Probability in the Engineering and Informational Sciences, **17**, 2003, 213–234. Printed in the U.S.A.

CONVERGENCE OF SIMULATION-BASED POLICY ITERATION

WILLIAM L. COOPER
Department of Mechanical Engineering
University of Minnesota
Minneapolis, MN 55455
E-mail: billcoop@me.umn.edu

SHANE G. HENDERSON
School of Operations Research and Industrial Engineering
Cornell University
Ithaca, NY 14853
E-mail: shane@orie.cornell.edu

MARK E. LEWIS
Department of Industrial and Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117
E-mail: melewis@engin.umich.edu

Simulation-based policy iteration (SBPI) is a modification of the policy iteration algorithm for computing optimal policies for Markov decision processes. At each iteration, rather than solving the average evaluation equations, SBPI employs simulation to estimate a solution to these equations. For recurrent average-reward Markov decision processes with finite state and action spaces, we provide easily verifiable conditions that ensure that simulation-based policy iteration almost-surely eventually never leaves the set of optimal decision rules. We analyze three simulation estimators for solutions to the average evaluation equations. Using our general results, we derive simple conditions on the simulation run lengths that guarantee the almost-sure convergence of the algorithm.

1. INTRODUCTION

The policy iteration (PI) algorithm is a method for computing optimal policies in Markov decision processes (MDPs). In essence, the algorithm consists of a *policy*

© 2003 Cambridge University Press 0269-9648/03 $16.00

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

evaluation step in which the value (the precise meaning of value depends on the choice of optimality criterion) of the current policy is computed, and a *policy improvement* step where, if possible, the current policy is improved upon. These two steps are repeated iteratively until some stopping requirements are met. In this article, we focus on the average-reward optimality criterion for recurrent MDPs with finite state and action spaces. For such MDPs, the PI algorithm is known to converge to an optimal policy in a finite number of iterations.

The evaluation step of policy iteration consists of solving a set of linear equations called the *average evaluation equations* (AEE) or *Poisson's equation*. Using the solution to the AEE, the improvement step then employs a one-step analysis to decide if the current policy can be improved. For MDPs with large state spaces, the linear systems that must be solved in the evaluation step can be prohibitively large, thereby rendering the PI algorithm impractical. This phenomenon is, of course, the well-known "curse of dimensionality," which causes severe difficulties for this (and all other) MDP solution procedures.

In this article, we analyze simulation-based policy iteration (SBPI), a modification of the policy iteration algorithm described in [2,3,5–7]. Rather than exactly solving the AEE in the policy evaluation step, SBPI estimates solutions of the AEE via simulation. It then uses these estimates as a proxy for the exact AEE solution in the policy improvement step. Note that this procedure does not require the solution of the large linear system. Provided that SBPI employs such "reasonable" estimators, the mean squared error of the estimates of the AEE solution will converge to zero as the run length of each simulation grows to infinity. Since "regular" PI converges to an optimal policy, one might then conjecture that SBPI should converge almost-surely to an optimal policy, so long as the run lengths grow to infinity. This is, in fact, not the case; we present a counterexample to this effect.

In light of this observation, it is natural to ask what conditions do ensure the almost-sure convergence of SBPI. Several earlier articles have given partial answers to this question. For instance, Cao [6] shows that if simulation estimates are "close enough" to a solution of the AEE, then SBPI will stop at an optimal policy. In addition, he notes that as the run lengths of the simulations grow to infinity, the simulation estimates converge with probability 1 to a solution of the AEE. However, he does not provide verifiable conditions on the run lengths that ensure that estimates are indeed accurate enough. As the above-mentioned example will show, allowing the run lengths to grow to infinity does not suffice for almost-sure convergence of the algorithm, although, typically, this *is* sufficient for convergence in probability of the algorithm; see Remark 4.4.

We present easily verifiable conditions that guarantee the almost-sure convergence of the simulation-based algorithm. Thus, the first contribution of this article is to close the gap in the literature left by the interesting work of Cao [5–7] and Cao and Chen [8]. Our second contribution is the presentation of explicit convergence conditions related to three specific estimators each based on probabilistic interpretations of the AEE. In particular, we describe how to estimate two possible solutions: the bias (one estimator) and the relative value function (two different estimators).

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

We apply our results to obtain *explicit* conditions on the run lengths (or the appropriate analog) of each estimator to guarantee almost-sure convergence of the algorithm.

Other methods similar to SBPI include the actor critic algorithm [1,11] and the modified policy iteration algorithm [16, Sect. 8.7.1]. In the prior, approximate solutions in both the policy evaluation and policy improvement steps are obtained using simulation and simple recursions, whereas in the latter approximate solutions in the policy evaluation step are obtained using *value iteration*. Like modified policy iteration, we obtain an approximate solution in the policy evaluation step and solve the policy improvement step exactly. Another approach to solving MDPs is through the use of Q-learning algorithms [3,20,21]. This approach is applicable when one does not have explicit knowledge of the transition matrix and may be viewed as a simulation-based variant of the value iteration method for solving MDPs.

