta$  and  $\Pi_u$  are non-expansive. The second inequality follows by using equivalence of norms [cf. the justification for Eq. (7) in the proof of Lemma 7 in Tamar et al. (2013b)].

Setting  $v = \frac{\gamma C_1}{\epsilon + \gamma C_1}$ , where  $\epsilon$  is such that  $\gamma + \epsilon < 1$  and plugging in (51), we obtain

$$\begin{aligned} & \| \Pi T^\theta y - \Pi T^\theta \bar{y} \|_v \\ &= v \| T_v^\theta y - T_v^\theta \bar{y} \|_{D^\theta} + (1 - v) \| \Pi_u T_u^\theta y - \Pi_u T_u^\theta \bar{y} \|_{D^\theta} \\ &\leq v \gamma \| y_v - \bar{y}_v \|_{D^\theta} + (1 - v) \gamma C_1 \| y_v - \bar{y}_v \|_{D^\theta} + (1 - v) \gamma^2 \| y_u - \bar{y}_u \|_{D^\theta} \\ &\leq v(\gamma + \epsilon) \| y_v - \bar{y}_v \|_{D^\theta} + (1 - v) \gamma \| y_u - \bar{y}_u \|_{D^\theta} \\ &\leq (\gamma + \epsilon) \| y - \bar{y} \|_v. \end{aligned}$$

The claim follows by setting  $\bar{\gamma} = \gamma + \epsilon$ . □

*Proof (Theorem 2)* The  $v$ -recursion in (13) is performing TD with function approximation for the value function, while the  $u$ -recursion is doing the same for the square value function. The convergence of  $v$ -recursion to the fixed point in (12) can be inferred from [Tsitsiklis and Roy \(1997\)](#).

Using an approach similar to [Tamar et al. \(2013a\)](#), we club both  $v$  and  $u$  recursions and establish convergence using a stability argument in the following: Let  $w_m = (v_m, u_m)^\top$ . Then, (13) can be seen to be equivalent to

$$\begin{aligned} w_{m+1} &= w_m + \zeta_3(m)(Mw_m + \xi + \Delta M_{m+1}), \text{ where} \\ M &= \begin{pmatrix} \Phi_v^\top D^\theta (\gamma P^\theta - I) \Phi_v & 0 \\ 2\gamma \Phi_u^\top D^\theta R^\theta P^\theta \Phi_v & \Phi_u^\top D^\theta (\gamma^2 P^\theta - I) \Phi_u \end{pmatrix} \text{ and} \\ \xi &= \begin{pmatrix} \Phi_v^\top D^\theta r^\theta \\ \Phi_u^\top D^\theta R^\theta r^\theta \end{pmatrix}. \end{aligned} \tag{52}$$

Further,  $\Delta M_{m+1}$  is a martingale difference, i.e.,  $\mathbb{E}[\Delta M_{m+1} \mid \mathcal{F}_m] = 0$ , where  $\mathcal{F}_m$  is the sigma field generated by  $w_l, \Delta M_l, l \leq m$ .

Let  $h(w) = Mw + \xi$ . Then, the ODE associated with (52) is

$$\dot{w}_t = h(w_t). \tag{53}$$

The above ODE has a unique globally asymptotically stable equilibrium, since  $M$  is a negative definite. To see the latter fact, observe that  $M$  is block triangular and hence its eigenvalues are that of  $\Phi_v^\top D^\theta (\gamma P^\theta - I) \Phi_v$  and  $\Phi_u^\top D^\theta (\gamma^2 P^\theta - I) \Phi_u$ . It can be inferred from Theorem 2 of [Tsitsiklis and Roy \(1997\)](#) that the aforementioned matrices are negative definite. For the sake of completeness, we provide a brief sketch in the following: For any  $V \in \mathbb{R}^{|\mathcal{X}|}$ , it can be shown that  $\| P^\theta V \|_{D^\theta} \leq \| V \|_{D^\theta}$  [see Lemma 1 in [Tsitsiklis and Roy \(1997\)](#) for a proof]. Now,

$$\begin{aligned} V^\top D^\theta \gamma P^\theta V &\leq \gamma \| (D^\theta)^{1/2} V \| \| (D^\theta)^{1/2} P V \| \\ &= \gamma \| V \|_{D^\theta} \| P V \|_{D^\theta} \\ &\leq \gamma \| V \|_{D^\theta}^2. \end{aligned}$$

Hence,  $V^\top D^\theta (\gamma P^\theta - I) V \leq (\gamma - 1) \| V \|_{D^\theta}^2 < 0$ . By (A3), we know that  $\Phi_v$  is full rank implying the negative definiteness of  $\Phi_v^\top D^\theta (\gamma P^\theta - I) \Phi_v$ . Using the same argument as above and replacing  $\Phi_v$  with  $\Phi_u$  and  $\gamma$  with  $\gamma^2$ , one can conclude that  $\Phi_u^\top D^\theta (\gamma^2 P^\theta - I) \Phi_u$ .

The final claim now follows by applying Theorems 2.1–2.2(i) of [Borkar and Meyn \(2000\)](#), provided we verify assumptions (A1)–(A2) there. The latter assumptions are given as follows:

**(A1)** The function  $h$  is Lipschitz. For any  $c$ , define  $h_c(w) = h(cw)/c$ . Then, there exists a continuous function  $h_\infty$  such that  $h_c \rightarrow h_\infty$  as  $c \rightarrow \infty$  uniformly on compacts. Furthermore, origin is an asymptotically stable equilibrium for the ODE

$$\dot{w}_t = h_\infty(w_t). \tag{54}$$

**(A2)** The martingale difference  $\{\Delta M_m, m \geq 1\}$  is square-integrable with

$$\mathbb{E}[\|\Delta M_{m+1}\|^2 \mid \mathcal{F}_m] \leq C_0(1 + \|w_m\|^2), m \geq 0,$$

where  $C_0 < \infty$ .

It is straightforward to verify (A1), as  $h_c(w) = Mw + \xi/c$  converges to  $h_\infty(w) = Mw$  as  $c \rightarrow \infty$ . Given that  $M$  is negative definite, it is easy to see that origin is a asymptotically stable equilibrium for the ODE (54). (A2) can also be verified by using the same arguments that were used to show that the martingale difference associated with the regular TD algorithm with function approximation satisfies a bound on the second moment (cf. Tsitsiklis and Roy 1997). □

**Step 2: (Analysis of  $\theta$ -recursion)** Due to timescale separation, the value of  $\lambda$  (updated on a slower timescale) is assumed to be constant for the analysis of the  $\theta$ -update. To see this in rigorous terms, first rewrite the  $\lambda$ -recursion as

$$\lambda_{n+1} = \Gamma_\lambda \left[ \lambda_n + \zeta_2(n) \hat{H}(n) \right].$$

where  $\hat{H}(n) = \frac{\zeta_1(n)}{\zeta_2(n)} \left( u_n^\top \phi_u(x^0) - (v_n^\top \phi_v(x^0))^2 - \alpha \right)$ . Since the critic recursions converge, it is easy to see that  $\sup_n \hat{H}(n)$  is finite. Combining with the observation that  $\frac{\zeta_1(n)}{\zeta_2(n)} = o(1)$  due to the assumption (A3) on step-sizes, we see that the  $\lambda$ -recursion above tracks the ODE  $\dot{\lambda} = 0$ .

In the following, we show that the update of  $\theta$  is equivalent to gradient descent for the function  $\widehat{L}(\theta, \lambda)$  and converges to a limiting set that depends on  $\lambda$ .

Consider the following ODE

$$\dot{\theta}_t = \check{\Gamma}(\nabla \widehat{L}(\theta_t, \lambda)), \tag{55}$$

with the limiting set  $\mathcal{Z}_\lambda = \{\theta \in C : \check{\Gamma}(\nabla \widehat{L}(\theta, \lambda)) = 0\}$ . In the above,  $\check{\Gamma}(\cdot)$  is a projection operator that ensures the evolution of  $\theta$  via the ODE (55) stays within the set  $\Theta := \prod_{i=1}^{\kappa_1} [\theta_{\min}^{(i)}, \theta_{\max}^{(i)}]$  and is defined as follows: For any bounded continuous function  $f(\cdot)$ ,

$$\check{\Gamma}(f(\theta)) = \lim_{\tau \rightarrow 0} \frac{\Gamma(\theta + \tau f(\theta)) - \theta}{\tau}. \tag{56}$$

Notice that the limit above may not exist and in that case, as pointed out on pp. 191 of Kushner and Clark (1978), one can define  $\check{\Gamma}(f(\theta))$  to be the set of all possible limit points. From the definition above, it can be inferred that for  $\theta$  in the interior of  $\Theta$ ,  $\check{\Gamma}(f(\theta)) = f(\theta)$ , while for  $\theta$  on the boundary of  $\Theta$ ,  $\check{\Gamma}(f(\theta))$  is the projection of  $f(\theta)$  onto the tangent space of the boundary of  $\Theta$  at  $\theta$ .

The main result regarding the convergence of the policy parameter  $\theta$  for both the RS-SPSA-G and RS-SF-G algorithms is as follows:

**Theorem 3** *Under (A1)–(A4), for any given Lagrange multiplier  $\lambda$ ,  $\theta_n$  updated according to (19) converges almost surely to  $\theta^* \in \mathcal{Z}_\lambda$ .*

The proof of the above theorem requires the following lemma which shows that the conditions  $m_n, \beta_n$  in (A4) ensure that the TD-critic does not introduce any bias from a finite sample run length of  $m_n$ .

**Lemma 6** *Let*

$$\begin{aligned} \mathcal{T}_n^{(i)} &\triangleq \left( (1 + 2\lambda v_n^\top \phi_v(x^0)) \frac{(v_n^+ - v_n)^\top \phi_v(x^0)}{\beta_n \Delta_n^{(i)}} - \lambda \frac{(u_n^+ - u_n)^\top \phi_u(x^0)}{\beta_n \Delta_n^{(i)}} \right), \\ \widehat{L}(\theta, \lambda) &\triangleq -\widehat{V}^\theta(x^0) + \lambda(\widehat{U}^\theta(x^0) - \widehat{V}^\theta(x^0)^2 - \alpha), \end{aligned}$$

where  $\widehat{V}(\theta) = \phi_{\bar{v}}(x^0)^\top \bar{v}$  and  $\widehat{U}(\theta) = \phi_{\bar{u}}(x^0)^\top \bar{u}$  denote the approximate value and square value functions for policy  $\theta$ .<sup>7</sup>

Then, we have that

$$\left| \mathbb{E} \left( \mathcal{T}_n^{(i)} \mid \theta_n \right) - \nabla \widehat{L}(\theta_n, \lambda) \right| = O(\beta_n^2), \text{ for } i = 1, \dots, \kappa_1.$$

*Proof* Let

$$\xi_{1,n} := \left( \mathcal{T}_n^{(i)} - \left( (1 + 2\lambda \bar{v}^\top \phi_v(x^0)) \frac{(\bar{v}^+ - \bar{v})^\top \phi_v(x^0)}{\beta_n \Delta_n^{(i)}} - \lambda \frac{(\bar{u}^+ - \bar{u})^\top \phi_u(x^0)}{\beta_n \Delta_n^{(i)}} \right) \right).$$

From Theorem 1, we know that the critic parameters  $v_n, u_n$  converge to their limits  $\bar{v}, \bar{u}$  at the rate  $O(m^{-1/2})$  and hence, after  $m_n$  steps of the TD-critic,  $\xi_{1,n} = O(\frac{1}{\sqrt{m_n \beta_n}})$ . Now, from (A4), we have that  $\frac{1}{\sqrt{m_n \beta_n}} \rightarrow 0$  and hence  $\xi_{1,n}$  vanishes asymptotically. Hence, we have

$$\mathcal{T}_n^{(i)} \rightarrow \left( (1 + 2\lambda \bar{v}^\top \phi_v(x^0)) \frac{(\bar{v}^+ - \bar{v})^\top \phi_v(x^0)}{\beta_n \Delta_n^{(i)}} - \lambda \frac{(\bar{u}^+ - \bar{u})^\top \phi_u(x^0)}{\beta_n \Delta_n^{(i)}} \right). \tag{57}$$

