A NEW LEARNING ALGORITHM
FOR OPTIMAL STOPPING

VIVEK S. BORKAR[1], JERVIS PINTO[2], TARUN PRABHU[3]

Abstract: A linear programming formulation of the optimal stopping problem for Markov decision processes is approximated using linear function approximation. Using this formulation, a reinforcement learning scheme based on a primal-dual method and incorporating a sampling device called 'split sampling' is proposed and analyzed. An illustrative example from option pricing is also included.

Key words: learning algorithm, optimal stopping, linear programming, primal-dual methods, split sampling, option pricing

[1] Tata Institute of Fundamental Research, Homi Bhabha Road, Mumbai 400005, India. Research supported in part by a grant from General Motors India Pvt. Ltd. This author thanks Arnab Basu for some useful inputs.

[2] St. Francis Institute of Technology, Mumbai 400103, India. (Currently with the School of Electrical Engg. and Computer Science, Oregon State University, Corvallis, OR 97331, USA.)

[3] St. Francis Institute of Technology, Mumbai 400103, India. (Currently with the School of Computing, University of Utah, Salt Lake City, UT 84112, USA.)

1 Introduction

In recent years, there has been much activity in developing reinforcement learning algorithms for approximate dynamic programming for Markov decision processes, based on real or simulated data. This is useful when exact analytical or numerical solution is either infeasible or expensive. See [?], [?] for book length treatments of this issue and overview. Some examples of such schemes are: Q-learning, actor-critic, or TD(λ) [?]. These can be viewed as being derived from the traditional iterative schemes for MDPs such as the value or policy iteration, with additional structure (such as approximation architecture or additional averaging) built on top of it. There is, however, a third computational scheme for classical MDPs such as finite / infinite horizon discounted cost or average cost, viz., the linear programming approach. The 'primal' formulation of this is a linear program (LP for short) over the so called 'occupation measures'. Its 'dual' is an LP over functions. An extensive account of these developments may be found in [?]. Our aim here is to formulate a learning scheme for the *optimal stopping* problem based on an old formulation of optimal stopping as a linear program in the spirit of the aforementioned.

Some relevant literature is as follows:

- The LP formulation of optimal stopping is the same as the 'minimal excessive majorant function' characterization of the value function (for maximization problems) that dates back to [?], or the 'maximal subsolution' (for minimization problems) as in [?], Chapter III, section 5.1. The computational implications have been explored in [?].

- Our scheme leads to an alternative to those proposed in, e.g., [?], [?], [?], [?], [?], for option pricing, which is perhaps the prime application area for optimal stopping at present. These are based on the classical learning algorithms such as Q-learning, not on the LP formulation.

- Also motivated by finance problems, [?], [?], [?] arrive at a formulation akin to ours via an abstract duality result, but their emphasis is on finding bounds on the solution via Monte Carlo. Our scheme differs from a 'pure' Monte Carlo in that it is a reinforcement learning scheme. As observed in [?], such a scheme can be viewed as a cross between pure Monte Carlo and pure (deterministic) numerical schemes, with its per iterate computation more than the former, but less than the latter, and its fluctuations (variance) more than the latter (which is zero) and less than the former. The key difference with pure Monte Carlo is that our scheme is based on one step *conditional* averaging rather than averaging, which leads to the differences mentioned above.

We present the LP formulation and the algorithm in the next section. Section 3 provides the mathematical justification for the scheme. The focus of this work is primarily theoretical. Nevertheless, section 4 describes numerical experiments for a simple illustrative option pricing model. Section 5 sketches an extension to the infinite horizon discounted cost problem to indicate the broader applicability of the approach.

2 The algorithm

Consider a discrete time Markov chain $\{X_n\}$ taking values in a compact metric space S with the transition probability kernel $p(dy|x)$. Let $N > 0$ be a prescribed integer. Given a bounded continuous function $g : S \to \mathcal{R}$ and a discount factor $0 < \alpha < 1$, our objective is to find the optimal stopping time τ^* that maximizes

$$E[\alpha^{N\wedge\tau} g(X_{N\wedge\tau})] \tag{1}$$

over all stopping times τ w.r.t. the natural filtration of $\{X_n\}$. A standard dynamic programming argument then tells us that the value function

$$V_n^*(x) \overset{\text{def}}{=} \sup E[\alpha^{(N\wedge\tau-n)} g(X_{N\wedge\tau})|X_n = x],$$

where the supremum is over all stopping times $\geq n$, satisfies

$$\begin{aligned} V_n^*(x) &= g(x) \vee \alpha \int V_{n+1}^*(y)p(dy|x), \quad 0 \leq n < N, && (2) \\ V_N^*(x) &= g(x). && (3) \end{aligned}$$

Our scheme will be based on the following observation that essentially goes back to Dynkin, whose proof is included for the sake of completeness:

Theorem 1 $\{V_n^*\}$ above is given by the solution to the LP:

Minimize $V_0(i_0)$ s.t.

$$V_n(x) \geq g(x),\ 0 \leq n \leq N$$

$$V_n(x) \geq \alpha \sum_y p(dy|x)V_{n+1}(y),\ 0 \leq n < N$$

Proof Note that $\{V_n^*\}$ is feasible for this LP. At the same time, if $\{V_n\}$ is any other solution, then

$$\begin{aligned} V_n(x) &= \zeta_n(x) + g(i) \vee \alpha \int V_{n+1}(y)p(dy|x), \quad 0 \leq n < N, \\ V_N(x) &= \zeta_N(x) + g(x), \end{aligned}$$

for some $\zeta_n(\cdot) \geq 0, 0 \leq n \leq N$. This is simply the dynamic programming equation for the optimal stopping problem with reward

$$E[\sum_{m=n}^{N\wedge\tau} \alpha^m \zeta_m(X_m) + \alpha^{N\wedge\tau} g(X_{N\wedge\tau})].$$

(Note that for the decision to stop at time n in state x, the 'running reward' $\zeta_n(x)$ will also be granted in addition to the 'stopping reward' $g(x)$.) A standard argument then shows that

$$V_0(x) = \sup E[\sum_{m=n}^{N\wedge\tau} \alpha^m \zeta_m(X_m) + \alpha^{N\wedge\tau} g(X_{N\wedge\tau})|X_0 = x] \geq V_0^*(x)$$

where the supremum is over all stopping times. Thus the solution of the above LP does indeed coincide with the value function. □

It is worth noting that: (i) the constraints above need hold only a.s. with respect to the law of X_n for each n, and, (ii) the nonnegativity constraints $V_n(\cdot) \geq 0$ do not have to be explicitly incorporated. By Lagrange multiplier theory ([**?**], pp. 216), the above optimization problem becomes

$$\min_V \max_\lambda [L(V, \Lambda)] \tag{4}$$

where $\Lambda = [\Lambda_1(N, dx), \Lambda_2(n, dx), \Lambda_3(n, dx), 0 \leq n < N]$ is the Lagrange multiplier (a string of positive measures on S) and the Lagrangian $L(V, \Lambda)$

is given by

$$
\begin{aligned}
L(V,\lambda) \quad & \overset{\text{def}}{=} \quad V_0(x_0) + \int \Lambda_1(N,dx)(g(x) - V_N(x)) + \sum_{n=0}^{N-1} \int \Lambda_2(n,dx)(g(x) \\
& -V_n(x)) + \sum_{n=0}^{N-1} \int \Lambda_3(n,dx)(\alpha \int p(dy|x)V_{n+1}(y) - V_n(x)). \quad (5)
\end{aligned}
$$