The rest of the article is organized as follows. Section 2 gathers useful definitions and results from MDP theory, thereby setting the framework for the remainder of the article. Section 3 describes a simple example for which SBPI does not converge, even though run lengths grow to infinity. Section 4 provides a general result that guarantees the almost-sure convergence of SBPI. Section 5 shows how to apply the general result to several specific estimators to yield explicit conditions that ensure convergence. Section 6 provides a brief summary and conclusions.

2. MARKOV DECISION PROCESS THEORY

In this section, we discuss an average reward criterion Markov decision process and introduce the policy iteration algorithm. Our notation closely follows that of Puterman [16]. Assume that the state space, $\mathbb{X}$, and the action space, $\mathbb{A}$, are finite. That is to say, when in state $x \in \mathbb{X}$, there is a finite set of actions A_x ($\mathbb{A} = \cup_{x\in\mathbb{X}} A_x$) from which a decision-maker can choose. Once the action a is chosen, a reward $r(x,a) < \infty$ is accrued, the process moves to state y with probability $p(y|x,a)$, and the process continues.

A deterministic decision rule d is a map from $\mathbb{X}$ to $\mathbb{A}$ such that when in state x, the action $d(x)$ is used. A deterministic, Markovian policy ψ is a sequence of decision rules that describes what decisions will be made for every decision epoch. That is to say, ψ is of the form $\{d_0, d_1, d_2, \ldots\}$. We are interested in the class of stationary, deterministic policies that use the same decision rule for all decision epochs. A policy in this class is of the form $\{d, d, \ldots\}$ and is denoted d^∞. Let P_d be the one-step transition matrix whose xy element is $p(y|x,d(x))$, and let Ψ denote the set of all nonanticipatory policies. We are now ready to define the gain and the bias of a policy ψ. Let X_k be the state of the system at stage k and let d_k be the decision rule at stage k under a particular policy ψ. The k-stage expected reward of the policy ψ given that the initial state is x is given by

$$J_\psi^k(x) \equiv E_\psi^x\left[\sum_{n=0}^{k-1} r(X_n, d_n(X_n))\right], \tag{2.1}$$

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

where E_ψ^x denotes expectation with respect to the probability measure determined by the initial state x and the policy ψ.

The *long-run average reward* or *gain* of a policy ψ given that the system started in state x is given by

$$g_\psi(x) = \liminf_{k\to\infty} J_\psi^k(x)/k.$$

The *optimal expected average reward* is $g^*(x) = \sup_{\psi\in\Psi} g_\psi(x)$. A policy ψ^* is called *long-run average* or *gain optimal* if $g_{\psi^*}(x) = g^*(x)$ for all $x \in \mathbb{X}$. If the Markov chain generated by a stationary, deterministic policy ψ is aperiodic, the *bias* of ψ given that the system started in state x is defined to be

$$h_\psi(x) = \sum_{n=0}^{\infty} E_\psi^x[r(X_n, \psi(X_n)) - g_\psi(X_n)]. \tag{2.2}$$

A slightly different definition of bias is required for periodic chains; however, we will not require the added level of generality in this article. For MDPs with multiple gain-optimal policies, the bias provides a natural criterion for selecting among them. For more detailed studies of the bias (and bias optimality), see Lewis and Puterman [12,13].

Let the set of all deterministic decision rules be denoted D and the set of deterministic stationary policies be denoted D^∞. In the later sections, we focus on policies in D^∞. So, in the interest of notational simplicity, we will use the notation d to mean both the policy $d^\infty = \{d, d, \ldots\}$ and the decision rule d; the precise meaning should be clear from the context. We will use the notation d^∞ when extra clarity is needed.

Before proceeding, we need a bit more terminology. Given a real-valued function f defined on $\mathbb{X}$, a decision rule $d' \in D$ is an element of $\arg\max_{d\in D}\{r_d + P_d f\}$ if

$$d'(x) \in \arg\max_{a\in A_x}\left\{r(x,a) + \sum_{y\in\mathbb{X}} p(y|x,a) f(y)\right\} \qquad \text{for each } x \in \mathbb{X}.$$

Unless otherwise stated, for the remainder of the article we assume that the Markov decision processes under consideration are *recurrent*. That is to say, all stationary, deterministic policies induce Markov chains that have one recurrent class and no transient states. Thus, we have the following:

1. The gain of any stationary, deterministic policy $d^\infty \in D^\infty$ is a constant vector which we express as $g_d 1$.
2. If (g, h) satisfies

$$h = r_d - g1 + P_d h, \tag{2.3}$$

then $g = g_d$ and h is unique up to a constant. The set of equations (2.3) are referred to as the *average evaluation equations* (AEE).

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

3. The pair (g_d, h_d) is the only solution of (2.3) that also satisfies the additional condition $P_d^* h = 0$, where P_d^* represents the stationary distribution of the Markov chain generated by d^∞.
4. The unique solution to the AEE that also satisfies $h(\alpha) = 0$ for a particular state α is called the *relative value function* of d with reference state α.

Thus, for a fixed policy, if we can estimate the gain and either the bias or a relative value function, we have an estimate of the solution of the AEE.