We next show that the RHS above is an order  $O(\beta_n^2)$  term away from the gradient of the Lagrangian  $L(\theta_n, \lambda)$ . Using a Taylor’s expansion of  $\widehat{V}(\cdot)$  around  $\theta_n$ , we obtain:

$$\widehat{V}(\theta_n + \beta_n \Delta_n) = \widehat{V}(\theta_n) + \beta_n \Delta_n^\top \nabla \widehat{V}(\theta_n) + \frac{\beta_n^2}{2} \Delta_n^\top \nabla^2 \widehat{V}(\theta_n) \Delta_n + O(\beta_n^3).$$

Taking expectations and rearranging terms, we obtain

$$\begin{aligned} &\mathbb{E} \left[ \left( \frac{\widehat{V}(\theta_n + \beta_n \Delta_n) - \widehat{V}(\theta_n)}{\beta_n \Delta_n^{(i)}} \right) \mid \theta_n \right] \\ &= \mathbb{E} \left[ \frac{\Delta_n^\top \nabla \widehat{V}(\theta_n)}{\Delta_n^{(i)}} \mid \theta_n \right] + \mathbb{E} \left[ \frac{\Delta_n^\top \nabla_{\theta_n}^2 \widehat{V}(\theta_n) \Delta_n}{\Delta_n^{(i)}} \mid \theta_n \right] + O(\beta_n^2) \\ &= \nabla_i \widehat{V}(\theta_n) + \mathbb{E} \left[ \sum_{j \neq i} \frac{\Delta_n^{(j)}}{\Delta_n^{(i)}} \nabla_j \widehat{V}(\theta_n) \mid \theta_n \right] + O(\beta_n^2) \\ &= \nabla_i \widehat{V}(\theta_n) + O(\beta_n^2). \end{aligned} \tag{58}$$

In the above, we have used the fact that  $\Delta_n$  is i.i.d. Rademacher and independent of  $\theta_n$ .

<sup>7</sup> For notational convenience, we drop the dependence of  $\bar{v}$  and  $\bar{u}$  on the underlying policy parameter  $\theta$  and this dependence should be clear from the context.

In a similar manner, defining  $\widehat{U}(\theta_n) = \phi_{\bar{u}}(x^0)^\top \bar{u}$  and  $\widehat{U}(\theta_n + \beta_n \Delta_n) = \phi_{\bar{u}^+}(x^0)^\top \bar{u}^+$ , we can conclude that

$$\mathbb{E} \left[ \left( \frac{\widehat{U}(\theta_n + \beta_n \Delta_n) - \widehat{U}(\theta_n)}{\beta_n \Delta_n^{(i)}} \right) \middle| \theta_n \right] = \nabla_i \widehat{U}(\theta_n) + O(\beta_n^2). \tag{59}$$

The claim now follows by plugging in (58)–(59) into (57). □

In order to prove Theorem 3, we require the well-known Kushner–Clark lemma (see Kushner and Clark 1978, pp. 191–196). For the sake of completeness, we recall this result below.

*Kushner–Clark lemma* Consider the following recursion in  $\kappa_1$ -dimensions:

$$x_{n+1} = \Gamma(x_n + a(n)(h(x_n) + \xi_{1,n} + \xi_{2,n})), \tag{60}$$

where  $\Gamma$  projects the iterate  $x_n$  onto a compact and convex set, say  $C \in \mathbb{R}^N$ . The ODE associated with (60) is given by

$$\dot{x}(t) = \bar{\Gamma}(h(x(t))), \tag{61}$$

where  $\bar{\Gamma}$  is a projection operator that keeps the ODE evolution within the set  $C$  and is defined as in (56).

We make the following assumptions:

- (B1)  $h$  is a continuous  $\mathbb{R}^{\kappa_1}$ -valued function.
- (B2) The sequence  $\xi_{1,n}, n \geq 0$  is a bounded random sequence with  $\xi_{1,n} \rightarrow 0$  almost surely as  $n \rightarrow \infty$ .
- (B3) The step-sizes  $a(n), n \geq 0$  satisfy  $a(n) \rightarrow 0$  as  $n \rightarrow \infty$  and  $\sum_n a(n) = \infty$ .
- (B4)  $\{\xi_{2,n}, n \geq 0\}$  is a sequence such that for any  $\epsilon > 0$ ,

$$\lim_{n \rightarrow \infty} P \left( \sup_{m \geq n} \left\| \sum_{i=n}^m a_i \xi_{1,i} \right\| \geq \epsilon \right) = 0.$$

- (B5) The ODE (61) has a compact subset  $K$  of  $\mathcal{R}^{\kappa_1}$  as its set of asymptotically stable equilibrium points.

The main result (see Kushner and Clark 1978, pp. 191–196) is as follows:

**Theorem 4** Assume (B1)–(B5). Then,  $x_n$  converges almost surely to the set  $K$ .

*Proof (Theorem 3)* We first rewrite the recursion (19) as follows:

$$\theta_{n+1}^{(i)} = \Gamma_i \left( \theta_n^{(i)} + \zeta_2(n) \left( \nabla \widehat{L}(\theta_n, \lambda) + \xi_{1,n} + \xi_{2,n} \right) \right), \tag{62}$$

where

$$\begin{aligned} \xi_{1,n} &= \mathbb{E} \left( \mathcal{T}_n^{(i)} \mid \theta_n \right) - \nabla \widehat{L}(\theta_n, \lambda), \\ \xi_{2,n} &= \mathcal{T}_n^{(i)} - \mathbb{E} \left( \mathcal{T}_n^{(i)} \mid \theta_n \right), \end{aligned}$$

with  $\mathcal{T}_n^{(i)}$  defined as in Lemma 6.

We now verify (B1)–(B5) for the above recursion:

- From (A1) together with the facts that the state space is finite and the projection  $\Gamma$  is onto a compact set, we have from Theorem 2 of Schweitzer (1968) that the stationary distributions  $D_\gamma^\theta(x|x^0)$  and  $\tilde{d}_\gamma^\theta(x|x^0)$  are continuously differentiable. This in turn implies continuity of  $\nabla\widehat{V}(\theta_n)$  and  $\nabla\widehat{U}(\theta_n)$ . Thus, (B1) follows for  $\nabla\widehat{L}(\theta_n, \lambda)$ .
- In light of Lemma 6 and (A4), we have that  $\xi_{1,n} \rightarrow 0$  as  $n \rightarrow \infty$ .
- (A4) implies (B3).
- A simple calculation shows that  $\mathbb{E}(\xi_{2,n})^2 \leq \mathbb{E}(\mathcal{T}_n^{(i)})^2 \leq C_3/\beta_n^2$  for some  $C_3 < \infty$ . Applying Doob’s inequality, we obtain

$$P\left(\sup_{l \geq k} \left\| \sum_{n=k}^l \zeta_2(n) \xi_{2,n} \right\| \geq \epsilon\right) \leq \frac{1}{\epsilon^2} \sum_{n=k}^\infty \zeta_2(n)^2 \mathbb{E} \|\xi_{2,n}\|^2. \tag{63}$$

$$\leq \frac{C_3}{\epsilon^2} \sum_{n=k}^\infty \frac{\zeta_2(n)^2}{\beta_n^2} = 0. \tag{64}$$

Thus, (B4) is satisfied.

- $\mathcal{Z}_\lambda$  is an asymptotically stable attractor for the ODE (55), with  $\widehat{L}(\theta, \lambda)$  itself serving as a strict Lyapunov function. This can be inferred as follows:

$$\frac{d\widehat{L}(\theta, \lambda)}{dt} = \nabla\widehat{L}(\theta, \lambda)\dot{\theta} = \nabla\widehat{L}(\theta, \lambda)\check{\Gamma}(-\nabla\widehat{L}(\theta, \lambda)) < 0.$$

The claim now follows from Kushner–Clark lemma. □

**Step 3: (Analysis of  $\lambda$ -recursion and convergence to a local saddle point)** We first show that the  $\lambda$ -recursion converges and then prove that the whole algorithm converges to a local saddle point of  $\widehat{L}(\theta, \lambda)$ .

We define the following ODE governing the evolution of  $\lambda$ :

$$\dot{\lambda}_t = \check{\Gamma}_\lambda[\widehat{\Lambda}^{\theta^{\lambda_t}}(x^0) - \alpha] = \check{\Gamma}_\lambda[\widehat{U}^{\theta^{\lambda_t}}(x^0) - \widehat{V}^{\theta^{\lambda_t}}(x^0)^2 - \alpha], \tag{65}$$

where  $\theta^{\lambda_t}$  is the limiting point of the  $\theta$ -recursion corresponding to  $\lambda_t$ . Further,  $\check{\Gamma}_\lambda$  is an operator similar to the operator  $\check{\Gamma}$  defined in (56) and is defined as follows: For any bounded continuous function  $f(\cdot)$ ,

$$\check{\Gamma}_\lambda(f(\lambda)) = \lim_{\tau \rightarrow 0} \frac{\Gamma_\lambda(\lambda + \tau f(\lambda)) - \lambda}{\tau}. \tag{66}$$

**Theorem 5**  $\lambda_n \rightarrow \mathcal{F}$  almost surely as  $n \rightarrow \infty$ , where  $\mathcal{F} \triangleq \{\lambda \mid \lambda \in [0, \lambda_{\max}], \check{\Gamma}_\lambda[\widehat{\Lambda}^{\theta^\lambda}(x^0) - \alpha] = 0, \theta^\lambda \in \mathcal{Z}_\lambda\}$ .

*Proof* The proof follows using standard stochastic approximation arguments. The first step is to rewrite the  $\lambda$ -recursion as follows:

$$\lambda_{n+1} = \Gamma_\lambda \left[ \lambda_n + \zeta_1(n) \left( \bar{u}^\top \phi_u(x^0) - (\bar{v}^\top \phi_v(x^0))^2 - \alpha + \xi_{2,n} \right) \right],$$

where  $\xi_{2,n} := \left( u_n^\top \phi_u(x^0) - (v_n^\top \phi_v(x^0))^2 \right) - \left( \bar{u}^\top \phi_u(x^0) - (\bar{v}^\top \phi_v(x^0))^2 \right)$ . Note that the converged critic parameters  $\bar{v}$  and  $\bar{u}$  are for the policy  $\theta^{\lambda_n}$ . The latter is a limiting point of the  $\theta$ -recursion, with the Lagrange multiplier  $\lambda_n$ . Owing to convergence of  $\theta$ -recursion and also TD-critic in the inner loop, we can conclude that  $\xi_{2,n} = o(1)$ . Thus,  $\xi_{2,n}$  adds an asymptotically vanishing bias term to the  $\lambda$ -recursion above. The claim follows by applying the



standard result in Theorem 2 of [Borkar \(2008\)](#) for convergence of stochastic approximation schemes.  $\square$

Recall that  $\widehat{L}(\theta, \lambda) \triangleq -\widehat{V}^\theta(x^0) + \lambda(\widehat{A}^\theta(x^0) - \alpha)$  and hence  $\nabla_\lambda \widehat{L}(\theta, \lambda) = \widehat{A}^\theta(x^0) - \alpha$ . Thus,

$$\check{I}_\lambda[\widehat{A}^{\theta^\lambda}(x^0) - \alpha] = 0,$$

is the same as

$$\check{I}_\lambda \nabla_\lambda \widehat{L}(\theta^\lambda, \lambda) = 0.$$

As in [Borkar \(2005\)](#), we invoke the envelope theorem of mathematical economics ([Mas-Colell et al. 1995](#)) to conclude that the ODE (65) is equivalent to the following

$$\dot{\lambda}_t = \check{I}_\lambda[\nabla_\lambda \widehat{L}(\theta^{\lambda_t}, \lambda_t)]. \tag{67}$$

Note that the above has to be interpreted in the *Cartheodory* sense, i.e., as the following integral equation

$$\lambda_t = \lambda_0 + \int_0^t \check{I}_\lambda[\nabla_\lambda \widehat{L}(\theta^{\lambda_s}, \lambda_s)] ds.$$

As noted in Lemma 4.3 of [Borkar \(2005\)](#), using the generalized envelope theorem from [Milgrom and Segal \(2002\)](#) it can be shown that the RHS of (67) coincides with that of (65) at differentiable points, while the ODE spends zero time at non-differentiable points (except at the points of maxima).