Now we shall describe a *gradient scheme* for estimating $V_n(x)$ using linear function approximations for $\{V_n\}$ and the square-roots of the Lagrange multipliers. For this, first suppose that $\Lambda_i(n,dx) = \lambda_i(n,x)m(dx), 1 \leq i \leq 3$, for some probability measure $m(dx)$ on S with full support [4]. In particular, $\lambda_i(n,\cdot)$'s are nonnegative. Approximate $V_n(x)$ and $\sqrt{\lambda_1}$, $\sqrt{\lambda_2}$ and $\sqrt{\lambda_3}$ as follows. Let $r \in \mathcal{R}^t$, $q_1 \in \mathcal{R}^{s_1}$, $q_2 \in \mathcal{R}^{s_2}$ and $q_3 \in \mathcal{R}^{s_3}$, with

$$
\begin{aligned}
V_n(x) \quad & \approx \quad \sum_{k'=1}^{t} r(k')\phi_{k'}(n,x), \\
\sqrt{\lambda_1}(n,x) \quad & \approx \quad \sum_{j=1}^{s_1} q_1(j)\varphi_{1j}(n,x), \\
\sqrt{\lambda_2}(n,x) \quad & \approx \quad \sum_{j=1}^{s_2} q_2(j)\varphi_{2j}(n,x), \\
\sqrt{\lambda_3}(n,x) \quad & \approx \quad \sum_{j=1}^{s_3} q_3(j)\varphi_{3j}(n,x),
\end{aligned}
$$

where $\{\phi_k, \varphi_{ij}\}$ are basis functions or 'features' selected a priori. Squaring the last three expressions above gives an approximation to $\lambda_1(n,x)$, $\lambda_2(n,x)$ and $\lambda_3(n,x)$, ensuring their nonnegativity automatically.

Then the original Lagrangian (**??**) is approximated in terms of r, q_1, q_2, q_3 by

$$
\begin{aligned}
L(r,q) \quad & = \quad \sum_{k'} r(k')\phi_{k'}(0,x_0) + \\
& \int \Big[(\sum_{l} q_1(l)\varphi_{1l}(N,x))^2 (g(x) - \sum_{k'} r(k')\phi_{k'}(N,x)) +
\end{aligned}
$$

[4] This itself can be the first step of the approximation procedure if the $\Lambda(i,dx)$ are not absolutely continuous w.r.t. $m(dx)$, the latter usually being some 'natural' candidate such as the normalized Lebesgue measure. For example, when $m(dx) =$ the normalized Lebesgue measure, convolution of $\Lambda_i(n,dx)$ with a smooth approximation of Dirac measure would give the desired approximation.

$$\sum_{n=0}^{N-1}(\sum_l q_2(l)\varphi_{2l}(n,x))^2(g(x)-\sum_{k'}r(k')\phi_{k'}(n,x)) +$$
$$\sum_{n=0}^{N-1}(\sum_l q_3(l)\varphi_{3l}(n,x))^2(\alpha\int p(dy|x)(\sum_{k'}r(k')\phi_{k'}(n+1,y)) -$$
$$\sum_{k'}r(k')\phi_{k'}(n,x))\Big]m(dx). \qquad (6)$$

Consider the following 'primal-dual' recursive scheme for solving this problem:

$$\begin{aligned} r_{m+1}(j) \quad = \quad & r_m(j) - a_m(\phi_j(0,x_0) + \int\Big[(\sum_l q_{1,m}(l)\varphi_{1l}(N,x))^2(-\phi_j(N,x)) + \\ & \sum_{n=0}^{N-1}(\sum_l q_{2,m}(l)\varphi_{2l}(n,x))^2(-\phi_j(n,x)) + \sum_{n=0}^{N-1}(\sum_l q_{3,m}(l)\varphi_{3l}(n,x))^2 \\ & (\alpha\int p(dy|x)\phi_j(n+1,y) - \phi_j(n,x)) + \eta r_m(j)\Big]m(dx)) \qquad (7) \end{aligned}$$

$$\begin{aligned} q_{1,m+1}(j) \quad = \quad & q_{1,m}(j) + b_m\int\Big[(2\varphi_{1j}(N,x)(\sum_l q_{1,m}(l)\varphi_{1l}(N,x)) \\ & (g(x) - \sum_{k'}r_m(k')\phi_{k'}(N,x)))\Big]m(dx) \qquad (8) \end{aligned}$$

$$\begin{aligned} q_{2,m+1}(j) \quad = \quad & q_{2,m}(j) + b_m\int\Big[(\sum_{n=0}^{N-1}2\varphi_{2j}(n,x)(\sum_l q_{2,m}(l)\varphi_{2l}(n,x)) \\ & (g(x) - \sum_{k'}r_m(k')\phi_{k'}(n,x)))\Big]m(dx) \qquad (9) \end{aligned}$$

$$\begin{aligned} q_{3,m+1}(j) \quad = \quad & q_{3,m}(j) + b_m\int\Big[(\sum_{n=0}^{N-1}2\varphi_{3j}(n,x)(\sum_l q_{3,m}(l)\varphi_{3l}(n,x))(\alpha\int p(dy|x) \\ & (\sum_{k'}r_m(k')\phi_{k'}(n+1,y)) - \sum_{k'}r_m(k')\phi_{k'}(n,x)))\Big]m(dx) \qquad (10) \end{aligned}$$

Here:

- (**??**) - (**??**) is the 'steepest ascent' for the weights for Lagrange multiplier approximation.

- (**??**) is the 'steepest descent' for the weights for the value function approximation. There is a small tweak to pure steepest descent, viz., the correction term $-\eta r_m(j)$ tagged at the end. This amounts to adding a quadratic correction term of $\frac{1}{2}\eta \sum_j r_m(j)^2$ to the originally *linear* objective function for the descent, thereby making is strictly convex. For the recursion, the additional term '$-\eta r_m(j)$' on the right amounts to an additional exponential averaging or 'smoothing'. If a very small positive quadratic term is added to the cost function of a linear program, the new optimum will still be one of the optima of the original linear case. That is, it won't affect the *minimizer* of the linear program (which is now a *quadratic* program). At the same time, it renders the objective function strictly convex which is empirically observed to make a big difference in the convergence behavior of our learning scheme.

- $\{b_m\}, \{a_m\}$ are stepsizes which we shall qualify later.

The approximation of $\{V_n\}$ is tantamount to restricting the domain of the problem to a prescribed subspace. The new feasible set will then be the intersection of the original feasible set with this subspace. Thus boundedness of the feasible set and convexity of the objective and constraint functions carry over. The further approximation of Lagrange multipliers is an additional feature here. This is best interpreted in terms of the Lagrangian above. We are not approximating the Lagrange multipliers of the original problem directly, but are instead seeking approximate Lagrange multipliers for the already approximated problem mentioned above. Specifically, the Lagrangian, which is convex-concave in the respective factor spaces of its product domain over the 'feasible set' = product of the feasible set for the primal variables $\times$ the domain of Lagrange multipliers, is now being restricted to a product of subspaces thereof. This retains convexity-concavity properties over the new feasible set given by the intersection of the 'feasible set' above with the product of subspaces in question, hence also ensures the existence of a saddle point in it.