It is well known that if (g, h) satisfies

$$h = \max_{d \in D} \{r_d - g1 + P_d h\}, \tag{2.4}$$

then $g = g^*$ and h is unique up to a constant. The set of equations (2.4) are referred to as the *average optimality equations* (AOE). Let D^* be the set of decision rules that achieve the maximum in (2.4); that is, if (g, h) is a solution to the AOE, then

$$D^* \equiv \arg\max_{d \in D} \{r_d + P_d h\}. \tag{2.5}$$

We refer to (2.5) as the *average optimality selection equations*. Theorem 8.4.2 of Puterman [16] guarantees that $D^* \neq \emptyset$, and for any $d^* \in D^*$, the stationary policy $(d^*)^\infty$ is gain optimal.

We conclude this section with a brief discussion of the policy iteration algorithm and the simulation-based policy iteration algorithm. The PI algorithm was originally discussed by Howard [10]. The reader is also referred to Puterman [16, Chap. 8]. The SBPI algorithm was suggested by Bertsekas [2] and discussed further by Cao [5–7]. However, we emphasize that neither of the aforementioned authors provide verifiable convergence conditions for the algorithm.

Policy iteration algorithm

- **Step (i). Initialization.** Select a decision rule $d_0 \in D$; set $j = 0$.
- **Step (ii). Policy Evaluation.** Obtain a solution (g_{d_j}, h_j) to the AEE for the decision rule d_j.
- **Step (iii). Policy Improvement.** Choose d_{j+1} that satisfies

$$d_{j+1}(x) \in \arg\max_{a \in A_x} \left\{ r(x,a) + \sum_{y \in \mathbb{X}} p(y|x,a) h_j(y) \right\} \qquad \text{for each } x \in \mathbb{X}, \tag{2.6}$$

  setting $d_{j+1}(x) = d_j(x)$ whenever possible.

- **Step (iv). Iteration.** If $d_{j+1} = d_j$, then stop. Otherwise, let $j = j + 1$ and return to Step (ii).

The key convergence results for the PI algorithm for recurrent MDPs with finite state and action spaces are collected in [16, Sect. 8.6] and [2, Sect. 4.3.2]. For such MDPs, at each iteration of the PI algorithm there is either a strict increase in the gain

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

or else the gain remains unchanged. If the gain remains unchanged, then we have an element of D^*, and hence a gain optimal policy. Since there are only finitely many stationary, deterministic policies when the state and action spaces are finite, the algorithm is guaranteed to terminate in finitely many steps with the desired average reward optimal policy. In Step (iii), the requirement that $d_{j+1}(x) = d_j(x)$ whenever possible is not necessary to guarantee that PI reaches an optimal decision rule for a recurrent MDP—but it does prevent "bouncing" among optimal decision rules and it ensures that the stopping rule is met (i.e., without the condition, PI will reach D^* and stay there, but it might cycle through multiple optima never stopping). However, the condition is required to ensure that PI reaches an optimal decision rule in more general models.

To specify the SBPI algorithm, we replace the second step of the PI algorithm with an estimate $(g_{d_j}^{n_j}, h_{d_j}^{n_j})$ of a solution to the AEE, where the sequences $\{n_j\}$ are prespecified parameters of the estimates. Note that in most cases, n_j will simply be the number of simulation runs used to obtain the estimate at iteration j, but we allow them to be more general. A second difference is that at the iteration step, we do not specify a stopping criterion. The SBPI algorithm follows.

Simulation-based policy iteration algorithm

- **Step (i). Initialization.** Choose a sequence $\{n_j : j \geq 0\}$ and a decision rule d_0. Let $j = 0$.
- **Step (ii). Policy Evaluation Approximation.** Obtain an estimate of the solution to the AEE $(g_{d_j}^{n_j}, h_{d_j}^{n_j})$ for the decision rule d_j.
- **Step (iii). Policy Improvement.** Using our current estimate, find a decision rule d_{j+1} that satisfies

$$d_{j+1}(x) \in \arg\max_{a \in A_x} \left\{ r(x,a) + \sum_{y \in \mathbb{X}} p(y|x,a) h_{d_j}^{n_j}(y) \right\} \qquad \text{for each } x \in \mathbb{X}, \tag{2.7}$$

  setting $d_{j+1}(x) = d_j(x)$ whenever possible.

- **Step (iv). Iteration.** Let $j = j + 1$ and return to Step (ii).

Since we are only estimating solutions to the AEE, the fact that a policy repeats does not guarantee optimality. One way to incorporate a stopping rule is to perform a step of standard policy iteration whenever we think that we possess an optimal decision rule. If, after this iteration, we have the same decision rule, then we have arrived at an optimal policy. Alternatively, we could stop if the SBPI algorithm repeats the same policy for "enough" consecutive iterations. We leave the analysis of stopping rules as a subject for future research.