We next claim that the limit  $\theta^{\lambda^*}$  corresponding to  $\lambda^*$  satisfies the variance constraint in (3), i.e.,

**Proposition 1** *For any  $\lambda^*$  in  $\widehat{\mathcal{F}} \triangleq \{\lambda \mid \lambda \in [0, \lambda_{\max}), \check{I}_\lambda[\widehat{A}^{\theta^\lambda}(x^0) - \alpha] = 0, \theta^\lambda \in \mathcal{Z}_\lambda\}$ , the corresponding limiting point  $\theta^{\lambda^*}$  satisfies the variance constraint  $\widehat{A}^{\theta^{\lambda^*}}(x^0) \leq \alpha$ .*

*Proof* Follows in a similar manner as Proposition 10.6 in [Bhatnagar et al. \(2013\)](#).  $\square$

From Theorems 3, 4, 5 and Proposition 1, it is evident that the actor recursion (19) converges to a tuple  $(\theta^{\lambda^*}, \lambda^*)$  that is a local minimum w.r.t.  $\theta$  and a local maximum w.r.t.  $\lambda$  of  $\widehat{L}(\theta, \lambda)$ . In other words, overall convergence is to a (local) saddle point of  $\widehat{L}(\theta, \lambda)$ . Further, the limit is also feasible for the constrained problem in (3) as  $\theta^{\lambda^*}$  satisfies the variance constraint there.

### 7.2 Convergence of the first-order algorithm: RS-SF-G

Note that since RS-SPSA-G and RS-SF-G use different methods to estimate the gradient, their proofs only differ in the second step, i.e., the convergence of the policy parameter  $\theta$ .

#### 7.2.1 Proof of Theorem 3 for SF

*Proof* As in the case of the SPSA algorithm, we rewrite the  $\theta$ -update in (20) using the converged TD-parameters and constant  $\lambda$  as

$$\theta_{n+1}^{(i)} = \Gamma_i \left( \theta_n^{(i)} - \zeta_2(n) \left( \frac{-\Delta_n^{(i)} (1 + 2\lambda \bar{v}^\top \phi_v(x^0))}{\beta} (\bar{v}^+ - \bar{v})^\top \phi_v(x^0) + \frac{\lambda \Delta_n^{(i)}}{\beta} (\bar{u}^+ - \bar{u})^\top \phi_u(x^0) + \xi_{1,n} \right) \right),$$

where  $\xi_{1,n} \rightarrow 0$  by using arguments analogous to those in the proof of Lemma 6. Next, we establish that  $\mathbb{E} \left[ \frac{\Delta_n^{(i)}}{\beta_n} (\bar{v}^+ - \bar{v})^\top \phi_v(x^0) \mid \theta, \lambda \right]$  is an asymptotically correct estimate of the gradient of  $\widehat{V}(\theta)$  in the following:

$$\mathbb{E} \left[ \frac{\Delta_n^{(i)}}{\beta_n} (\bar{v}^+ - \bar{v})^\top \phi_v(x^0) \mid \theta_n, \lambda \right] \rightarrow \nabla_i \bar{v}^\top \phi_v(x^0) \text{ a.s. as } n \rightarrow \infty.$$

The above follows in a similar manner as Proposition 10.2 of Bhatnagar et al. (2013). On similar lines, one can see that

$$\mathbb{E} \left[ \frac{\Delta_n^{(i)}}{\beta_n} (\bar{u}^+ - \bar{u})^\top \phi_u(x^0) \mid \theta_n, \lambda \right] \rightarrow \nabla_i \bar{u}^\top \phi_u(x^0) \text{ a.s. as } n \rightarrow \infty.$$

Thus, (20) can be seen to be a discretization of the ODE (55) and the rest of the analysis follows in a similar manner as in the SPSA proof. □

### 7.2.2 Convergence of the second-order algorithms: RS-SPSA-N and RS-SF-N

Convergence analysis of the second-order algorithms involves the same steps as that of the first-order algorithms. In particular, the first step involving the TD-critic and the third step involving the analysis of  $\lambda$ -recursion follow along similar lines as earlier, whereas  $\theta$ -recursion analysis in the second step differs significantly.

**Step 2: (Analysis of  $\theta$ -recursion for RS-SPSA-N and RS-SF-N)** Since the policy parameter is updated in the descent direction with a Newton decrement, the limiting ODE of the  $\theta$ -recursion for the second order algorithms is given by

$$\dot{\theta}_t = \check{\Gamma} \left( \Upsilon (\nabla^2 L(\theta_t, \lambda))^{-1} \nabla L(\theta_t, \lambda) \right), \tag{68}$$

where  $\check{\Gamma}$  is as before [see (56)]. Let

$$\mathcal{Z}_\lambda = \left\{ \theta \in \Theta : -\nabla L(\theta, \lambda)^T \Upsilon (\nabla_\theta^2 L(\theta, \lambda))^{-1} \nabla L(\theta, \lambda) = 0 \right\}.$$

denote the set of asymptotically stable equilibrium points of the ODE (68) and  $\mathcal{Z}_\lambda^\varepsilon$  its  $\varepsilon$ -neighborhood. Then, we have the following analogue of Theorem 3 for the RS-SPSA-N and RS-SF-N algorithms:

**Theorem 6** *Under (A1)–(A5), for any given Lagrange multiplier  $\lambda$  and  $\varepsilon > 0$ , there exists  $\beta_0 > 0$  such that for all  $\beta \in (0, \beta_0)$ ,  $\theta_n \rightarrow \theta^* \in \mathcal{Z}_\lambda^\varepsilon$  almost surely.*

### 7.2.3 Proof of Theorem 6 for RS-SPSA-N

Before we prove Theorem 6, we establish that the Hessian estimate  $H_n$  in (30) converges almost surely to the true Hessian  $\nabla_\theta^2 L(\theta_n, \lambda)$  in the following lemma.

**Lemma 7** For all  $i, j \in \{1, \dots, \kappa_1\}$ , we have the following claims with probability one:

- (i)  $\left\| \frac{L(\theta_n + \beta_n \Delta_n + \beta_n \widehat{\Delta}_n, \lambda) - L(\theta_n, \lambda)}{\beta_n^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}} - \nabla_{\theta_n^{(i,j)}}^2 L(\theta_n, \lambda) \right\| \rightarrow 0,$
- (ii)  $\left\| \frac{L(\theta_n + \beta_n \Delta_n + \beta_n \widehat{\Delta}_n, \lambda) - L(\theta_n, \lambda)}{\beta_n \widehat{\Delta}_n^{(i)}} - \nabla_{\theta_n^{(i)}} L(\theta_n, \lambda) \right\| \rightarrow 0,$
- (iii)  $\left\| H^{(i,j)} - \nabla_{\theta_n^{(i,j)}}^2 L(\theta_n, \lambda) \right\| \rightarrow 0,$
- (iv)  $\left\| M - \Upsilon(\nabla_{\theta_n}^2 L(\theta_n, \lambda))^{-1} \right\| \rightarrow 0.$

*Proof* The proofs of the above claims follow from Propositions 10.10, 10.11 and Lemmas 7.10 and 7.11 of [Bhatnagar et al. \(2013\)](#), respectively.  $\square$

*Proof (Theorem 6 for RS-SPSA-N)* As in the case of the first order methods, due to timescale separation, we can treat  $\lambda_n \equiv \lambda$ , a constant and use the converged TD-parameters to arrive at the following equivalent update rules for the Hessian recursion (30) and  $\theta$ -recursion (31):

$$\begin{aligned}
 H_{n+1}^{(i,j)} &= H_n^{(i,j)} + \zeta_2'(n) \left[ \frac{(1 + \lambda_n(\bar{v}_n + \bar{v}_n^+)^\top \phi_v(x^0))(\bar{v}_n - \bar{v}_n^+)^\top \phi_v(x^0)}{\beta_n^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}} \right. \\
 &\quad \left. + \frac{\lambda(\bar{u}_n^+ - \bar{u}_n)^\top \phi_u(x^0)}{\beta_n^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}} - H_n^{(i,j)} \right], \\
 \theta_{n+1}^{(i)} &= \Gamma_i \left[ \theta_n^{(i)} + \zeta_2(n) \sum_{j=1}^{\kappa_1} M_n^{(i,j)} \left( \frac{(1 + 2\lambda \bar{v}_n^\top \phi_v(x^0))(\bar{v}_n^+ - \bar{v}_n)^\top \phi_v(x^0)}{\beta_n \Delta_n^{(j)}} \right. \right. \\
 &\quad \left. \left. - \frac{\lambda(\bar{u}_n^+ - \bar{u}_n)^\top \phi_u(x^0)}{\beta_n \Delta_n^{(j)}} \right) \right].
 \end{aligned}$$

By a completely parallel argument to the proof of Lemma 6 in conjunction with Lemma 7, the  $\theta$ -recursion above is equivalent to the following:

$$\theta_{n+1}^{(i)} = \bar{\Gamma}_i \left( \theta_n^{(i)} + \zeta_2(n) (\nabla^2 L(\theta_n, \lambda))^{-1} \nabla L(\theta_n, \lambda) \right). \tag{69}$$

The above can be seen as a discretization of the ODE (68), with  $\mathcal{Z}_\lambda$  serving as its asymptotically stable attractor. The rest of the claim follows in a similar manner as Theorem 3.  $\square$

### 7.2.4 Proof of Theorem 6 for RS-SF-N

*Proof* We first establish the following result for the gradient and Hessian estimators employed in RS-SF-N:  $\square$

**Lemma 8** We have the following claims with probability one:

- (i)  $\left\| E \left[ \frac{1}{\beta_n^2} \bar{H}(\Delta_n)(L(\theta_n + \beta_n \Delta_n, \lambda) - L(\theta_n, \lambda)) \mid \theta_n, \lambda \right] - \nabla_{\theta}^2 L(\theta_n, \lambda) \right\| \rightarrow 0.$
- (ii)  $\left\| E \left[ \frac{1}{\beta_n} \Delta_n(L(\theta_n + \beta_n \Delta_n, \lambda) - L(\theta_n, \lambda)) \mid \theta_n, \lambda \right] - \nabla L(\theta_n, \lambda) \right\| \rightarrow 0.$

*Proof* The proofs of the above claims follow from Propositions 10.1 and 10.2 of [Bhatnagar et al. \(2013\)](#), respectively.  $\square$

The rest of the analysis is identical to that of RS-SPSA-N.

**Remark 11 (On convergence rate)** In the above, we established asymptotic limits for all our algorithms using the ODE approach. To the best of our knowledge, there are no convergence rate results available for multi-timescale stochastic approximation schemes, and hence, for actor-critic algorithms. This is true even for the actor-critic algorithms that do not incorporate any risk criterion. In [Konda and Tsitsiklis \(2004\)](#), the authors provide asymptotic convergence rate results for *linear* two-timescale recursions. It would be an interesting direction for future research to obtain concentration bounds for general (non-linear) two-timescale schemes.

While a rigorous analysis on convergence rate of our proposed schemes is difficult, one could make a few concessions and use the following argument to see that the SPSA-based algorithms converge quickly: In order to analyse the rate of convergence of  $\theta$ -recursion, assume (for sufficiently large  $n$ ) that the TD-critic has converged in the inner-loop. This is because, the trajectory lengths  $m_n \rightarrow \infty$  as  $n \rightarrow \infty$  and under appropriate step-size settings (or with iterate averaging) one can obtain convergence rate of the order  $O(1/\sqrt{m})$  on the root mean square error of TD (see [Theorem 1](#)). Now, if one holds  $\lambda$  fixed, then invoking asymptotic normality results for SPSA [see [Proposition 2](#) in [Spall \(1992\)](#)] it can be shown that  $n^{1/3}(\theta_n - \theta^\lambda)$  is asymptotically normal, where  $\theta^\lambda$  is a limit point in the set  $\mathcal{Z}_\lambda$ . Similar results also hold for second-order SPSA variants [cf. [Theorem 3a](#) in [Spall \(2000\)](#)]. Both the aforementioned claims are proved using a well-known result on asymptotic normality of stochastic approximation schemes due to [Fabian \(1968\)](#).