Next we describe *stochastic approximation versions* for the above *gradient algorithms* for q and r. To do so, we first replace all conditional averages on the right-hand side with an actual evaluation at a simulated transition and then replace the full iterate by an incremental move in the desired (steepest ascent/descent) direction, weighted by a stochastic approximation-type stepsize. That is, we assume that $\{a_m, b_m\}_{m\geq 0}$ now satisfy the *usual* conditions:

$$a_m, b_m > 0, \quad \sum_m a_m = \sum_m b_m = \infty, \quad \sum_m (a_m^2 + b_m^2) < \infty. \tag{11}$$

In addition, we require that

$$\frac{b_n}{a_n} \rightarrow 0,$$

which ensures that the Lagrange multiplier iterations operate on a slower time scale than the value function iterations [**?**].

We assume that we have access to simulated pairs of $S-$valued random variables $(X_m, X'_{m+1}), m \geq 0$, available to us, where $\{X_m\}$ are i.i.d. with law $m(\cdot)$ on S, and X'_{m+1} is generated given X_m with the law $p(\ \cdot\ |X_n)$, conditionally independent of all other random variables realized till n. These may be generated, e.g., by a simulation device that generates them according to a model of such transitions, or be sampled from sorted empirical data.Then the *stochastic approximation version* of equations (**??**), (**??**), (**??**), (**??**) becomes

$$\begin{aligned} r_{m+1}(j) &= r_m(j) - a_m(\phi_j(0, i_0) + (\sum_l q_{1,m}(l)\varphi_{1l}(N, X_m))^2(-\phi_j(N, X_m)) + \\ &\sum_{n=0}^{N-1}(\sum_l q_{2,m}(l)\varphi_{2l}(n, X_m))^2(-\phi_j(n, X_m)) + \\ &\sum_{n=0}^{N-1}(\sum_l q_{3,m}(l)\varphi_{3l}(n, X_m))^2(\alpha\phi_j(n+1, X'_{m+1}) - \phi_j(n, X_m)) \\ &+\eta r_m(j)) \end{aligned} \tag{12}$$

$$\begin{aligned} q_{1,m+1}(j) &= \Gamma(q_{1,m}(j) + b_m(2\varphi_{1j}(N, X_m)(\sum_l q_{1,m}(l)\varphi_{1l}(N, X_m)) \\ &\quad (g(X_m) - \sum_{k'} r_m(k')\phi_{k'}(N, X_m)))) \end{aligned} \tag{13}$$

$$q_{2,m+1}(j) = \Gamma(q_{2,m}(j) + b_m(\sum_{n=0}^{N-1} 2\varphi_{2j}(n, X_m)(\sum_l q_{2,m}(l)\varphi_{2l}(n, X_m))$$

$$(g(X_m) - \sum_{k'} r_m(k')\phi_{k'}(n, X_m)))) \tag{14}$$

$$\begin{aligned} q_{3,m+1}(j) \quad = \quad & \Gamma(q_{3,m}(j) + b_m(\sum_{n=0}^{N-1} 2\varphi_{3j}(n, X_m)(\sum_{l} q_{3,m}(l)\varphi_{3l}(n, X_m)) \\ & (\alpha \sum_{k'} r_m(k')\phi_{k'}(n+1, X'_{m+1}) - \sum_{k'} r_m(k')\phi_{k'}(n, X_m)))) \end{aligned} \tag{15}$$

Γ above is the projection operator that projects the r.h.s. of each of (**??**) - (**??**) to an interval $[-M, M]$ for a prescribed $M >> 1$. This keeps the iterates bounded *a priori*. We *assume* that M is chosen large enough that the exact saddle point for the approximate Lagrangian is within the interior of the range $\mathcal{O}$ of Γ. Such projection operations are a common device in stochastic approximation when *a priori* boundedness of iterates is not guaranteed.

3 Convergence analysis (sketch)

Here we sketch a convergence proof for the proposed algorithm, partial because the further approximation of Lagrange multipliers introduces some additional complications. This is adapted from [**?**] with one major difference - in [**?**] the state space is finite, whereas here it is a general compact metric space, which leads to some additional technicalities. We shall be brief as the details would be quite tedious, albeit routine.

The algorithm is a special case of the *two time-scale stochastic approximation* algorithm [**?**], i.e., the $(d+s)-$dimensional recursion

$$x_{n+1} \quad = \quad x_n + a(n)[h(x_n, y_n) + M_{n+1}], \tag{16}$$

$$y_{n+1} \quad = \quad y_n + b(n)[w(x_n, y_n) + M'_{n+1}], \tag{17}$$

where $h : \mathcal{R}^{d+s} \to \mathcal{R}^d, w : \mathcal{R}^{d+s} \to \mathcal{R}^s$ are locally Lipschitz and $\{M_n\}, \{M'_n\}$ are *martingale difference sequences*, i.e.,

$$\begin{aligned} E[M_{n+1}|X_m, Y_m, M_m, M'_m, m \leq n] \quad &= \quad 0, \ n \geq 0, \\ E[M'_{n+1}|X_m, Y_m, M_m, M'_m, m \leq n] \quad &= \quad 0, \ n \geq 0, \end{aligned}$$

satisfying:

$$E[\|M_{n+1}\|^2 | X_m, M_m, Y_m, m \leq n] \leq K(1 + \|x_n\|^2 + \|y_n\|^2),$$
$$E[\|M'_{n+1}\|^2 | X_m, M_m, Y_m, m \leq n] \leq K(1 + \|x_n\|^2 + \|y_n\|^2),$$

for $n \geq 0$ and some $K > 0$. Suppose that the iterates remain bounded, i.e.,

$$P(\sup_n \|(x_n, y_n)\| < \infty) = 1, \tag{18}$$

and the following holds:

1. the ordinary differential equations (o.d.e.s)

$$\dot{x}(t) = h(x(t), y), \ \ y \in \mathcal{R}^s, \tag{19}$$

   are well-posed and have a unique globally asymptotically stable equilibrium $\gamma(y)$ for each y with $\gamma(\cdot)$ continuous, and,

2. the o.d.e.

$$\dot{y}(t) = w(\gamma(y(t)), y(t)), \tag{20}$$

   is well-posed and has a unique globally asymptotically stable equilibrium y^*.

Then

$$P((x_n, y_n) \rightarrow (\gamma(y^*), y^*)) = 1.$$

See, e.g., [**?**] or [**?**], Chapter 6.

To apply this to our case, direct verification shows that we have $x \approx r, y \approx q = [q_1, q_2, q_3]$, $h(r,q) = -\nabla_r \tilde{L}(r,q)$, $w(x,y) = \nabla_q \tilde{L}(r,q)$, where ∇_r, ∇_q denote resp. the gradients in r and q variables, and $\tilde{L} : \mathcal{R}^{t+s} \rightarrow \mathcal{R}^{t+s}, s = s_1 + s_2 + s_3$, defined by $\tilde{L}(r,q) = L(r,q) + \frac{1}{2}\eta\|r\|^2$, is a map that is strictly convex in the first argument for each fixed value of the second and strictly concave in the second argument for each fixed, *feasible* (i.e., satisfying the constraints) value of the first.