Note also that in the policy evaluation approximation step, we estimate the average cost, yet in the policy improvement step, we do not require this estimate. Indeed, all that is required for the policy improvement step is $h_{d_j}^{n_j}$. In light of the fact that no stopping rule is specified, we feel that having a running gain estimate is an

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

"added bonus." Since not much extra work is required to obtain the estimate, it is included in the algorithm. Finally, the condition that we set $d_{j+1}(x) = d_j(x)$ whenever possible is not needed for the primary results below. We could allow ties to be broken in any arbitrary manner, so long as the tie-breaking mechanism does not depend on the past (or future) evolution of the algorithm.

3. A COUNTEREXAMPLE

In this section, we show by example that SBPI need not converge almost-surely to the set of optimal policies, even when the number of simulation replications at each iteration grows to infinity. In subsequent sections, we present conditions that do guarantee such convergence.

Consider the following simple example with three states, labeled 1, 2, and 3. We obtain a single-period reward of zero whenever we are in state 1, and a single-period reward of one whenever we are in state 2 or state 3. When in state 1, there are two possible actions: α and β; so, $A_1 = \{\alpha, \beta\}$. With action α, we jump to state 2 with probability p and jump to state 3 with probability $1 - p$. With action β, we jump to state 2 with probability $1 - p$ and we jump to state 3 with probability p. We assume that the action sets A_2 and A_3 are singletons (i.e., we have no decision when in states 2 and 3). When in state 2, we return to state 2 with probability p and jump to state 1 with probability $1 - p$. Similarly, when in state 3, we return to state 3 with probability $1 - p$ and jump to state 1 with probability p. Suppose that $p \in (\frac{1}{2}, 1)$.

For the above-described example, there are only two decision rules to consider. In a slight abuse of notation, we shall call these rules α and β, corresponding to the action chosen in state 1. A simple check shows that the gain of α is $g_\alpha = (2p^2 - 2p + 1)/(p^2 - p + 1)$ and that the gain of β is $g_\beta = \frac{2}{3}$. It follows that the optimal decision rule is α [this would hold for any $p \in (0,1)$].

It is easy to show that a solution to the AEE is given by (g_d, v_d), where

$$v_d(x) = E_d^x \sum_{n=0}^{\tau_1 - 1} [r(X_n, d(X_n)) - g_d(X_n)],$$

and $\tau_1 = \inf\{n \geq 0 : X_n = 1\}$. Computing this function for each of our policies gives us

$$v_\alpha = \begin{pmatrix} 0 \\ \dfrac{p}{p^2 - p + 1} \\ \dfrac{1 - p}{p^2 - p + 1} \end{pmatrix} \quad \text{and} \quad v_\beta = \begin{pmatrix} 0 \\ \dfrac{1}{3(1 - p)} \\ \dfrac{1}{3p} \end{pmatrix}. \tag{3.1}$$

Since $v_\alpha(1) = v_\beta(1) = 0$, v_d is a relative value of d with reference state 1. Direct computations show that the policy improvement step of deterministic policy itera-

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

tion selects α when the "current" policy is either α or β (i.e., $\arg\max\{r_d + P_d v_\alpha\} = \arg\max\{r_d + P_d v_\beta\} = \alpha$).

Next, we describe a particular implementation of SBPI that employs simulation estimates of v_α and v_β. To make things as simple as possible, we will use the exact values of g_α and g_β in constructing our estimates. (We could also estimate the gain using the methods described in Section 5.2. If we did this, we could ensure that the estimate was close enough to the actual gain that the effects to be described would still prevail. Since this would merely make things more complicated, but not add any additional insight, we shall simply use the actual gain for this example.) Since we know $v_d(1) = 0$ without doing any computations, we need not estimate its value at state 1. Note also that $v_\alpha(1)$ and $v_\beta(1)$ are both multiplied by zero in the policy improvement step. Hence, we need only estimate the values of $v_d(2)$ and $v_d(3)$.

To generate a simulation estimate for both $v_\alpha(2)$ and $v_\alpha(3)$, we employ a single draw from the canonical space of infinite sequences of uniform-(0,1) random variables. Denote a generic element from this space by $\mathbf{u} = \{u_n : n = 0,1,2,\ldots\}$. We construct the estimates $\hat{v}_\alpha(2)$ and $\hat{v}_\alpha(3)$ by using the entries of $\mathbf{u}$ to generate transitions of the Markov chain induced by policy α.

Let $T : \mathbb{X} \times (0,1) \to \mathbb{X}$ be the transition function that maps the current state of the Markov chain and a single uniform-(0,1) to the next state of the Markov chain. We take

$$T(2,u) = \begin{cases} 2 & \text{for } u \in (0,p) \\ 1 & \text{for } u \in [\,p,1), \end{cases}$$

$$T(3,u) = \begin{cases} 1 & \text{for } u \in (0,p) \\ 3 & \text{for } u \in [\,p,1). \end{cases}$$