The second-order schemes such as RS-SPSA-N score over their first order counterpart RS-SPSA-G from an asymptotic normality results perspective. This is because obtaining the optimal convergence rate for RS-SPSA-G requires that the step-size  $\zeta_2(n)$  is set to  $\zeta_2(0)/n$  where  $\zeta_2(0) > 1/\lambda_{\min}(\nabla_\theta^2 L(\theta^\lambda, \lambda))$ , whereas there is no such constraint for the second-order algorithm RS-SPSA-N. Here  $\lambda_{\min}(A)$  denotes the minimum eigenvalue of the matrix  $A$ . The reader is referred to [Dippon and Renz \(1997\)](#) for a detailed discussion on convergence rate of (one timescale) SPSA-based schemes using asymptotic mean-square error.

**Remark 12 (Unstable equilibria)** The limit set  $\mathcal{Z}_\lambda$  contains both stable and unstable equilibria and the  $\theta$ -recursion can possibly end up in a unstable equilibrium point. One may avoid this situation by including additional noise in the randomized policy that drives the  $\theta$ -recursion. For instance, define a  $\eta$ -offset policy as

$$\hat{\mu}(a | x) = \frac{\mu(a | x) + \eta}{\sum_{a' \in \mathcal{A}(x)} (\mu(a' | x) + \eta)}.$$

The above policy can be used in place of the regular  $\mu(\cdot | x)$ , so that the algorithm is pulled away from an unstable equilibria. Providing theoretical guarantees for such a scheme is non-trivial and we have left it for future work.

## 8 Convergence analysis of the average reward risk-sensitive actor-critic algorithm

As in the discounted setting, we use the ODE approach ([Borkar 2008](#)) to analyze the convergence of our average reward risk-sensitive actor-critic algorithm. The proof involves three main steps:

1. The first step is the convergence of  $\rho, \eta, V,$  and  $U,$  for any fixed policy  $\theta$  and Lagrange multiplier  $\lambda.$  This corresponds to a TD(0) (with extension to  $\eta$  and  $U$ ) proof. Using arguments similar to that in Step 2 of the proof of RS-SPSA-G, one can show that the  $\theta$  and  $\lambda$  recursions track  $\hat{\theta}_t = 0$  and  $\hat{\lambda}_t = 0,$  when viewed from the TD critic timescale  $\{\zeta_3(t)\}.$  Thus, the policy  $\theta$  and Lagrange multiplier  $\lambda$  are assumed to be constant in the analysis of the critic recursion.
2. The second step is to show the convergence of  $\theta_n$  to an  $\varepsilon$ -neighborhood  $\mathcal{Z}_\lambda^\varepsilon$  of the set of asymptotically stable equilibria  $\mathcal{Z}_\lambda$  of ODE

$$\dot{\theta}_t = \check{I}(\nabla L(\theta_t, \lambda)), \tag{70}$$

where the projection operator  $\check{I}$  ensures that the evolution of  $\theta$  via the ODE (70) stays within the compact and convex set  $\Theta \subset \mathbb{R}^{k_1}$  and is defined in (56). Again here it is assumed that  $\lambda$  is fixed because  $\theta$ -recursion is on a faster time-scale than  $\lambda$ 's.

3. The final step is the convergence of  $\lambda$  and showing that the whole algorithm converges to a local saddle point of  $L(\theta, \lambda).$  where the limit is shown to satisfy the variance constraint in (40).

**Step 1: Critic’s convergence**

**Lemma 9** *For any given policy  $\mu, \{\hat{\rho}_n\}, \{\hat{\eta}_n\}, \{v_n\},$  and  $\{u_n\},$  defined in Algorithm 2 and by the critic recursion (46) converge to  $\rho(\mu), \eta(\mu), v^\mu,$  and  $u^\mu$  almost surely, where  $v^\mu$  and  $u^\mu$  are the unique solutions to*

$$\Phi_v^\top \mathbf{D}^\mu \Phi_v v^\mu = \Phi_v^\top \mathbf{D}^\mu T_v^\mu(\Phi_v v^\mu), \quad \Phi_u^\top \mathbf{D}^\mu \Phi_u u^\mu = \Phi_u^\top \mathbf{D}^\mu T_u^\mu(\Phi_u u^\mu), \tag{71}$$

respectively. In (71),  $\mathbf{D}^\mu$  denotes the diagonal matrix with entries  $d^\mu(x)$  for all  $x \in \mathcal{X},$  and  $T_v^\mu$  and  $T_u^\mu$  are the Bellman operators for the differential value and square value functions of policy  $\mu,$  defined as

$$T_v^\mu J = \mathbf{r}^\mu - \rho(\mu)\mathbf{e} + \mathbf{P}^\mu J, \quad T_u^\mu J = \mathbf{R}^\mu \mathbf{r}^\mu - \eta(\mu)\mathbf{e} + \mathbf{P}^\mu J, \tag{72}$$

where  $\mathbf{r}^\mu$  and  $\mathbf{P}^\mu$  are the reward vector and transition probability matrix of policy  $\mu, \mathbf{R}^\mu = \text{diag}(\mathbf{r}^\mu),$  and  $\mathbf{e}$  is a vector of size  $n$  (the size of the state space  $\mathcal{X}$ ) with elements all equal to one.

*Proof* The proof for the average reward  $\rho(\mu)$  and differential value function  $v^\mu$  follows in a similar manner as Lemma 5 in Bhatnagar et al. (2009a). It is based on verifying the Assumptions (A1)–(A2) of Borkar and Meyn (2000), and uses the second part of Assumption (A3) of our paper, i.e.,  $v \in \mathbb{R}^{k_2},$  for every  $v \in \mathbb{R}^{k_2}.$  The proof for  $\rho(\mu)$  and  $v^\mu$  can be easily extended to the square average reward  $\eta(\mu)$  and square differential value function  $u^\mu.$   $\square$

**Step 2: Actor’s convergence**

Let  $\mathcal{Z}_\lambda = \{\theta \in \Theta : \check{I}(-\nabla L(\theta, \lambda)) = 0\}$  denote the set of asymptotically stable equilibrium points of the ODE (70) and  $\mathcal{Z}_\lambda^\varepsilon = \{\theta \in \Theta : \|\theta - \theta_0\| < \varepsilon, \theta_0 \in \mathcal{Z}_\lambda\}$  denote the set of points in the  $\varepsilon$ -neighborhood of  $\mathcal{Z}_\lambda.$  The main result regarding the convergence of the policy parameter in (47) is as follows:

**Theorem 7** *Assume (A1)–(A4). Then, given  $\varepsilon > 0, \exists \beta > 0$  such that for  $\theta_n, n \geq 0$  obtained by the algorithm, if  $\sup_{\theta_n} \|\mathcal{B}(\theta_n, \lambda)\| < \beta,$  then  $\theta_n$  governed by (47) converges almost surely to  $\mathcal{Z}_\lambda^\varepsilon$  as  $n \rightarrow \infty.$*

*Proof* Let  $\mathcal{F}(n) = \sigma(\theta_m, m \leq n)$  denote a sequence of  $\sigma$ -fields. We have

$$\begin{aligned} \theta_{n+1} &= \Gamma\left(\theta_n - \zeta_2(n)\left(-\delta_n\psi_n + \lambda(\epsilon_n\psi_n - 2\widehat{\rho}_{n+1}\delta_n\psi_n)\right)\right) \\ &= \Gamma\left(\theta_n + \zeta_2(n)\left(1 + 2\lambda\widehat{\rho}_{n+1}\right)\delta_n\psi_n - \zeta_2(n)\lambda\epsilon_n\psi_n\right) \\ &= \Gamma\left(\theta_n - \zeta_2(n)\left[1 + 2\lambda\left(\widehat{\rho}_{n+1} - \rho(\theta_n)\right) + \rho(\theta_n)\right]\mathbb{E}\left[\delta^{\theta_n}\psi_n|\mathcal{F}(n)\right]\right. \\ &\quad - \zeta_2(n)\left[1 + 2\lambda\left(\widehat{\rho}_{n+1} - \rho(\theta_n)\right) + \rho(\theta_n)\right]\left(\delta_n\psi_n - \mathbb{E}\left[\delta_n\psi_n|\mathcal{F}(n)\right]\right) \\ &\quad - \zeta_2(n)\left[1 + 2\lambda\left(\widehat{\rho}_{n+1} - \rho(\theta_n)\right) + \rho(\theta_n)\right]\mathbb{E}\left[(\delta_n - \delta^{\theta_n})\psi_n|\mathcal{F}(n)\right] \\ &\quad + \zeta_2(n)\lambda\mathbb{E}\left[\epsilon^{\theta_n}\psi_n|\mathcal{F}(n)\right] + \zeta_2(n)\lambda\left(\epsilon_n\psi_n - \mathbb{E}\left[\epsilon_n\psi_n|\mathcal{F}(n)\right]\right) \\ &\quad \left. + \zeta_2(n)\lambda\mathbb{E}\left[(\epsilon_n - \epsilon^{\theta_n})\psi_n|\mathcal{F}(n)\right]\right). \end{aligned}$$

By setting  $\xi_n = \widehat{\rho}_{n+1} - \rho(\theta_n)$ , we may write the above equation as

$$\theta_{n+1} = \Gamma\left(\theta_n - \zeta_2(n)\left[1 + 2\lambda(\xi_n + \rho(\theta_n))\right]\mathbb{E}\left[\delta^{\theta_n}\psi_n|\mathcal{F}(n)\right]\right) \tag{73}$$

$$\begin{aligned} &- \zeta_2(n)\left[1 + 2\lambda(\xi_n + \rho(\theta_n))\right]\underbrace{\left(\delta_n\psi_n - \mathbb{E}\left[\delta_n\psi_n|\mathcal{F}(n)\right]\right)}_{*} \\ &- \zeta_2(n)\left[1 + 2\lambda(\xi_n + \rho(\theta_n))\right]\underbrace{\mathbb{E}\left[(\delta_n - \delta^{\theta_n})\psi_n|\mathcal{F}(n)\right]}_{+} \\ &+ \zeta_2(n)\lambda\mathbb{E}\left[\epsilon^{\theta_n}\psi_n|\mathcal{F}(n)\right] + \zeta_2(n)\lambda\underbrace{\left(\epsilon_n\psi_n - \mathbb{E}\left[\epsilon_n\psi_n|\mathcal{F}(n)\right]\right)}_{*} \\ &+ \zeta_2(n)\lambda\underbrace{\mathbb{E}\left[(\epsilon_n - \epsilon^{\theta_n})\psi_n|\mathcal{F}(n)\right]}_{+}. \end{aligned} \tag{74}$$

Since Algorithm 2 uses an unbiased estimator for  $\rho$ , we have  $\widehat{\rho}_{n+1} \rightarrow \rho(\theta_n)$ , and thus,  $\xi_n \rightarrow 0$ . The terms (+) asymptotically vanish in light of Lemma 9 (Critic convergence). Finally the terms (\*) can be seen to vanish using standard martingale arguments [cf. Theorem 2 in Bhatnagar et al. (2009a)]. Thus, (73) can be seen to be equivalent in an asymptotic sense to

$$\theta_{n+1} = \Gamma\left(\theta_n - \zeta_2(n)\left[1 + 2\lambda\rho(\theta_n)\right]\mathbb{E}\left[\delta^{\theta_n}\psi_n|\mathcal{F}(n)\right] + \zeta_2(n)\lambda\mathbb{E}\left[\epsilon^{\theta_n}\psi_n|\mathcal{F}(n)\right]\right). \tag{75}$$

From the foregoing, it can be seen that the actor recursion in (47) asymptotically tracks the stable fixed points of the ODE

$$\dot{\theta}_t = \check{\Gamma}\left(\nabla L(\theta_t, \lambda) + \mathcal{B}(\theta_t, \lambda)\right). \tag{76}$$

Note that the bias of Algorithm 2 in estimating  $\nabla L(\theta, \lambda)$  is (see Lemma 5)

$$\mathcal{B}(\theta, \lambda) = \sum_x D^\theta(x) \left\{ - (1 + 2\lambda\rho(\theta)) [\nabla \bar{V}^\theta(x) - \nabla v^{\theta^\top} \phi_v(x)] + \lambda [\nabla \bar{U}^\theta(x) - \nabla u^{\theta^\top} \phi_u(x)] \right\}.$$

Since the bias  $\sup_\theta \|\mathcal{B}(\theta, \lambda)\| \rightarrow 0$  by assumption, the trajectories (76) converge to those of (55) uniformly on compacts for the same initial condition and the claim follows.  $\square$

**Remark 13 (Bias in estimating gradient)** We do not always expect that  $\sup_\theta \|\mathcal{B}(\theta, \lambda)\| \rightarrow 0$ . However, if there is no bias or negligibly small bias in the actor-critic algorithm, which is directly related to the choice of the critic’s function space, then we will definitely gain from using actor-critic instead of policy gradient. Note that the choice between actor-critic and policy gradient is a bias–variance tradeoff, and similar to any other bias–variance tradeoff, if the variance reduction is more significant (given the number of samples used to estimate each gradient) than the introduced bias, then it would be advantageous to use actor-critic instead of policy gradient. Also note that this tradeoff exists even in the original form (risk neutral) of actor-critic and policy gradient and has nothing to do with the risk-sensitive objective function studied in this paper. For more details on this, we refer the reader to Theorem 2 and Remark 2 in [Bhatnagar et al. \(2009b\)](#).