We start by assuming for the time being that the iterates remain bounded a.s., thus ensuring (**??**) in this context. Observe that (**??**) is for us a gradient descent scheme

$$\dot{r}(t) = -\nabla_r \tilde{L}(r(t), q) \tag{21}$$

for each q for the strictly convex $\tilde{L}(\cdot, q)$ and thus has a unique globally asymptotically stable equilibrium $\gamma(q) = \text{argmin}(\tilde{L}(\cdot, q))$.

In turn, one can invoke the envelope theorem of mathematical economics ([?], pp. 964-966) to conclude that (??) is a gradient ascent

$$\dot{q}(t) = \nabla \min_r \tilde{L}(r, q(t)). \tag{22}$$

We claim that this has a unique globally asymptotically stable equilibrium

$$q^* = \text{argmax}(\min_r L(r, \cdot)).$$

More generally, being a gradient ascent scheme on a compact set (recall that we are confining the iterates to a closed bounded set by projection), it will converge to a local maximum $\hat{q}$ of $\min_r L(r, \cdot)$. (We ignore the unstable equilibria as they can be ruled out by standard 'avoidance of traps' arguments from stochastic approximation as in Chapter 4 of [?].) Let $\hat{r}$ be the corresponding minimizer of $L(\cdot, \hat{q})$, which exists since this is a quadratic convex function once $\hat{q}$ is fixed. Note that we do not have feasibility of $\hat{r}$ *a priori*. It turns out that for the algorithm as proposed, this cannot be established directly. It is, however, *generically* true if q were the exact Lagrange multiplier. To see this, consider the abstract convex programming problem in $\mathcal{R}^d$ of minimizing $f(x)$ subject to $g_i(x) \leq 0, 1 \leq i \leq K$. The primal-dual o.d.e.s that are counterparts of the above would be

$$\begin{aligned} \dot{x}(t) &= -\nabla f(x) - \sum_i \lambda_i \nabla g_i(x), & (23) \\ \dot{\lambda}_i(t) &= g_i(x(t)) \; \forall \; i. & (24) \end{aligned}$$

Suppose $x^*, \lambda_i^*, 1 \leq i \leq K$, is a limit point for $(x(\cdot), \lambda(\cdot))$ such that x^* is not feasible for, say, only the i-th constraint (the situation where more than one constraint is violated can be handled similarly). Suppose the critical points of g_i and $f + \sum_i \lambda_j g_j$ do not overlap, which is generically true. Then by (??), $g_i(x^*) > 0$ and $\lambda_i(t)$ grows without bound. Since we are projecting the iterates to a very large set, it would settle to a very high value. But then the r.h.s. of (??) is dominated by $-\lambda_i \nabla^x g_i(x^*)$, whence x^* cannot be an equilibrium. The contradiction shows that $x(t)$ will be pushed towards the feasible set. The argument continues to apply if we use λ^2 in place of λ.

Unfortunately this logic fails in the present case because we further approximate the Lagrange multipliers, an inevitable step for the continuous state space we work with, and desirable even otherwise if the state space is finite but very large. In the present case for example, the o.d.e.s satisfied by the q's have the counterpart of the r.h.s. of (**??**) averaged with respect to a measure, see, e.g., (**??**) – (**??**). Thus we can expect the constraints to be satisfied only in an average sense. Nevertheless, if the approximation error remains small, then Theorem 1 of [**?**] implies that the approximating o.d.e. will closely track the original and will converge to a small neighborhood of what the original would converge to. This will mean that we do have convergence to a small neighborhood of the feasible set.[5] If $f + \sum_j \lambda_j^2 g_j$ is strictly concave in x and convex in λ on the feasible set as is the case here, it will remain so on a small neighborhood of it as well and thus the o.d.e.s and therefore the stochastic approximation scheme will converge ('a.s.' in the latter case) to a neighborhood of the unique saddle point by the 'two time scale' argument of [**?**], Chapter 6.

The following example due to an astute referee shows that things can indeed go wrong if the approximation is not good. Consider the linear program: Minimize x subject to $x \geq 1, x \leq 2$. The Lagrangian is $x + \lambda_1(-x + 1) + \lambda_2(x - 2)$. If we approximate $[\lambda_1, \lambda_2]$ by a single basis function $\alpha[1, 1]$, then both λ_1 and λ_2 are approximated by the same number q' and the Lagrangian becomes $x - q'$, whose minimum for fixed q' is $-\infty$. Thus how good the scheme will remain after the Lagrange multipliers are approximated will depend on how good the approximation is. At the bottom of this is the issue of how close to the original the limit of an ascent / descent scheme remains when restricted to a subspace, an issue of larger interest in approximation theory for optimization. See, e.g., [**?**] which suggests random projections as a theoretically justifiable possibility.

Assuming the approximation is good enough, we can hope for a feasible solution. On feasible set, our $L(r, \cdot)$ is strictly convex and the minimizer is therefore unique and global. The convergence analysis of the o.d.e. would then follow as in [**?**].

[5] Note, however, that we are glossing over some technicalities here – (**??**) – (**??**) for us are infinite dimensional.

Returning to the verification of (**??**), note that the a.s. boundedness of (**??**)-(**??**) is free, since we are projecting the r.h.s. to a bounded set. This, however, can introduce a boundary correction in the aforementioned o.d.e. But this correction term is zero if the driving vector field of the o.d.e. is transversal to the boundary of the set to which the iterates are projected and points inwards [**?**]. In view of the discussion of the preceding paragraph, the relevant vector field for us is the negative gradient of a convex function (cf. (**??**)), and such transversality can be safely assumed (i.e., will be generically true). Given the a priori a.s. boundedness of (**??**)-(**??**), that of (**??**) follows by a straightforward application of the criterion of [**?**] (see also [**?**], Chapter 3) as follows. This criterion requires that we consider the scaled limit

$$\tilde{h}(r,q) = \lim_{a\uparrow\infty} \frac{-\nabla_r \tilde{L}(ar,q)}{a},$$

and the associated limiting o.d.e.

$$\dot{r}(t) = \tilde{h}(r(t),q).$$

This is the same as (**??**) with $g(\cdot)$ set identically equal to zero in the definition of $\tilde{L}$, and will have the origin as its globally asymptotically stable equilibrium, uniformly in $q \in [-M,M]^s$. Hence the arguments of [**?**] can be mimicked to prove a.s. boundedness of $\{r_n\}$.

We summarize our conclusions as:

Theorem The iterates $\{r_n\}, \{(q_{1,n}, q_{2,n}, q_{3,n})\}$ converge a.s. to a small neighborhood of the saddle point of $\tilde{L}$.