We will not need to generate transitions from state 1, so we do not discuss this. For $x = 2,3$, let $\{X_n^x\}$ denote the chain started in state x that uses $\mathbf{u}$ and T to make state transitions. It is crucial to note that $\{X_n^2\}$ and $\{X_n^3\}$ evolve according to the *same* $\mathbf{u}$. Our simulation estimators are

$$\hat{v}_\alpha(x) = \sum_{n=0}^{\tau_1^x - 1} (r(X_n^x) - g_\alpha) = \sum_{n=0}^{\tau_1^x - 1} \frac{p - p^2}{p^2 - p + 1} \qquad \text{for } x = 2,3, \tag{3.2}$$

where τ_1^x is the hitting time of state 1 for the chain $\{X_n^x\}$. It is easy to check that for $x = 2,3$, $\hat{v}_\alpha(x)$ is an unbiased estimator for $v_\alpha(x)$. In light of our joint construction, we see that

$$P\left(\hat{v}_\alpha(2) = m\,\frac{p - p^2}{p^2 - p + 1}, \hat{v}_\alpha(3) = \frac{p - p^2}{p^2 - p + 1}\right) = p^{m-1}(1-p), \qquad m = 2,3,\ldots, \tag{3.3}$$

$$P\left(\hat{v}_\alpha(2) = \frac{p - p^2}{p^2 - p + 1}, \hat{v}_\alpha(3) = m\,\frac{p - p^2}{p^2 - p + 1}\right) = p(1-p)^{m-1}, \qquad m = 2,3,\ldots. \tag{3.4}$$

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

Note that by virtue of our construction, exactly one of the two versions immediately hits state 1. Using T, we can also generate analogous estimators for decision rule β. Mimicking the above development, we get

$$P\left(\hat{v}_\beta(2) = m\left(\frac{1}{3}\right), \hat{v}_\beta(3) = \frac{1}{3}\right) = p^{m-1}(1-p), \qquad m = 2,3,\ldots, \tag{3.5}$$

$$P\left(\hat{v}_\beta(2) = \frac{1}{3}, \hat{v}_\beta(3) = m\left(\frac{1}{3}\right)\right) = p(1-p)^{m-1}, \qquad m = 2,3,\ldots. \tag{3.6}$$

By (3.3)–(3.6), it follows that

$$P(\hat{v}_\alpha(3) > \hat{v}_\alpha(2)) = P(\hat{v}_\beta(3) > \hat{v}_\beta(2)) = 1 - p. \tag{3.7}$$

Now, consider a simulation-based policy improvement step based on the average $\hat{v}_d^n$ of n independent replications of $\hat{v}_d$. Elementary calculations show that if $\hat{v}_d^n(3) > \hat{v}_d^n(2)$, then simulation-based policy improvement will choose policy β (which is *not* the choice that would have been made by deterministic policy improvement). A sufficient condition for $\hat{v}_d^n(3) > \hat{v}_d^n(2)$ is that each individual replication satisfies $\hat{v}_d(3) > \hat{v}_d(2)$. Therefore by (3.7), when using n independent replicates to perform policy improvement, the probability of error is at least $\epsilon_n \equiv (1-p)^n$ and, consequently, the probability of making the correct choice is at most $1 - \epsilon_n$.

The above construction provides a concrete example that allows us to see explicitly how incorrect choices can be made in the policy improvement step. To see what ramifications this has for the possible convergence of SBPI, fix $\epsilon \in (0,1)$ and let $\xi(n) = \min\{i : (1-\epsilon_n)^i < \epsilon\}$. Suppose now that we implement SBPI using the above constructions with the number of replications at iteration j given by

$$n_j = n \quad \text{for } j \in [\theta(n-1)+1, \theta(n)],$$

where $\theta(n) = \sum_{i=1}^{n} \xi(i)$. Note that $\xi(n)$ is the number of successive times that n is used as the number of replications. By the definitions of $\xi(n)$ and ϵ_n, we see that $\xi(n) \to \infty$ as $n \to \infty$. Furthermore, $n_j \to \infty$ as $j \to \infty$, because $\xi(n) < \infty$ for each n. In summary, although n_j does tend to infinity, it does so rather slowly—too slowly, in fact, for SBPI to converge to the optimal policy. Indeed, $P(d_j = \alpha \text{ for all } j \in [\theta(n-1)+1, \theta(n)]) < \epsilon$. Consequently, for any k, we have $P(d_j = \alpha \text{ for all } j \geq k) \leq \lim_{m\to\infty} \epsilon^m = 0$. Therefore, SBPI fails to converge almost-surely to α. Note also that the specific construction of the above simulation estimators was chosen to simplify calculations and is not required for convergence to fail.

So, SBPI need not converge to an optimal policy, even when using unbiased estimates for the solution to the AEE and letting the number of replications grow to infinity. As we shall see in the coming sections, it is important to ensure that the number of replications grows "fast enough."

4. THE CONVERGENCE RESULT

In this section, we present our main convergence results concerning the SBPI algorithm. We will provide easily verifiable conditions under which the SBPI algorithm

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

is, with probability 1, absorbed into the set of optimal decision rules. As was shown in Section 3, it does not suffice to simply let the number of simulation replications grow to infinity.