**Step 3:  $\lambda$  Convergence and overall convergence of the algorithm**

As in the discounted setting, we first show that the  $\lambda$ -recursion converges and then prove convergence to a local saddle point of  $L(\theta, \lambda)$ . Consider the ODE

$$\dot{\lambda}_t = \check{\Gamma}_\lambda(\Lambda(\theta^{\lambda_t}) - \alpha), \tag{77}$$

where  $\check{\Gamma}_\lambda$  is a projection operator that forces the evolution of  $\lambda$  via (65) is within  $[0, \lambda_{\max}]$  and is defined in (66).

**Theorem 8**  $\lambda_n \rightarrow \mathcal{F}$  almost surely as  $t \rightarrow \infty$ , where  $\mathcal{F} \triangleq \{\lambda \mid \lambda \in [0, \lambda_{\max}], \check{\Gamma}_\lambda(\Lambda(\theta^\lambda) - \alpha) = 0, \theta^\lambda \in \mathcal{Z}_\lambda\}$ .

*Proof* The proof follows in a similar manner as that of Theorem 3 in [Bhatnagar and Lakshmanan \(2012\)](#).  $\square$

As in the discounted setting, the following proposition claims that the limit  $\theta^{\lambda^*}$  corresponding to  $\lambda^*$  satisfies the variance constraint in (40), i.e.,

**Proposition 2** For any  $\lambda^*$  in  $\hat{\mathcal{F}} \triangleq \{\lambda \mid \lambda \in [0, \lambda_{\max}], \check{\Gamma}_\lambda[\Lambda^{\theta^\lambda}(x^0) - \alpha] = 0, \theta^\lambda \in \mathcal{Z}_\lambda\}$ , the corresponding limiting point  $\theta^{\lambda^*}$  satisfies the variance constraint  $\Lambda^{\theta^{\lambda^*}}(x^0) \leq \alpha$ .

Using arguments similar to that used to prove convergence of RS-SPSA-G, it can be shown that that the ODE (77) is equivalent to  $\dot{\lambda}_t = \check{\Gamma}_\lambda[\nabla_\lambda L(\theta^{\lambda_t}, \lambda_t)]$  and thus, the actor parameters  $(\theta_n, \lambda_n)$  updated according to (47) converge to a (local) saddle point  $(\theta^{\lambda^*}, \lambda^*)$  of  $L(\theta, \lambda)$ . Moreover, the limiting point  $\theta^{\lambda^*}$  satisfies the variance constraint in (40).

**9 Experimental results**

We evaluate our algorithms in the context of a traffic signal control application. The objective in our formulation is to minimize the total number of vehicles in the system, which indirectly

minimizes the delay experienced by the system. The motivation behind using a risk-sensitive control strategy is to reduce the variations in the delay experienced by road users.

### 9.1 Implementation

We consider both infinite horizon discounted and average settings for the traffic signal control MDP, formulated as in Prashanth and Bhatnagar (2011). We briefly recall their formulation here: The state at each time  $t$ ,  $x_n$ , is the vector of queue lengths and elapsed times and is given by  $x_n = (q_1(n), \dots, q_N(n), t_1(n), \dots, t_N(n))$ , where  $N$  is the number of signalled lanes in the road network considered. Here  $q_i$  and  $t_i$  denote the queue length and elapsed time since the signal turned to red on lane  $i$ . The actions  $a_n$  belong to the set of feasible sign configurations. The single-stage cost function  $h(x_n)$  is defined as follows:

$$\begin{aligned}
 h(x_n) = & r_1 * \left[ \sum_{i \in I_p} r_2 * q_i(n) + \sum_{i \notin I_p} s_2 * q_i(n) \right] \\
 & + s_1 * \left[ \sum_{i \in I_p} r_2 * t_i(n) + \sum_{i \notin I_p} s_2 * t_i(n) \right], \tag{78}
 \end{aligned}$$

where  $r_i, s_i \geq 0$  such that  $r_i + s_i = 1$  for  $i = 1, 2$  and  $r_2 > s_2$ . The set  $I_p$  is the set of prioritized lanes in the road network considered. While the weights  $r_1, s_1$  are used to differentiate between the queue length and elapsed time factors, the weights  $r_2, s_2$  help in prioritization of traffic.

Given the above traffic control setting, we aim to minimize both the long run discounted and average sum of the cost function  $h(x_n)$  in (78). The underlying policy that guides the selection of the sign configuration in each of the algorithms we implemented (see below for the complete list) is a parameterized Boltzmann family and has the form

$$\mu_\theta(x, a) = \frac{e^{\theta^\top \phi_{x,a}}}{\sum_{a' \in \mathcal{A}(x)} e^{\theta^\top \phi_{x,a'}}}, \quad \forall x \in \mathcal{X}, \quad \forall a \in \mathcal{A}. \tag{79}$$

The experiments for each algorithm that we implement is comprised of the following two phases:

*Policy search phase* Here each iteration involved the simulation run with the nominal policy parameter  $\theta$  as well as the perturbed policy parameter  $\theta^+$  (algorithm-specific). We run each algorithm for 500 iterations, where the run length for a particular policy parameter is 150 steps.

*Policy test phase* After the completion of the policy search phase, we freeze the policy parameter and run 50 independent simulations with this (converged) choice of the parameter. The results presented subsequently are averages over these 50 runs.

We implement the following algorithms using the Green Light District (GLD) simulator (Wiering et al. 2004)<sup>8</sup>:

<sup>8</sup> We would like to point out that the experimental setting involves ‘costs’ and not ‘rewards’ and the algorithms implemented should be understood as optimizing a negative reward.



Discounted setting

1. *SPSA-G* This is a first-order risk-neutral algorithm with SPSA-based gradient estimates that updates the parameter  $\theta$  as follows:

$$\theta_{n+1}^{(i)} = \Gamma_i \left( \theta_n^{(i)} + \frac{\zeta_2(n)}{\beta \Delta_n^{(i)}} (v_n^+ - v_n)^\top \phi_v(x^0) \right),$$

where the critic parameters  $v_n, v_n^+$  are updated according to (13). Note that this is a two-timescale algorithm with a TD critic on the faster timescale and the actor on the slower timescale. Unlike RS-SPSA-G, this algorithm, being risk-neutral, does not involve the Lagrange multiplier recursion.

2. *SF-G* This is a first-order risk-neutral algorithm that is similar to SPSA-G, except that the gradient estimation scheme used here is based on the smoothed functional (SF) technique. The update of the policy parameter in this algorithm is given by

$$\theta_{n+1}^{(i)} = \Gamma_i \left( \theta_n^{(i)} + \zeta_2(n) \left( \frac{\Delta_n^{(i)}}{\beta} (v_n^+ - v_n)^\top \phi_v(x^0) \right) \right).$$

3. *SPSA-N* This is a risk-neutral algorithm and is the second-order counterpart of SPSA-G. The Hessian update in this algorithm is as follows: For  $i, j = 1, \dots, \kappa_1, i < j$ , the update is

$$H_{n+1}^{(i,j)} = H_n^{(i,j)} + \zeta_2'(n) \left[ \frac{(v_n - v_n^+)^\top \phi_v(x^0)}{\beta^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}} - H_n^{(i,j)} \right], \tag{80}$$

and for  $i > j$ , we set  $H_{n+1}^{(i,j)} = H_{n+1}^{(j,i)}$ . As in RS-SPSA-N, let  $M_n \triangleq H_n^{-1}$ , where  $H_n = \Upsilon([H_n^{(i,j)}]_{i,j=1}^{\kappa_1})$ . The actor updates the parameter  $\theta$  as follows:

$$\theta_{n+1}^{(i)} = \Gamma_i \left[ \theta_n^{(i)} + \zeta_2(n) \sum_{j=1}^{\kappa_1} M_n^{(i,j)} \left( \frac{(v_n^+ - v_n)^\top \phi_v(x^0)}{\beta \Delta_n^{(j)}} \right) \right]. \tag{81}$$

The rest of the symbols, including the critic parameters, are as in RS-SPSA-N.

4. *SF-N* This is a risk-neutral algorithm and is the second-order counterpart of SF-G. It updates the Hessian and the actor as follows: For  $i, j, k = 1, \dots, \kappa_1, j < k$ , the Hessian update is

$$\begin{aligned} \text{Hessian: } H_{n+1}^{(i,i)} &= H_n^{(i,i)} + \zeta_2'(n) \left[ \frac{(\Delta_n^{(i)^2} - 1)}{\beta^2} (v_n - v_n^+)^\top \phi_v(x^0) - H_n^{(i,i)} \right], \\ H_{n+1}^{(j,k)} &= H_n^{(j,k)} + \zeta_2'(n) \left[ \frac{\Delta_n^{(j)} \Delta_n^{(k)}}{\beta^2} (v_n - v_n^+)^\top \phi_v(x^0) - H_n^{(j,k)} \right], \end{aligned}$$

and for  $j > k$ , we set  $H_{n+1}^{(j,k)} = H_{n+1}^{(k,j)}$ . As before, let  $M_n \triangleq H_n^{-1}$ , with  $H_n$  formed as in SPSA-N. Then, the actor update for the parameter  $\theta$  is as follows:

$$\text{Actor: } \theta_{n+1}^{(i)} = \Gamma_i \left[ \theta_n^{(i)} + \zeta_2(n) \sum_{j=1}^{\kappa_1} M_n^{(i,j)} \frac{\Delta_n^{(j)}}{\beta} (v_n^+ - v_n)^\top \phi_v(x^0) \right].$$

The rest of the symbols, including the critic parameters, are as in RS-SPSA-N.

5. *RS-SPSA-G* This is the first-order risk-sensitive actor-critic algorithm that attempts to solve (40) and updates according to (19).
6. *RS-SF-G* This is a first-order algorithm and the risk-sensitive variant of SF-G that updates the actor according to (20).
7. *RS-SPSA-N* This is a second-order risk-sensitive algorithm that estimates gradient and Hessian using SPSA and updates them according to (31).
8. *RS-SF-N* This second-order risk-sensitive algorithm is the SF counterpart of RS-SPSA-N, and updates according to (36).
9. *TAMAR* This is a straightforward adaptation of the algorithm proposed in Tamar et al. (2012). The main difference between this and our algorithms is that TAMAR uses a Monte Carlo critic, while our algorithms employ a TD critic. Moreover, TAMAR incorporates the  $\lambda$ -recursion that is identical to that of our algorithms (see Eq. 21). In contrast, the algorithm proposed in Tamar et al. (2012) is for a fixed  $\lambda$  that may not be optimal. Note that even though TAMAR is an algorithm proposed for a stochastic shortest path (SSP) setting, it can be implemented in the traffic signal control problem since we truncate the simulation after 150 steps.