Remarks:

1. We have chosen $\{X_n\}$ to be i.i.d. with law $m(\cdot)$. More generally, i.i.d. with law π (say), or even stationary Markov with marginal π would do as long as π has full support. The only difference will be that the o.d.e.s above will be suitably modified. For example, the limiting o.d.e. for (**??**) would then be

$$\begin{aligned}\dot{q}_{1t}(j) &= 2\int \pi(dx)\varphi_{1j}(N,x)(\sum_{\ell} q_{1t}(\ell)\varphi_{1\ell}(N,x))(g(x) \\ &\quad -\sum_k r_t(k)\int p(dy|x,u)\phi_k(N,y)).\end{aligned}$$

and not

$$\begin{aligned}\dot{q}_{1t}(j) &= 2\int m(dx)\varphi_{1j}(N,x)(\sum_{\ell} q_{1t}(\ell)\varphi_{1\ell}(N,i))(g(i) \\ &\quad -\sum_{k} r_t(k)\sum_{i'} p(i'|i,u)\phi_k(N,i')),\end{aligned}$$

Similar comments apply to the limiting o.d.e.s for (**??**), (**??**), (**??**). In case π is absolutely continuous w.r.t. m with Radon-Nikodym derivative $\kappa(\cdot)$, this amounts to using basis functions $\{\sqrt{\kappa}(\cdot)\varphi_{1j}(\cdot,\cdot)\}$ in place of $\{\varphi_{1j}(\cdot,\cdot)\}$, and so on. Naturally, the weights will correspondingly adjust themselves. This does not affect the analysis above. If π is not absolutely continuous w.r.t. m, it will have to be first approximated by a measure that is so (see, e.g., the footnote above) and then once again the same argument applies, albeit with an additional layer of approximation.

2. One important and common special case of the above is when $\{X_n\}$ is a single simulation run of a Markov chain with $X'_{n+1} = X_{n+1}\ \forall n$. This brings us to another important issue, viz., that of *split* sampling that we have employed. We underscore a simple observation which applies *across the board* to simulation based schemes that depend on *conditional* averaging rather than averaging (in particular, reinforcement learning based schemes, see, e.g., [**?**]). The key issue is the conditional averaging of X_{n+1} given X_n at time n with the correct conditional law. The further averaging with respect to the law of X_n (π above) is secondary as long as it meets some basic requirements, such as 'all states are being sampled comparably often'. In fact, as we observed above, its effect can be absorbed into the choice of basis functions. This suggests that one can replace the pair (X_n, X_{n+1}) with a pair (X_n, X'_{n+1}) so that: (i) the law of X_n is a law ξ of our choice, (ii) the conditional law of X'_{n+1} given X_n is $p(dy|X_n)$, and, (iii) $\{X_n, n \geq 0\}$ are i.i.d. / Markov. The analysis of the preceding section then continues to apply with π replaced by ξ. This separates or 'splits' the two issues of the (core) conditional averaging and the (secondary) averaging with respect to the stationary law of the 'current state'. In turn, it suggests the possibility of judiciously choosing the $\{X_n\}, \xi$ above to speed up convergence. This is an appealing possibility when the desired regions of the state space are not being explored fast enough as, e.g., in rare

event simulation. In fact, in typical applications in queuing networks or finance, we work with a Markov chain which only makes local, 'nearest neighbor' moves. In addition, the state space can split into multiple 'quasi-invariant' sets with the property that transitions between them are rare. Therefore the chain may scan a large state space rather slowly unless it is 'rapidly mixing' in a precise sense. Traditionally importance sampling methods have been used to circumvent this difficulty [**?**]. For schemes like ours where it is the conditional averaging that holds the key, one may simply 'rig' the $\{X_n\}$ to obtain the desired speed-up without resorting to importance sampling with its computational overheads. The 'best' choice of $\{X_n\}$ then becomes another issue to ponder about. With the simple choice of i.i.d. $\{X_n\}$ and $\xi = m$ that we took, there were already significant computational advantages seen in our numerical experiments. Note that now two randomizations are needed per iterate as opposed to one in the single simulation run. This essentially doubles the simulation budget. Nevertheless, this disadvantage is more than compensated for by other benefits of split sampling, notably the rapidity with which it scans the entire state space, obviating thereby the need for importance sampling.

The foregoing, however, assumes that this is an off-line scheme. In an on-line version, one perforce uses a single 'simulation' run.

3. We make here some generic comments regarding approximation of LPs by linear function approximation. Consider the LP: Maximize $\mathbf{c}^T\mathbf{x}$ on a polytope F of feasible solutions. The linear function approximation amounts to restricting the search for optimum to $F \cap H$ where H is the linear span of the basis functions. Let $\mathbf{x}^*$ denote the optimum for the original problem. Then the error in optimal reward due to approximation will be clearly bounded by $\|\mathbf{c}\|$ times the minimum distance between $\mathbf{x}^*$ and $F \cap H$. Consider a two dimensional F given by $[-a, a] \times [-b, b], b << a$ with $\mathbf{c} = [1, 1]$. Then the error is much smaller for $H = (-\infty, \infty) \times \{0\}$ than for $H = \{0\} \times (-\infty, \infty)$. This highlights the importance of the choice of basis functions. See [**?**], [**?**] for a more detailed error analysis of such approximations, albeit for a specific (viz., discounted) cost criterion.

As already mentioned, we have the additional burden of a further approximation of the Lagrange multiplier. Let λ, λ', x^* denote resp. the correct Lagrange multiplier, the approximate Lagrange multiplier, and the optimal solution, all for the approximate problem with function approximation. Then simple algebra shows that the error in the optimum is bounded by $K\|x^*\|(\|\lambda\| \vee \|\lambda'\|)\|\lambda - \lambda'\|$, for some constant K. In conjunction with an a priori bounds on the Lagrange multiplier such as the one on p. 516, [?], this gives an error estimate for the Lagrange multiplier approximation.

This error analysis is, however, very crude and a finer analysis similar to [?], [?] for the present case would be welcome, though difficult. In a companion work, some error analysis has been provided in a related context when the basis functions are chosen using random projection [?].

4 An example from option pricing

In this section we describe a simple numerical example to price a one dimensional American put, adapted from [?] to illustrate our scheme. As in [?], we assume that the risk-neutral stock price process follows the stochastic differential equation:

$$dS_t = rS_t dt + \sigma S_t dZ_t,$$

where r and σ are constants, $Z.$ is a standard Brownian motion and the stock does not pay dividends. The expiration time for the option is denoted by T. Let X_n denote the discretization $S(n\Delta t), n \geq 0$, and $N = T/\Delta t$, the corresponding total number of discrete time points in $(0, T]$ under consideration. Then $(X_n : n \leq N)$ is a Markov process. Conditioned on $X_n = x$, let

$$X_{n+1} = \exp[\xi_{n+1}],\ \xi_{n+1} \approx N(\mu, \sigma^2),$$

where $\hat{\mu} = (r - \sigma^2/2)\Delta t + \log \text{x}$, $\hat{\sigma}^2 = \Delta t \sigma^2$, and $N(\hat{\mu}, \hat{\sigma}^2)$ (see, e.g., [?], p. 94). Let the strike price be K. The intrinsic value $g(X_n)$ is given by,

$$g(X_n) = \max(0, K - X_n).$$

We use the following values :

1. $r = 0.06$, $\sigma = 0.2$. The strike price is $K = 25$. The expiration time for the option is $T = 25$ and $\Delta t = 1$. The number of periods, N is 25.