Before we proceed, we need to briefly describe how SBPI generates simulation estimates. We assume that each estimator $h_d^n(x)$ is of the form $h_d^n(x) = \zeta^n(x, d, U^n)$, where U^n is a random element from a space $\mathbf{U^n}$ and $\zeta^n(x,\cdot,\cdot)$ is a deterministic function that maps elements of D and $\mathbf{U^n}$ to realizations of $h_d^n(x)$. Recall that $\{n_j\}$ is a prespecified sequence of parameters in the SBPI algorithm. The quantity n_j tells us to use function ζ^{n_j} and space $\mathbf{U^{n_j}}$ to generate the simulation estimates of a solution of the AEE for the decision rule d_j. Note that d_j is itself random, since it depends on the outcomes of previous simulations. In Theorem 4.2, we assume that $\{U^{n_j}\}$ are independent. This means that conditional on d_j, the estimates $h_{d_j}^{n_j}(x) = \zeta^{n_j}(x, d_j, U^{n_j})$ are independent of the prior evolution of SBPI up to iteration j. Consequently, d_{j+1} is also independent of the prior evolution of SBPI up to iteration j, conditional on d_j.

One of the important concepts for our general convergence results is the comparison of SBPI with standard deterministic policy iteration. For each decision rule $d \in D$, define

$$\phi(d) \equiv \arg\max_{d' \in D} \{r_{d'} + P_{d'} h_d\};$$

that is, $\phi(d)$ is the set of policies that could be selected in the (standard) policy iteration algorithm when the current policy is d.

A key quantity is the probability

$$q(n, d) = P\left(\arg\max_{d' \in D} \{r_{d'} + P_{d'} h_d^n\} \subseteq \phi(d)\right). \tag{4.1}$$

Expression (4.1) gives the probability that an optimization using h_d^n as a proxy for h_d yields solutions that are also obtained from an optimization using h_d. We begin with a well-known lemma; see, for example, [4, Thm. 1.9, p. 422] or [17, Lemma 2.9.1].

LEMMA 4.1: *If* $\sum_{j=0}^{\infty}[1 - y_j] < \infty$ *and* $0 \le y_j \le 1$ *for all* j, *then* $\lim_{k\to\infty} \Pi_{j\ge k} y_j = 1$.

We are now ready to give sufficient conditions for the SBPI algorithm to reach D^* and remain there indefinitely with probability 1. In Section 5, we will show how the conditions can be applied in various situations. The following is the main result of this section.

THEOREM 4.2: *Consider the SBPI algorithm and suppose the following:*

1. $\{U^{n_j} : j \ge 0\}$ *are independent.*
2. $\{q_j : j \ge 0\}$ *satisfy*

$$\sum_{j=0}^{\infty} [1 - q_j] < \infty, \tag{4.2}$$

where $q_j \equiv \min_{d \in D} q(n_j, d)$.

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

Then, there exists an a.s.-finite random variable L such that

$$P(d_j \in D^* \text{ for all } j \geq L) = 1. \tag{4.3}$$

PROOF: For an element ω of the underlying probability space Ω, define $L(\omega) = \min\{\ell \in \mathbb{Z}^+ : d_j \in D^* \text{ for all } j \geq \ell\}$, provided that $\{\ell \in \mathbb{Z}^+ : d_j \in D^* \text{ for all } j \geq \ell\} \neq \emptyset$. The result will follow upon showing that $P(\exists\, \ell \in \mathbb{Z}^+ : d_j \in D^* \text{ for all } j \geq \ell) = 1$. For each n, define the event $A_n = \{d_j \in D^* \text{ for all } j \geq n\}$. We see that $\{\exists\, \ell : d_j \in D^* \text{ for all } j \geq \ell\} = \bigcup_{n\geq 0} A_n$. Hence,

$$P(\exists\, \ell : d_j \in D^* \text{ for all } j \geq \ell) = P\left(\bigcup_{n\geq 0} A_n\right) = \lim_{n\to\infty} P(A_n). \tag{4.4}$$

Therefore, it suffices to show that $\lim_{n\to\infty} P(A_n) = 1$.

Let $\tilde{n}$ be the number of deterministic decision rules for the MDP. Since we assumed our problem to have finite state and action spaces, it follows that $\tilde{n} < \infty$. By virtue of [16, Thm. 8.4.4 and the proof of Prop. 8.6.1] or [2, Exc. 4.11, p. 214], we see that if the sequence of decision rules produced by the SBPI algorithm, $d_0, d_1, \ldots, d_{\tilde{n}}$, satisfy $d_j \in \phi(d_{j-1})$ for $j = 1, \ldots, \tilde{n}$, then $d_{\tilde{n}}$ is an optimal policy. Similarly, if $d \in D^*$, then $\phi(d) \subseteq D^*$. The key fact is that once deterministic policy iteration arrives at an average reward optimal policy, running further iterations cannot cause it to move to a suboptimal policy. Therefore, for $n \geq \tilde{n}$, we have that

$$A_n \supseteq \{d_j \in \phi(d_{j-1}) \text{ for all } j \geq n\} \supseteq \{d_j \in \phi(d_{j-1}) \text{ for all } j > n - \tilde{n}\},$$

and so

$$P(A_n) \geq P(d_j \in \phi(d_{j-1}) \text{ for all } j > n - \tilde{n}) \geq \prod_{j>n-\tilde{n}} q_{j-1}.$$