Let  $D_n$  denote the sum of rewards obtained from a single simulation run in the policy search phase. Further, let  $z_n := \sum_{m=0}^{150} \nabla \ln \mu_\theta(x_m, a_m)$  denote the likelihood derivative. Then, the update rule is given by

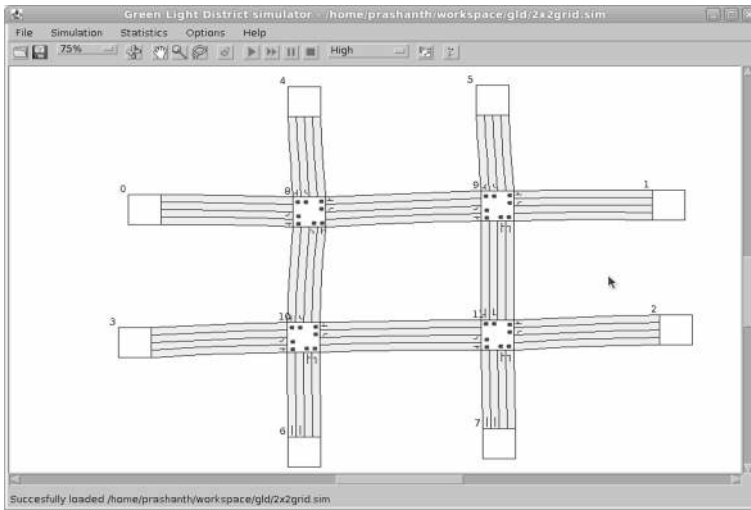
$$\begin{aligned} \tilde{V}_{n+1} &= \tilde{V}_n + \zeta_3(n)(D_n - \tilde{V}_n) \\ \tilde{\Lambda}_{n+1} &= \tilde{\Lambda}_n + \zeta_3(n)(D_n^2 - \tilde{V}_n^2 - \tilde{\Lambda}_n) \\ \theta_{n+1}^{(i)} &= \Gamma_i \left( \theta_n + \zeta_2(n)(D_n - \lambda_n(D_n^2 - 2D_n \tilde{V}_n))z_n^{(i)} \right), i = 1, \dots, \kappa_1, \\ \lambda_{n+1} &= \Gamma_\lambda \left[ \lambda_n + \zeta_1(n)(\Lambda_n - \alpha) \right]. \end{aligned}$$

Note that the  $\theta$ -recursion above corrects an error (we believe it is a typo) in the corresponding update rule [i.e., Eq. 13 in Tamar et al. (2012)]. Unlike the above, Eq. 13 in Tamar et al. (2012) is missing the multiplier  $D_n$  in the last term in the  $\theta$ -recursion. The latter multiplier originates from the gradient of the value function [see Lemma 4.2 in Tamar et al. (2012)].

*Average setting*

1. *AC* This is an actor-critic algorithm that minimizes the long-run average sum of the single-stage cost function  $h(x_n)$ , without considering any risk criteria. This is similar to Algorithm 1 in Bhatnagar et al. (2009a).
2. *RS-AC* This is the risk-sensitive actor-critic algorithm that attempts to solve (40) and is described in Sect. 6.

All our algorithms incorporate function approximation owing to the curse of dimensionality associated with larger road networks. For instance, assuming only 20 vehicles per lane of a  $2 \times 2$ -grid network, the cardinality of the state space is approximately of the order  $10^{32}$  and the situation is aggravated as the size of the road network increases. We employ the feature selection scheme from Prashanth and Bhatnagar (2012) in each of our algorithms. The features are obtained with coarse congestion estimates along the lanes of the road network as input. For instance, instead of the exact queue length on a lane, the coarse congestion information specifies whether the queue length was between 0 to  $L_1$  units, between  $L_1$  and



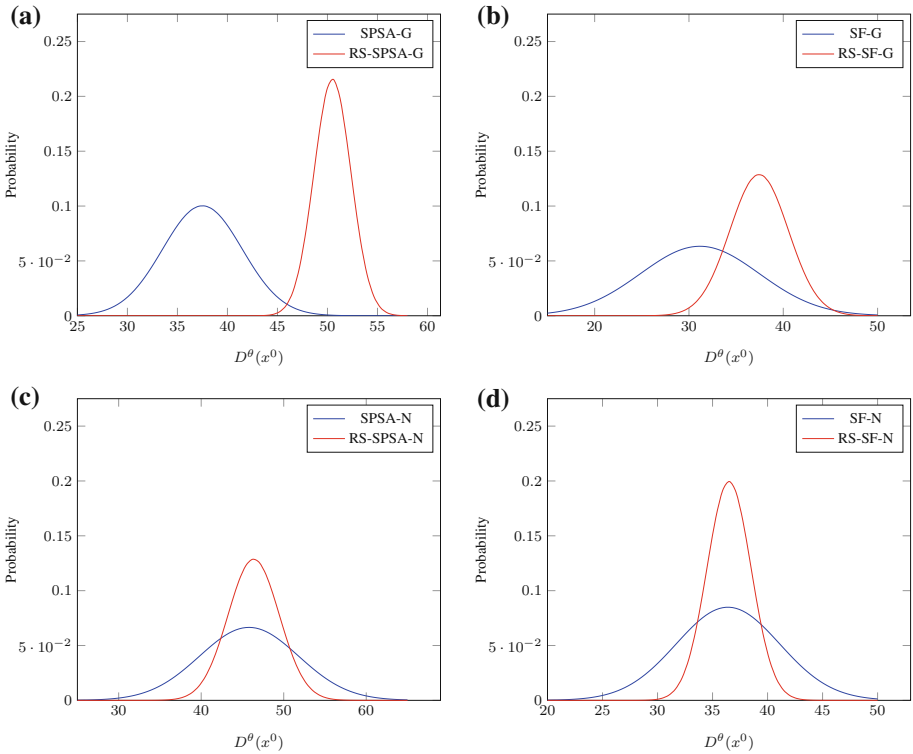
**Fig. 2** The 2 × 2-grid network used in our traffic signal control experiments

$L_2$  units or greater than  $L_2$  units. By placing magnetic sensor loops on the lane at distances  $L_1$  and  $L_2$  from the junction, it is possible to obtain coarse congestion information. Assume another threshold  $T_1$  for the elapsed time. Using the aforementioned coarse inputs on queue lengths and elapsed times for each lane in the road network considered, the feature selection is performed in a graded fashion as follows: queue length less than  $L_1$  and elapsed time less than  $T_1$  leading a to feature value that recommends red light, queue length more than  $L_2$  and elapsed time more than  $T_1$  leading to a feature value that recommends green light, with the feature values for the intermediate scenarios graded appropriately. For a detailed description of the feature selection scheme, the reader is referred to Section V-B of Prashanth and Bhatnagar (2012). The values  $L_1$ ,  $L_2$  and  $T_1$  are set to 6, 14 and 130, as recommended in Prashanth and Bhatnagar (2012).

Figure 2 shows a snapshot of the road network used for conducting the experiments from GLD simulator. Traffic is added to the network at each time step from the edge nodes. The spawn frequencies specify the rate at which traffic is generated at each edge node and follow a Poisson distribution. The spawn frequencies are set such that the proportion of the number of vehicles on the main roads (the horizontal ones in Fig. 2) to those on the side roads is in the ratio of 100:5. This setting is close to what is observed in practice and has also been used for instance in Prashanth and Bhatnagar (2011) and Prashanth and Bhatnagar (2012). In all our experiments, we set the weights in the single stage cost function (78) as follows:  $r_1 = r_2 = 0.5$  and  $r_2 = 0.6, s_2 = 0.4$ . For the SPSA and SF-based algorithms in the discounted setting, we set the parameter  $\delta = 0.2$  and the discount factor  $\gamma = 0.9$ . The parameter  $\alpha$  in the formulations (40) and (3) was set to 20. The step-size sequences are chosen as follows:

$$\zeta_1(n) = \frac{1}{n}, \quad \zeta_2(n) = \frac{1}{n^{0.75}}, \quad \zeta'_2(n) = \frac{1}{n^{0.7}}, \quad \zeta_3(n) = \frac{1}{n^{0.66}}, \quad n \geq 1. \quad (82)$$

Further, the constant  $k$  related to  $\zeta_4(n)$  in the risk-sensitive average reward algorithm is set to 1. It is easy to see that the choice of step-sizes above satisfies (A4). The projection operator  $\Gamma_i$  was set to project the iterate  $\theta^{(i)}$  onto the set  $[0, 10]$ , for all  $i = 1, \dots, \kappa_1$ , while the projection



**Fig. 3** Performance comparison in the discounted setting using the distribution of  $D^\theta(x^0)$ . **a** SPSA-G versus RS-SPSA-G, **b** SF-G versus RS-SF-G, **c** SPSA-N versus RS-SPSA-N, **d** SF-N versus RS-SF-N

operator for the Lagrange multiplier used the set  $[0, 1000]$ . The initial policy parameter  $\theta_0$  was set to the  $\kappa_1$ -dimensional vector of ones. All the experiments were performed on a 2.53GHz Intel quad core machine with 3.8GB RAM.

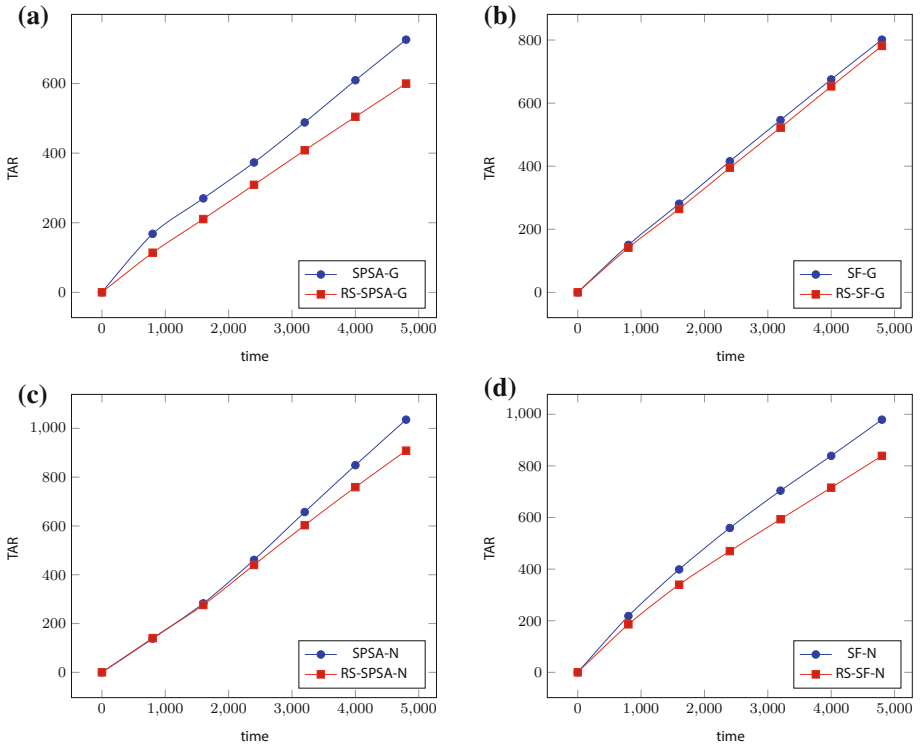
### 9.2 Results

Figure 3 shows the distribution of the discounted cumulative cost  $D^\theta(x^0)$  for the algorithms in the discounted setting. Figure 4 shows the total arrived road users (TAR) obtained for all the algorithms in the discounted setting, whereas Fig. 5 presents the average junction waiting time (AJWT) for the first-order SF-based algorithm RS-SF-G.<sup>9</sup> TAR is a throughput metric that measures the number of road users who have reached their destination, whereas AJWT is a delay metric that quantifies the average delay experienced by the road users.