2. The stock price is constrained to a price range of $[10, 40]$. The continuous state space is made discrete by aggregating states in an interval of 0.01. This leads to a state space S with cardinality $|S| = 3000$. The boundaries of the state space are *reflecting*, i.e., whenever the price x hits the boundaries, it moves in the opposite direction with the same probabilities as before but stays at the same position with a probability equal to that of making the boundary-crossing move.

3. The discount factor α is 0.99 with a correction factor η of 0.0005.

4. The iterates for the Lagrange multipliers are bounded in the interval [-1000,1000]. The rule used for projecting the iterates back into the bounding interval is described as: When the iterates $q1$, $q2$ or $q3$ exceed a bound, they are simply reset to the boundary value. In our experiments, we observed that the bounds were never hit.

5. 14 basis functions ϕ were chosen to approximate the value function and 8 basis functions, φ were chosen to approximate each of the Lagrange multipliers. In the absence of any intuition, the basis functions for the Lagrange multipliers were chosen to be sinusoids of varying frequencies having the general form:

$$\begin{aligned} \varphi_{[1,2,3]i}(n, x) &= A(i)(1 + \sin(B(i, x, n))) \\ &\quad \text{where } 1 \le i \le 8, \quad x \in S, \quad 0 \le n \le N. \end{aligned}$$

   The exact equations are:

$$\begin{aligned} \varphi_{[1,2,3]i}(n, x) &= A_i \left(1 + \sin\left(\frac{0.01i(n+2)x}{N} + 40i\right)\right), \\ A_1 &= 0.0125, \quad A_{k+1} \; = \; 0.95A_k, \;\; k \in [2, 8] \\ &\quad \text{where } 1 \le i \le 8, \quad x \in S, \quad 0 \le n \le N. \end{aligned}$$

For the value function, the basis functions are simply a collection of decreasing 'linear times exponential of polynomial' functions. The parameters in the equations were chosen empirically. Fig. (1) shows the basis functions for the value function for n=0.

Figure 1: Basis functions for V at $n = 0$

The equations have the following general form:

$$\begin{aligned} \phi_i(n,x) &= A(i,x)\exp(B(n,x)) + D \\ &\quad \text{where } 1 \leq i \leq 14, \quad 1 \leq x \leq 3000, \quad 0 \leq n \leq N. \end{aligned}$$

The exact equations are:

$$
\begin{aligned}
\phi_i(n,x) &= A_i\left(\frac{-0.2x}{2225}+1.3318\right)\times \\
& \quad \exp\left((0.01x-60)\left(x+\frac{(n+1)}{N}\right)*10^{-5}\right)+D, \\
A_1 &= 0.78, \quad A_{k+1} \;=\; 0.9A_k, \;\; k\in[2,14], \\
D &= 0.0001, \\
& \quad \text{where } 1\le i\le 14, \quad 1\le x\le 3000, \quad 0\le n\le N.
\end{aligned}
$$

6. Equations (**??**), (**??**), (**??**), (**??**), were iterated 80000 times each. The step sizes are initially set as

$$a_0 = 0.2, \qquad b_0 = 0.05.$$

They are kept constant for the first 40000 iterations, and are thereafter decremented every 5000 iterations as follows:

$$a_m \longrightarrow \frac{a_m}{1.02}, \qquad b_m \longrightarrow \frac{b_m}{1.02}.$$

In particular, we take the same rate of decrease for both $\{a_n\}$ and $\{b_n\}$, contrary to what the theory dictates, separating them only by a constant factor. The simulations show good results regardless, indicating the robustness of the scheme. Intuitively, a common rate of decrement can work when o.d.e. timescales are 'naturally separated', i.e., the o.d.e. for the fast iteration converges faster than the o.d.e. for the slower iteration. This, however, can in general be hard to verify a priori. When feasible, having a common rate of decrement has the advantage of simplicity and also avoids slowing down of the scheme implicit in the use of a slower time scale.

Table (1) contains the results of the algorithm for a few initial states (averaged over at least 10 runs of the algorithm for each initial price). The expected values are the exact solutions $V_0(x_0)$ of the DP equations in (**??**). Fig (2.a) is the plot of the value function evaluated at the initial state $x_0 = 35$. For this choice of initial state, the value function visibly oscillates about some

Initial Price (x_0)	**Expected Value**	**Observed Value**	**Error (%)**
10.05	14.95	16.287	8.945
11.11	13.89	13.72	-1.222
15.74	9.26	8.939	-3.472
17.18	7.82	7.861	0.522
20.89	5.645	5.749	1.855
22.26	5.174	5.262	1.714
24.19	4.670	4.854	3.930
25.15	4.475	4.651	3.947
25.61	4.392	4.530	3.162
26.28	4.282	4.443	3.783
28.62	3.979	4.108	3.253
30.12	3.835	3.937	2.659
33.8	3.583	3.606	0.643
37.98	3.371	3.373	0.058
39.69	3.283	3.305	0.660

Table 1: Experimental Results

Figure 2: Algorithm behavior for the initial state $x_0 = 35$

value. Fig. (2.b) is a plot of the running average of the value function in Fig. (2.a). The running average is computed as follows.

$$\begin{aligned} w_{i+1}(x_0) &= \frac{iw_i + J_{i+1}(x_0)}{i+1}, \quad \text{with} \\ w_0(x_0) &= J_0(x_0), \end{aligned}$$

where $J_k(x_0)$ is the approximation of the value function for the initial state x_0 during period $n = 0$ at iteration i.

We note that the running average *appears* to show convergence within 40000 iterations but actually is still moving very slowly (difficult to see due to the scale of the y-axis). We therefore use the longer run for true convergence and a better approximation. Fig. (2.c) – Fig. (2.f) show the plots of the r, q_1, q_2 and q_3 weights for the first feature (i.e at $j = 1$ in (**??**) – (**??**)).

The following approach was used to evaluate multiple x_0's using a single simulation run. Whenever a state was encountered for the first time, a set of weights was assigned for that state. Thus, at time t, if n distinct states had been visited prior to t, there would be n sets of weights, each having been initialized at different times. The iterate equations were then applied to each of these n sets at every time step. On a typical run, 500 initial states are typically encountered within the first 550 iterations. Although we do not do so here, this simultaneous evaluation of multiple states using the same simulation can be easily parallelized.

Our experiments also indicate a small variance in the algorithm's output from one simulation run to the next. We further reduce this by averaging the outputs of the algorithm taken over a number of runs. In our experiments, the algorithm was evaluated about 10 times on average for each initial state. A summary of the results of our extensive tests is given in Table (2).

Range of Error	**No. of states**	**Percentage**
Less than 5%	2733	91.1
Between 5 and 10%	165	5.5
Greater than 10%	1	0.03
Total	2899	96.63

Table 2: Performance Summary

The remaining 3.37% (101 states) were not encountered in our tests even once. We note that all of these states lie in regions of the state space that are approximated very accurately by the algorithm.

Figure 3: Average error variation over initial states

Fig. (3) shows a plot of the algorithms's percentage error over the entire state space. The high error at the lower boundary of the state space can be expected due to the truncation of the Gaussian. That the error improves with the size of the state is possibly a reflection of the fact that the value function is 'flatter', i.e., less sensitive to state variation, for larger states. This is specific to the present example.