The second inequality follows from the independence of the sequence $\{U^{n_j}\}$. Assumption (4.2) and Lemma 4.1 together imply that $\lim_{n\to\infty} \Pi_{j>n-\tilde{n}}\, q_{j-1} = 1$. Thus, $P(A_n) \to 1$ as $n \to \infty$, which completes the proof. ■

Remark 4.3: The independence assumption on the sequence $\{U^{n_j}\}$ ensures that $\{d_j\}$ forms a (nonhomogeneous) Markov chain on D. This basic structure would be preserved if we also used $\{U^{n_j}\}$ to estimate the transition probabilities that are required for the policy improvement step. If we did this and redefined $q(n, d)$ to be simply the probability of getting a "correct" maximizer, then Theorem 4.2 could also be applied to problems with unknown transition probabilities. However, we shall not venture further down this path.

Remark 4.4: An investigation of the proof of Theorem 4.2 reveals that although allowing the run lengths to grow to infinity is not sufficient for almost-sure convergence of SBPI, it *is* sufficient for convergence in probability. To see why, note that

$$P(d_n \in D^*) \geq \prod_{j=n-\tilde{n}}^{n-1} q_j,$$

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

and this finite product converges to 1 as $n \to \infty$ if and only if $q_j \to 1$ as $j \to \infty$. As we will see in Section 5, the condition that $q_j \to 1$ as $j \to \infty$ is typically implied by ensuring that the simulation run lengths grow to infinity.

5. SIMULATION ESTIMATES FOR SOLUTIONS TO THE AVERAGE EVALUATION EQUATIONS

Theorem 4.2 gives conditions under which SBPI eventually hits, and never leaves, the set of optimal policies. Condition (4.2) is the key to that result, but it is not a natural condition to check in applications. In this section, we discuss three different estimators of solutions to the AEE, and in each case, we give easily verifiable conditions that, in turn, imply (4.2).

We first give some preliminary results that streamline the presentation. For a given policy d^∞, recall that the AEE have multiple solutions, all of which differ by an additive constant term. Let h_d be a solution to the AEE and let h_d^n be an estimator of h_d, where n is related to the simulation run length. The specific definition of h_d and h_d^n will vary. Let n_j denote the run length used to estimate h_{d_j} on the jth iteration of SBPI, where d_j^∞ is the policy at the jth iteration. For a function $v : \mathbb{X} \to \mathbb{R}$, define $\|v\| = \max_{x \in \mathbb{X}} |v(x)|$.

LEMMA 5.1: *There exists $\delta > 0$ such that if*

$$\sum_{j=0}^{\infty} \max_{d \in D} P(\|h_d^{n_j} - h_d\| > \delta) < \infty,$$

then (4.2) holds.

PROOF: To verify (4.2), observe that for a given policy d^∞, there exists $\delta_d > 0$ such that for any h that satisfies $\|h - h_d\| < \delta_d$,

$$\arg\max_{a \in A_x} \left\{ r(x,a) + \sum_{y \in \mathbb{X}} p(y|x,a) h(y) \right\} \subseteq \arg\max_{a \in A_x} \left\{ r(x,a) + \sum_{y \in \mathbb{X}} p(y|x,a) h_d(y) \right\}$$

$$\text{for all } x \in \mathbb{X}. \qquad \textbf{(5.1)}$$

If we take $\delta = \min_d \delta_d$ (recall that there are only a finite number of stationary deterministic policies, so that $\delta > 0$), then $q(n,d) \geq P(\|h_d^n - h_d\| \leq \delta)$. The result now follows. ■

We immediately obtain the following corollary. Let $\text{mse}(Y)$ denote the mean squared error $E(Y - \mu)^2$ of the estimator Y of μ.

COROLLARY 5.2: *Suppose that for all $x \in \mathbb{X}$ and $d^\infty \in D^\infty$, $\text{mse}(h_d^{n_j}(x)) = E(h_d^{n_j}(x) - h_d(x))^2 \leq c_d / n_j$ for some deterministic constant c_d. If*

$$\sum_{j=1}^{\infty} \frac{1}{n_j} < \infty, \qquad \textbf{(5.2)}$$

then (4.2) holds.

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

PROOF: Let δ be as in Lemma 5.1. By Boole's and Chebychev's inequalities,

$$\begin{aligned} P(\|h_d^{n_j} - h_d\| > \delta) &= P\left(\bigcup_{x\in\mathbb{X}} \{|h_d^{n_j}(x) - h_d(x)| > \delta\}\right) \\ &\leq \sum_{x\in\mathbb{X}} P(|h_d^{n_j}(x) - h_d(x)| > \delta) \\ &\leq \sum_{x\in\mathbb{X}} \frac{\mathrm{mse}(h_d^{n_j}(x))}{\delta^2} \\ &\leq \sum_{x\in\mathbb{X}} \frac{c_d}{n_j \delta^2}. \end{aligned} \tag{5.3}$$

Since D^∞ is finite, the result follows by Lemma 5.1. ■

We now apply these results to three different estimators of a solution to the AEE. In what follows, we let $X = \{X_i : i \geq 0\}$ denote the Markov chain induced by a policy $d^\infty \in D^\infty$. To maintain readability, we suppress the dependence of X on d.