The performance of the algorithms in the average setting is presented in Fig. 6. In particular, Fig. 6a shows the distribution of the average reward  $\rho$ , while Fig. 6b presents the average junction waiting time (AJWT) for the average cost algorithms.

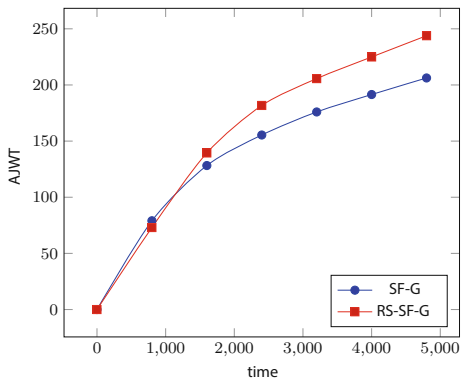
**Observation 1** *Risk-sensitive algorithms that we propose result in a long-term (discounted or average) cost that is higher than their risk-neutral variants, but with a significantly lower empirical variance of the cost in both discounted as well as average cost settings.*

<sup>9</sup> The AJWT performance of the other algorithms in the discounted setting is similar and the corresponding plots are omitted here.



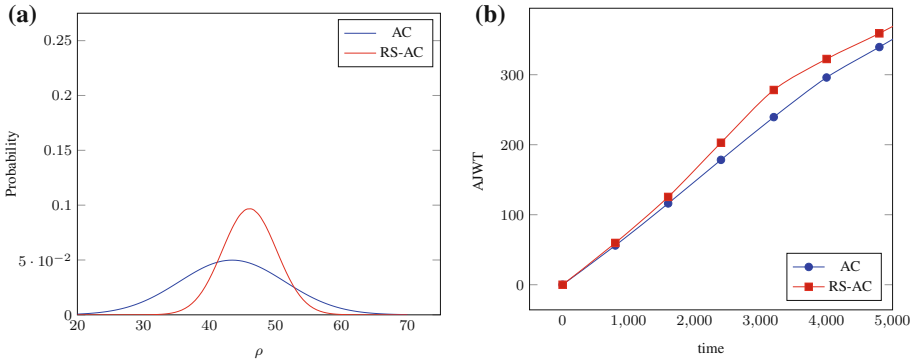
**Fig. 4** Performance comparison of the algorithms in the discounted setting using the total arrived road users (TAR). **a** SPSA-G versus RS-SPSA-G, **b** SF-G versus RS-SF-G, **c** SPSA-N versus RS-SPSA-N, **d** SF-N versus RS-SF-N

**Fig. 5** Performance comparison of the first-order SF-based algorithms, SF-G and RS-SF-G, using the average junction waiting time (AJWT)



The above observation is apparent from Figs. 3 and 6a, which present results for discounted and average cost settings respectively.

**Observation 2** From a traffic signal control application standpoint, the risk-sensitive algorithms exhibit a mean throughput/delay that is close to that of the corresponding risk-neutral algorithms, but with a lower empirical variance in throughput/delay.



**Fig. 6** Performance comparison of the risk-neutral (AC) and risk-sensitive (RS-AC) average reward actor-critic algorithms using two different metrics. **a** average reward  $\rho$  distribution, **b** average junction waiting time

**Table 1** Throughput (TAR) for algorithms in the discounted setting: standard deviation from 50 independent simulations shown after  $\pm$

Algorithm	Risk-neutral	Risk-sensitive
SPSA-G	754.84 $\pm$ 317.06	622.38 $\pm$ 28.36
SF-G	832.34 $\pm$ 82.24	810.82 $\pm$ 36.56
SPSA-N	1077.2.66 $\pm$ 250.42	942.3 $\pm$ 65.77
SF-N	1013.62 $\pm$ 152.22	870.5 $\pm$ 61.61

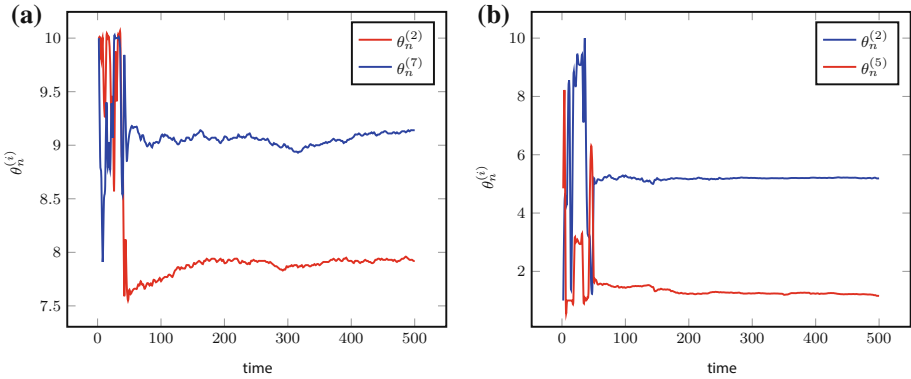
Figures 4, 5 and 6b validate the first part of the observation above, while the results for the discounted risk-sensitive algorithms in Table 1 substantiate the second part in the above observation. In particular, Table 1 presents the mean and standard deviation of the final TAR value (i.e., the TAR value observed at the end of the policy test phase) for both first-order and second-order algorithms in the discounted setting and it is evident that the risk-sensitive algorithms exhibit a lower empirical variance in TAR when compared to their risk-neutral counterparts.

From the results in Figs. 3, 4 and Table 1, it is apparent that the second-order schemes (RS-SPSA-N and RS-SF-N) in the discounted setting exhibit better results in comparison to first-order methods (RS-SPSA-G and RS-SF-G), from the mean and variance of the long-term discounted cost as well as the throughput (TAR) performance.

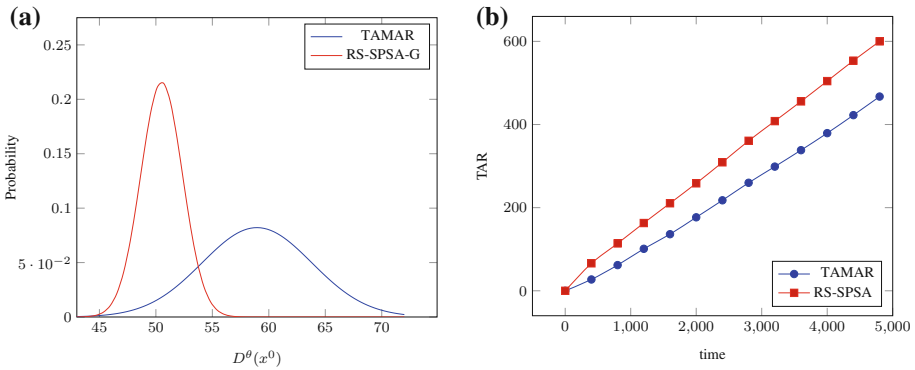
**Observation 3** *The policy parameter  $\theta$  converges for the risk-sensitive algorithms.*

The above observation is validated for SPSA based algorithms in the discounted setting in Fig. 7a, b. Note that we established theoretical convergence of our algorithms earlier (see Sects. 7, 8) and these plots confirm the same. Further, these plots also show that the transient period, i.e., the initial phase when  $\theta$  has not converged, is short. Similar observations hold for the other algorithms as well. The results of this section indicate the rapid empirical convergence of our proposed algorithms. This observation coupled with the fact that they guarantee low variance of return, make them attractive for implementation in risk-constrained systems.

**Observation 4** *RS-SPSA, which is based on an actor-critic architecture, outperforms TAMAR, which employs a policy gradient approach.*



**Fig. 7** Convergence of SPSA based algorithms in the discounted setting—illustration using two (arbitrarily chosen) coordinates of the parameter  $\theta$ . **a** RS-SPSA-G, **b** RS-SPSA-N



**Fig. 8** Performance comparison of RS-SPSA and TAMAR (Tamar et al. 2012) algorithms using two different metrics. **a** Distribution of  $D^\theta(x^0)$ , **b** total arrived road users (TAR)

Figure 8 shows the distribution of the cumulative cost  $D^\theta(x^0)$  and the total arrived road users (TAR) obtained for TAMAR and RS-SPSA algorithms. It is evident that RS-SPSA performs better than TAMAR in terms of mean as well as variance of the cumulative cost and also in terms of the throughput (TAR) observed. These results illustrate the benefits of using an actor-critic architecture. Note that both algorithms use the same parameterized Boltzmann policy (see Eq. 79) and the results have been obtained with the same number of updates, i.e., 500 SPSA updates, which is equivalent to 1000 policy gradient updates, as each iteration of SPSA uses two trajectories to estimate the gradient. While the results in Fig. 8 implicitly indicate that RS-SPSA gives a better estimate of the gradient in comparison to TAMAR, we make this observation explicit in Table 2, which plots the results from the following experiment:

**Step 1** (True gradient estimation): Estimate  $\nabla_\theta \Lambda(x^0)$  using the likelihood ratio method, along the lines of Lemma 4.2 in Tamar et al. (2012). For this purpose, simulate a large number, say  $T_1 = 1000$ , of trajectories of the underlying MDP (as before, we truncate the trajectories to 150 steps). This estimate can be safely assumed to be very close to the true gradient and hence, we shall use it as the benchmark for comparing our SPSA based actor-critic scheme vs. the policy gradient approach of TAMAR.

**Table 2**  $\ell_2$  distance between gradient estimated using either RS-SPSA or TAMAR and a likelihood ratio benchmark: mean and standard error from 100 replications shown before and after  $\pm$ , respectively

Policy	TAMAR	RS-SPSA
$\theta^{(i)} = 0.5, \forall i$	$655.77 \pm 18.65$	$142.1 \pm 9.56$
$\theta^{(i)} = 1, \forall i$	$694.99 \pm 16.67$	$149.82 \pm 10.25$
$\theta^{(i)} = 2, \forall i$	$720.99 \pm 14.85$	$146.67 \pm 9.31$
$\theta^{(i)} = 5, \forall i$	$941.53 \pm 25.39$	$200.08 \pm 13.25$
$\theta^{(i)} = 7, \forall i$	$1167.78 \pm 37.14$	$210.73 \pm 12.97$
$\theta^{(i)} = 10, \forall i$	$1489.32 \pm 43.43$	$277.15 \pm 11.93$

**Step 2** (Policy gradient approach of TAMAR):

- Fix a policy parameter.
- Run two simulations for the policy above.
- Estimate  $\nabla_{\theta} \Lambda(x^0)$  using the scheme in TAMAR.
- Calculate the distance (in  $\ell_2$  norm) between the estimate above and the benchmark defined in Step 1.

Repeat the above steps 100 times and collect the mean and standard errors of the  $\ell_2$  distance in the last step above.

**Step 3** (Actor-critic approach of RS-SPSA):

- Fix a policy parameter.
- Run two simulations—one for the unperturbed parameter and the another for the perturbed parameter, where perturbation is performed as in RS-SPSA (see Sect. 4.3).
- Estimate  $\nabla_{\theta} \Lambda(x^0)$  using the scheme in RS-SPSA.
- Calculate the distance (in  $\ell_2$  norm) between the estimate above and the benchmark defined in Step 1.

Repeat the above steps 100 times and collect the mean and standard errors of the relevant  $\ell_2$  distance as in Step 2.

From the mean and standard errors presented in Table 2 for six different policies, it is evident that RS-SPSA produces more accurate estimates of the policy gradients than TAMAR, which explains its faster convergence (compared to TAMAR) in the experiments of Fig. 8. The trend did not change by having the true gradient estimated from a larger number of trajectories. In particular, with  $T_1 = 5000$  (see Step 1 above), the relevant  $\ell_2$  distances for TAMAR and RS-SPSA were observed to be  $(683.06 \pm 26.75)$  and  $(143.02 \pm 14.44)$ , respectively for the policy  $\theta^{(i)} = 1, \forall i$ .