The experiments were carried out on a machine with an Intel(R) Core(TM)2 2.13 GHz processor and 2 GB RAM. Evaluating the value of a single state using the above approach on our test machine took 29.2 seconds using Matlab^{TM} 7.0

5 Possible extensions

The broad approach above also extends to other Markov decision processes. We illustrate this by outlining the corresponding developments for the infinite horizon discounted cost. Note that for such problems, linear function approximation based schemes have been used *in a rigorously justified manner* essentially for the linear, policy evaluation problem, i.e., to estimate the value function for a fixed policy, and not for learning the optimal policy (barring few exceptions such as [**?**])[6]. Our scheme, on the contrary, directly handles the optimization issue.

We assume a controlled discrete time Markov chain on a state space S with d states and an infinite number of stages. At state i, the use of control u specifies a transition probability $p(i'|i,u)$ to the next state i'. The control u is constrained to take values in a prescribed finite subset $U(i)$ of a fixed finite set U. We aim to optimize the discounted cost starting from a state i_0. This is defined as

$$J_\pi(i) \overset{\text{def}}{=} E\left[\sum_{j=0}^{\infty} \beta^j k(i_j, \mu_j(i_j)) | i_0 = i\right] \tag{25}$$

under an *admissible* policy $\pi \overset{\text{def}}{=} \{\mu_0, \mu_1, \ldots\}$, where the functions $\mu_j, j = 0, 1, 2, \ldots,$ map states i_j into controls $u_j \in U(i_j)$ for all $i_j \in S$ and $k(\cdot,\cdot)$ is the one-stage cost function. (That such 'Markov' controls which depend on the current state do at least as well as more general controls depending on the entire past history and possibly extraneous randomization, is a well

[6] Schemes like [**?**] do learn the optimal policy, but the linear function approximation based component remains an essentially linear policy evaluation scheme on a separate timescale, with the (nonlinear) optimization scheme operating on a slower timescale. There have been uses of linear function approximation in a fully nonlinear framework as well, but adequate theoretical justification is lacking in these cases.

known fact from Markov decision theory [**?**].) $0 < \beta < 1$ is the discount factor. The optimal discounted cost is given by

$$J^*(i) \overset{\text{def}}{=} \min_{\pi} J_\pi(i), \ i = 1, 2, \ldots, d. \tag{26}$$

A cornerstone of Markov decision theory is the following well known result [**?**]:

Theorem There exists a unique vector $V \overset{\text{def}}{=} \{V(1), \ldots, V(d)\}$ satisfying the Bellman equation

$$V(i) = \min_{u \in U(i)} \left[k(i,u) + \beta \sum_{i'=1}^{d} p(i'|i,u)V(i') \right], \ i = 1, 2, \ldots, d. \tag{27}$$

Furthermore, the *stationary* policy $\mu_n \equiv \mu$ is optimal if and only if $\mu(i)$ attains the minimum on the right for each i.

Next we recall the linear programming formulation for the above problem:

Maximize $V(i_0)$ s.t.

$$V(i) - k(i,u) - \beta \sum_{i'=1}^{d} p(i'|i,u)V(i') \leq 0, \ (i,u) \in S \times U(i).$$

This is the '*dual*' linear program, the '*primal*' being in terms of the so called occupation measures [**?**]. By Lagrange multiplier theory, the above optimization problem becomes

$$\max_V \min_\lambda [L(V, \lambda)] \tag{28}$$

where $\lambda = [[\lambda(i,u)]]$ is the Lagrange multiplier and the Lagrangian $L(V, \lambda)$ is given by

$$L(V, \lambda) \overset{\text{def}}{=} V(i_0) - \sum_{i,u} \lambda(i,u)(V(i) - k(i,u) - \beta \sum_{i'=1}^{d-1} p(i'|i,u)V(i')), \tag{29}$$

where $\lambda(i,u) \geq 0$. Now we shall describe a *gradient scheme* for estimating the optimal $V(i_0)$ using linear function approximations for V and $\sqrt{\lambda}$. Let $r \in \mathcal{R}^m$ and $q \in \mathcal{R}^n$. Let $V(i) \approx \sum_{k'=1}^{m} r(k')\phi_{k'}(i)$, $\sqrt{\lambda}(i,u) \approx \sum_{j=1}^{n} q(j)\varphi_j(i,u)$, $1 \leq i < d$, $\forall \ u \in U(i)$. Squaring the latter gives an approximation to $\lambda(i,u)$, ensuring its nonnegativity automatically.

Then the original Lagrangian (**??**) is approximated in terms of r, q by

$$\begin{aligned} L(r,q) &= \sum_{k'=1}^{m} r(k')\phi_{k'}(i_0) - \sum_{i,u}((\sum_j q(j)\varphi_j(i,u))^2(\sum_{k'} r(k')(\phi_{k'}(i) - \\ &\quad \beta\sum_{i'} p(i'|i,u)\phi_{k'}(i')) - k(i,u))). \end{aligned} \tag{30}$$

This corresponds to the approximate linear program:

Maximize $\sum_k r(k)\phi_k(i_0)$ s.t.

$$\sum_k r(k)\phi_k(i) - k(i,u) - \beta\sum_{i'=1}^{d}\sum_k r(k)p(i'|i,u)\phi_k(i') \leq 0,\ (i,u) \in S \times U(i),$$

with $(\sum_j q(j)\varphi_j(i,u))^2$ playing the role of the Lagrange multipliers. The 'primal-dual' scheme we consider is:

$$\begin{aligned} q_{t+1}(j) &= q_t(j) + 2b_t\sum_{i,u}\varphi_j(i,u)(\sum_\ell q(\ell)\varphi_\ell(i,u))(\sum_{k'} r_t(k')(\phi_{k'}(i) \\ &\quad -\beta\sum_{i'} p(i'|i,u)\phi_{k'}(i')) - k(i,u)),\ 1 \leq j \leq n, \end{aligned} \tag{31}$$

where $\{b_t\}$ are stepsizes which we shall qualify later. This is the 'steepest descent' for the weights for Lagrange multiplier approximation. Also, for $1 \leq k' \leq m$, and $\eta > 0$ very small,

$$\begin{aligned} r_{t+1}(k') &= r_t(k') - a_t(-\phi_{k'}(i_0) + \sum_{i,u}(\sum_j q_t(j)\varphi_j(i,u))^2 \\ &\quad (\phi_{k'}(i) - \beta\sum_{i'} p(i'|i,u)\phi_{k'}(i')) + \eta r_t(k')), \end{aligned} \tag{32}$$

where $\{a_t\}$ are stepsizes which we shall qualify later. This is the 'steepest ascent' for the weights for the value function approximation with one modification: We have added the term '$-\eta r_t(k')$' on the right as before.