## 10 Conclusions and future work

We proposed novel actor-critic algorithms for control in risk-sensitive discounted and average reward MDPs. All our algorithms involve a TD critic on the fast timescale, a policy gradient (actor) on the intermediate timescale, and a dual ascent for Lagrange multipliers on the slowest timescale. In the discounted setting, we pointed out the difficulty in estimating the gradient of the variance of the return and incorporated simultaneous perturbation based SPSA and SF approaches for gradient estimation in our algorithms. The average setting, on the other



hand, allowed for an actor to employ compatible features to estimate the gradient of the variance. We provided proofs of convergence to locally (risk-sensitive) optimal policies for all the proposed algorithms. Further, using a traffic signal control application, we observed that our algorithms resulted in lower variance empirically as compared to their risk-neutral counterparts.

As future work, it would be interesting to develop a risk-sensitive algorithm that uses a single trajectory in the discounted setting. An orthogonal direction of future research is to obtain finite-time bounds on the quality of the solution obtained by our algorithms. As mentioned earlier, this is challenging as, to the best of our knowledge, there are no convergence rate results available for multi-timescale stochastic approximation schemes, and hence, for actor-critic algorithms.

**Acknowledgments** This work was supported in part by the National Science Foundation (NSF) under Grants CMMI-1434419, CNS-1446665, and CMMI-1362303, and by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-15-10050.

## References

- Altman, E. (1999). *Constrained Markov decision processes* (Vol. 7). Boca Raton: CRC Press.
- Barto, A., Sutton, R., & Anderson, C. (1983). Neuron-like elements that can solve difficult learning control problems. *IEEE Transaction on Systems, Man and Cybernetics*, 13, 835–846.
- Basu, A., Bhattacharyya, T., & Borkar, V. (2008). A learning algorithm for risk-sensitive cost. *Mathematics of Operations Research*, 33(4), 880–898.
- Baxter, J., & Bartlett, P. (2001). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15, 319–350.
- Bertsekas, D. (1995). *Dynamic programming and optimal control*. Belmont, MA: Athena Scientific.
- Bertsekas, D. (1999). *Nonlinear programming*. Belmont, MA: Athena Scientific.
- Bertsekas, D., & Tsitsiklis, J. (1996). *Neuro-dynamic programming*. Belmont, MA: Athena Scientific.
- Bhatnagar, S. (2005). Adaptive multivariate three-timescale stochastic approximation algorithms for simulation based optimization. *ACM Transactions on Modeling and Computer Simulation*, 15(1), 74–107.
- Bhatnagar, S. (2007). Adaptive Newton-based multivariate smoothed functional algorithms for simulation optimization. *ACM Transactions on Modeling and Computer Simulation*, 18(1), 1–35.
- Bhatnagar, S. (2010). An actor-critic algorithm with function approximation for discounted cost constrained Markov decision processes. *Systems & Control Letters*, 59(12), 760–766.
- Bhatnagar, S., & Lakshmanan, K. (2012). An online actor-critic algorithm with function approximation for constrained Markov decision processes. *Journal of Optimization Theory and Applications*, 153(3), 688–708.
- Bhatnagar, S., Fu, M., Marcus, S., & Wang, I. (2003). Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. *ACM Transactions on Modeling and Computer Simulation*, 13(2), 180–209.
- Bhatnagar, S., Sutton, R., Ghavamzadeh, M., & Lee, M. (2007). Incremental natural actor-critic algorithms. In: *Proceedings of advances in neural information processing systems* (Vol. 20, pp. 105–112).
- Bhatnagar, S., Sutton, R., Ghavamzadeh, M., & Lee, M. (2009a). Natural actor-critic algorithms. *Automatica*, 45(11), 2471–2482.
- Bhatnagar, S., Sutton, R., Ghavamzadeh, M., & Lee, M. (2009b). Natural actor-critic algorithms. Technical report TR09-10, Department of Computing Science, University of Alberta.
- Bhatnagar, S., Hemachandra, N., & Mishra, V. (2011). Stochastic approximation algorithms for constrained optimization via simulation. *ACM Transactions on Modeling and Computer Simulation*, 21(3), 15.
- Bhatnagar, S., Prasad, H., & Prashanth, L. (2013). *Stochastic recursive algorithms for optimization* (Vol. 434). Berlin: Springer.
- Borkar, V. (2001). A sensitivity formula for the risk-sensitive cost and the actor-critic algorithm. *Systems & Control Letters*, 44, 339–346.
- Borkar, V. (2002). Q-learning for risk-sensitive control. *Mathematics of Operations Research*, 27, 294–311.
- Borkar, V. (2005). An actor-critic algorithm for constrained Markov decision processes. *Systems & Control Letters*, 54(3), 207–213.

- Borkar, V. (2008). *Stochastic approximation: A dynamical systems viewpoint*. Cambridge: Cambridge University Press.
- Borkar, V. (2010). Learning algorithms for risk-sensitive control. In *Proceedings of the nineteenth international symposium on mathematical theory of networks and systems* (pp. 1327–1332).
- Borkar, V. S., & Meyn, S. P. (2000). The ode method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2), 447–469.
- Chen, H., Duncan, T., & Pasik-Duncan, B. (1999). A Kiefer–Wolfowitz algorithm with randomized differences. *IEEE Transactions on Automatic Control*, 44(3), 442–453.
- Delage, E., & Mannor, S. (2010). Percentile optimization for Markov decision processes with parameter uncertainty. *Operations Research*, 58(1), 203–213.
- Dippon, J., & Renz, J. (1997). Weighted means in stochastic approximation of minima. *SIAM Journal on Control and Optimization*, 35(5), 1811–1827.
- Fabian, V. (1968). On asymptotic normality in stochastic approximation. *The Annals of Mathematical Statistics*, 39, 1327–1332.
- Filar, J., Kallenberg, L., & Lee, H. (1989). Variance-penalized Markov decision processes. *Mathematics of Operations Research*, 14(1), 147–161.
- Filar, J., Krass, D., & Ross, K. (1995). Percentile performance criteria for limiting average Markov decision processes. *IEEE Transaction of Automatic Control*, 40(1), 2–10.
- Gill, P., Murray, W., & Wright, M. (1981). *Practical optimization*. London: Academic press.
- Howard, R., & Matheson, J. (1972). Risk sensitive Markov decision processes. *Management Science*, 18(7), 356–369.
- Katkovnik, V., & Kulchitsky, Y. (1972). Convergence of a class of random search algorithms. *Automatic Remote Control*, 8, 81–87.
- Konda, V., & Tsitsiklis, J. (2000). Actor-critic algorithms. In *Proceedings of advances in neural information processing systems* (Vol. 12, pp. 1008–1014).
- Konda, V. R., & Tsitsiklis, J. N. (2004). Convergence rate of linear two-time-scale stochastic approximation. *Annals of Applied Probability*, 14(2), 796–819.
- Korda, N., & Prashanth, L. (2015). On TD (0) with function approximation: Concentration bounds and a centered variant with exponential convergence. In *International conference on machine learning (ICML)*.
- Kushner, H., & Clark, D. (1978). *Stochastic approximation methods for constrained and unconstrained systems*. Berlin: Springer.
- Mannor, S., & Tsitsiklis, J. (2011). Mean–variance optimization in Markov decision processes. In *Proceedings of the twenty-eighth international conference on machine learning* (pp. 177–184).
- Mannor, S., & Tsitsiklis, J. N. (2013). Algorithmic aspects of mean–variance optimization in Markov decision processes. *European Journal of Operational Research*, 231(3), 645–653.
- Marbach, P. (1998). Simulated-based methods for Markov decision processes. Ph.D. thesis, Massachusetts Institute of Technology.
- Mas-Colell, A., Whinston, M., & Green, J. (1995). *Microeconomic theory*. Oxford: Oxford University Press.
- Mihatsch, O., & Neuneier, R. (2002). Risk-sensitive reinforcement learning. *Machine Learning*, 49(2), 267–290.
- Milgrom, P., & Segal, I. (2002). Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2), 583–601.
- Nilim, A., & Ghaoui, L. E. (2005). Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5), 780–798.
- Peters, J., Vijayakumar, S., & Schaal, S. (2005). Natural actor-critic. In *Proceedings of the sixteenth european conference on machine learning* (pp. 280–291).
- Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4), 838–855.
- Prashanth, L., & Bhatnagar, S. (2011). Reinforcement learning with function approximation for traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 12(2), 412–421.
- Prashanth, L., & Bhatnagar, S. (2012). Threshold tuning using stochastic optimization for graded signal control. *IEEE Transactions on Vehicular Technology*, 61(9), 3865–3880.
- Prashanth, L., & Ghavamzadeh, M. (2013). Actor-critic algorithms for risk-sensitive MDPs. In *Proceedings of advances in neural information processing systems* (Vol. 26, pp. 252–260).
- Prashanth, L., Jie, C., Fu, M., Marcus, S., & Szepesvari, C. (2016). Cumulative prospect theory meets reinforcement learning: Prediction and control. In *Proceedings of the 33rd international conference on machine learning* (pp. 1406–1415).
- Puterman, M. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. London: Wiley.
- Ruppert, D. (1991). Stochastic approximation. In B. K. Ghosh & P. K. Sen (Eds.), *Handbook of Sequential Analysis* (pp. 503–529). New York: Marcel Dekker.

- Ruszczyński, A. (2010). Risk-averse dynamic programming for Markov decision processes. *Mathematical Programming*, 125, 235–261.
- Schweitzer, P. J. (1968). Perturbation theory and finite Markov chains. *Journal of Applied Probability*, 5, 401–413.
- Sharpe, W. (1966). Mutual fund performance. *Journal of Business*, 39(1), 119–138.
- Shen, Y., Stannat, W., & Obermayer, K. (2013). Risk-sensitive Markov control processes. *SIAM Journal on Control and Optimization*, 51(5), 3652–3672.
- Sion, M. (1958). On general minimax theorems. *Pacific Journal of Mathematics*, 8(1), 171–176.
- Sobel, M. (1982). The variance of discounted Markov decision processes. *Applied Probability*, 19, 794–802.
- Spall, J. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3), 332–341.
- Spall, J. (1997). A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica*, 33(1), 109–112.
- Spall, J. (2000). Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transactions on Automatic Control*, 45(10), 1839–1853.
- Styblinski, M. A., & Opalski, L. J. (1986). Algorithms and software tools for IC yield optimization based on fundamental fabrication parameters. *IEEE Transactions on Computer Aided Design CAD*, 1(5), 79–89.
- Sutton, R. (1984). *Temporal credit assignment in reinforcement learning*. Ph.D. thesis, University of Massachusetts Amherst.
- Sutton, R. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3, 9–44.
- Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge: MIT Press.
- Sutton, R., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of advances in neural information processing systems* (Vol. 12, pp. 1057–1063).
- Sutton, R. S., McAllester, D. A., Singh, S. P., Mansour, Y., et al. (1999). Policy gradient methods for reinforcement learning with function approximation. In *NIPS, Citeseer* (Vol. 99, pp. 1057–1063).
- Tamar, A., & Mannor, S. (2013). Variance adjusted actor critic algorithms. [arXiv:1310.3697](https://arxiv.org/abs/1310.3697).
- Tamar, A., Di Castro, D., & Mannor, S. (2012). Policy gradients with variance related risk criteria. In *Proceedings of the twenty-ninth international conference on machine learning* (pp. 387–396).
- Tamar, A., Di Castro, D., & Mannor, S. (2013a). Policy evaluation with variance related risk criteria in markov decision processes. [arXiv:1301.0104](https://arxiv.org/abs/1301.0104).
- Tamar, A., Di Castro, D., & Mannor, S. (2013b). Temporal difference methods for the variance of the reward to go. In *Proceedings of the thirtieth international conference on machine learning* (pp. 495–503).
- Tsitsiklis, J. N., & Van Roy, B. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5), 674–690.
- Wiering, M., Vreeken, J., van Veenen, J., & Koopman, A. (2004). Simulation and optimization of traffic in a city. In *IEEE intelligent vehicles symposium* (pp. 453–458).
- Williams, R. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8, 229–256.
- Xu, H., & Mannor, S. (2012). Distributionally robust Markov decision processes. *Mathematics of Operations Research*, 37(2), 288–300.