Next we describe stochastic approximation versions for the above gradient algorithms for q and r. To do so, we first replace all conditional averages on the right-hand side with an actual evaluation at a simulated transition and then replace the full iterate by an incremental move in the desired (steepest ascent/descent) direction, weighted by a stochastic approximation-type stepsize. That is, we assume that $\{a_t, b_t\}_{t\geq 0}$ now satisfy the *usual* conditions:

$$a_t, b_t > 0, \quad \sum_t a_t = \sum_t b_t = \infty, \quad \sum_t (a_t^2 + b_t^2) < \infty, \quad \frac{b_t}{a_t} \to 0. \tag{33}$$

The last condition above ensures that the iteration (**??**) operates on a slower time-scale than the iteration (**??**). As many applications of this class of problems are 'on-line' in nature, we state the on-line version of the scheme, as opposed to the split-sampling used above. Thus we have a simulation run $(X_t, Z_t) \in S \times U(X_t)$, $t \geq 0$, as the backdrop. Then the stochastic approximation version of equation (**??**) becomes

$$\begin{aligned} q_{t+1}(j) &= q_t(j) + 2b_t(\varphi_j(X_t, Z_t)(\sum_\ell q_t(\ell)\varphi_\ell(X_t, Z_t))(\sum_{k'} r_t(k')(\phi_{k'}(X_t) \\ &\quad -\beta\phi_{k'}(X_{t+1})) - k(X_t, Z_t))), \ 1 \leq j \leq n, \end{aligned} \tag{34}$$

and similarly, (**??**) becomes

$$\begin{aligned} r_{t+1}(k') &= r_t(k') + a_t(\phi_{k'}(i_0) - (\sum_j q_t(j)\varphi_j(X_t, Z_t))^2(\phi_{k'}(X_t) \\ &\quad -\beta\phi_{k'}(X_{t+1})) - \eta r_t(k')), \end{aligned} \tag{35}$$

$1 \leq k' \leq m$, for $t \geq 0$.

Convergence of this scheme requires that for each pair $(i \in S, u \in U(i))$, we have

$$\liminf_{n\uparrow\infty} \frac{1}{n} \sum_{m=0}^{n-1} I\{X_m = i, Z_m = u\} > 0 \text{ a.s.}$$

(See, e.g., [**?**], Section 7.4.) This means that the relative frequency with which each state-action pair is sampled be nonzero. If on-line optimization is sought, this then makes it essential that at each step, one assign a small non-zero probability even to the actions that are not candidates for optimal.

Convergence analysis can then be given exactly along the lines of the above. A similar scheme is possible for the average cost problem.

References

[1] T. K. I. AHAMED, V. S. BORKAR AND S. JUNEJA, "Adaptive importance sampling for Markov chains using stochastic approximation", *Operations Research* 54 (2006), 489-504.

[2] L. ANDERSEN AND M. BROADIE, "Primal-dual simulation algorithm for pricing multidimensional American options", *Management Science* 50 (2004), 1222-1234.

[3] K. BARMAN AND V. S. BORKAR, "A note on linear function approximation using random projections", *Systems and Control Letters* 57 (2008), 784-786.

[4] A. BENSOUSSAN, *Stochastic Control by Functional Analysis Methods*, North Holland, Amsterdam, 1982.

[5] D. P. BERTSEKAS, *Dynamic Programming and Optimal Control, Vol. I, 3rd ed.*, Athena Scientific, Belmont, Massachusetts, 2005.

[6] D. P. BERTSEKAS, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, Mass., 1999.

[7] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, Mass., 1996.

[8] N. BOLIA, P. GLASSERMAN AND S. JUNEJA, "Function-approximation-based importance sampling for pricing American options", *Proceedings of the 2004 Winter Simulation Conference*, IEEE Press (2004), 604-611.

[9] V. S. BORKAR, "Stochastic approximation with two time scales", *Systems and Control Letters* 29 (1997), 291-294.

[10] V. S. BORKAR, "An actor-critic algorithm for constrained Markov decision processes", *Systems and Control Letters* 54 (2005), 207-213.

[11] V. S. BORKAR, *Stochastic Approximation: A Dynamical Systems View*, Hindustan Publ. Co. New Delhi, India, and Cambridge Uni. Press, Cambridge, UK, 2008.

[12] V. S. BORKAR AND S. P. MEYN, “The O.D.E. method for convergence of stochastic approximation and reinforcement learning”, *SIAM J. Control and Optim.* 38 (2000), 447-469.

[13] M. J. CHO AND R. H. STOCKBRIDGE, “Linear programming formulation for optimal stopping problems”, *SIAM J. Control and Optim.* 40 (2002), 1965-1982.

[14] D. CHOI AND B. VAN ROY, “A generalized Kalman filter for fixed point approximation and efficient temporal difference learning”, *Discrete Event Dynamic Systems* 16 (2006), 207-239.

[15] D. P. DE FARIAS AND B. VAN ROY, “The linear programming approach to approximate dynamic programming”, *Operations Research* 51 (2003), 850-865.

[16] D. P. DE FARIAS AND B. VAN ROY, “ On constraint sampling in the linear programming approach to approximate dynamic programming”, *Mathematics of Operations Research* 29, (2004), 462-478.

[17] E. B. DYNKIN, “The optimum choice of the instant of stopping a Markov process”, *Dokl. Acad. Nauk SSSR* 150 (1963), 238-240 (in Russian; English translation in *Soviet Math. Dokl.* 4, 627-629).

[18] P. GLASSERMAN, *Monte Carlo Methods in Financial Engineering*, Springer Verlag, New York, 2003.

[19] M. B. HAUGH AND L. KOGAN, “Pricing American options: a duality approach”, *Operations Research* 52 (2004), 258-270.

[20] O. HERNÁNDEZ-LERMA AND J.-B. LASSERRE, *Discrete-Time Markov Control Processes*, Springer Verlag, New York, 1996.

[21] M. W. HIRSCH, “Convergent activation dynamics in continuous time networks”, *Neural Networks* 2 (1989), 331-349.

[22] V. R. KONDA AND J. N TSITSIKLIS, “On actor-critic algorithms”, *SIAM J. Control and Optim.* 42 (2003), 1143-1166.

[23] H. J. KUSHNER AND D. S. CLARK, *Stochastic Approximation for Constrained and Unconstrained Systems*, Springer Verlag, New York, 1978.

[24] D. G. LUENBERGER, *Optimization by Vector Space Methods*, John Wiley, New York, 1968.

[25] F. A. LONGSTAFF AND E. S. SCHWARTZ, “Valuing American options by simulation: a simple least-square approach”, *Review of Financial Studies*, 14 (2001), 113-147.

[26] A. MAS-COLELL, M. D. WHINSTON AND J. R. GREEN, *Microeconomic Theory*, Oxford Uni. Press, Oxford, UK, 1995.

[27] L. C. G. ROGERS, “Monte Carlo valuations of American options”, *Mathematical Finance* 12 (2002), 271-286.

[28] R. S. SUTTON AND A. G BARTO, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, Mass., 1998.

[29] C. SZEPESVARI AND W. D. SMART, “Interpolation-based Q-learning”, *Proc. of the 21st Intl. Conf. on Machine Learning*, Banff, Alberta, Canada (2004), 100-108.

[30] J. N. TSITSIKLIS AND B. VAN ROY, “Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives”, *IEEE Transactions on Automatic Control* 44 (1999), 1840-1851.

[31] J. N. TSITSIKLIS AND B. VAN ROY, “Regression methods for pricing complex American-style options”, IEEE Transactions on Neural Networks, Vol. 12, No. 4 (special issue on computational finance), 2001, pp. 694-703.

[32] H. YU AND D. P. BERTSEKAS, “A least squares Q-learning algorithm for optimal stopping problems”, Lab. for Information and Decision Systems Report 2731, MIT, February 2007.