5.1. An Unbiased Estimate of the Bias

Define h to be the *bias*,

$$h(x) = \sum_{i=0}^{\infty} E^x[r(X_i) - g],$$

where E^x denotes expectation with respect to the probability measure under which $X_0 = x$. As has previously been discussed, (g, h) solves the AEE. We can estimate h using simulation as follows.

Suppose that it is possible to obtain samples from the stationary distribution, π say, of the Markov chain $\{X_i : i \geq 0\}$. Let $Y = \{Y_i : i \geq 0\}$ denote a version of the chain with initial state Y_0 sampled from π, so that Y is stationary. Similarly, let $X^x = \{X_i^x : i \geq 0\}$ denote a version of the chain initiated in state x. We require that X^x and Y be constructed on a common probability space. Define the stopping time

$$\eta = \inf\{k \geq 0 : X_k^x = Y_k\} \tag{5.4}$$

to be the first time that the two sample paths meet. To ensure that η is finite, we will typically require, in addition to the recurrent assumption, that the chain be aperiodic. We constrain the joint construction so that for $k > \eta$, $X_k^x = Y_k$ (after the two processes meet, they "stick" together).

Now, let

$$H(x) = \sum_{k=0}^{\eta} [r(X_k^x) - r(Y_k)]$$

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

be the difference in rewards accrued between the two chains until they meet at time η. Under appropriate conditions, $H(x)$ is an unbiased estimator for the bias $h(x)$, as the following result shows.

PROPOSITION 5.3: *Suppose that d is a stationary deterministic policy and that $E\eta < \infty$, where E denotes expectation with respect to the probability measure on the space on which both X^x and Y are constructed. Then, $EH(x) = h(x)$.*

PROOF: We have that

$$EH(x) = E\sum_{k=0}^{\infty}[r(X_k^x) - r(Y_k)]I(\eta \geq k)$$

$$= \sum_{k=0}^{\infty} E[r(X_k^x) - r(Y_k)]I(\eta \geq k) \tag{5.5}$$

$$= \sum_{k=0}^{\infty} E[r(X_k^x) - r(Y_k)] - \sum_{k=0}^{\infty} E[r(X_k^x) - r(Y_k)]I(\eta < k), \tag{5.6}$$

where the interchange in (5.5) is justified because the rewards $r(\cdot)$ are bounded and $E\eta < \infty$. The first term in (5.6) is the bias as given by $h(x)$, since $Er(Y_k) = g$. The second term in (5.6) is zero, because $X_k^x = Y_k$ for $k > \eta$. ■

Proposition 5.3 gives conditions under which we can construct a random variable $H(x)$ whose expectation is $h(x)$. By repeating the above construction n independent times to yield $H^1(x)$, $H^2(x),\ldots,H^n(x)$, we can obtain an unbiased estimator $h^n(x)$ of $h(x)$ given by

$$h^n(x) = \frac{1}{n}\sum_{i=1}^{n} H^i(x).$$

A critical ingredient in constructing the estimator $h^n(x)$ is obtaining a sample from π to serve as the initial point in a sample path of Y. Such samples can be obtained through the use of "perfect sampling" for Markov chains, of which the most widely used method is currently "coupling from the past" [14,15]. Unlike the approach of gathering samples after simulating the chain for some predetermined "burn-in" time, the samples are *exactly* distributed according to π. Furthermore, by repeating the coupling-from-the-past construction several independent times, we can obtain independent samples from π. An accessible introduction to coupling from the past is given in [22].

It is well known that it is possible to construct X^x and Y so that their forward coupling time (5.4) has a finite moment generating function in a neighborhood of the origin (see, e.g., [18, p. 419] or [19, p. 45]). This implies, in particular, that

$$E[(\eta_d(x))^2] < \infty \qquad \forall x \in \mathbb{X}, \tag{5.7}$$

where we have explicitly included the dependence of the coupling time η on the policy d^∞ and the starting state x of $\{X_k^x\}$.

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

LEMMA 5.4: *Suppose that (5.7) holds for all $d^\infty \in D^\infty$. Then, there exists a constant $\kappa < \infty$ such that*

$$\sup_{x \in \mathbb{X}, d \in D} \operatorname{var}(H_d(x)) < \kappa.$$

PROOF: For any state $x \in \mathbb{X}$ and policy $d^\infty \in D^\infty$,

$$E[(H_d(x))^2] \leq E\left(\sum_{k=0}^{\eta_d(x)-1} 2\|r_d\| \right)^2 = 4\|r_d\|^2 E[(\eta_d(x))^2] < \infty. \tag{5.8}$$

The result now follows from the assumption that the state and action spaces are finite. ■

We can now apply Corollary 5.2 to obtain the following result. The proof is immediate from Lemma 5.4 and the unbiasedness of the bias estimator h^n, since $\operatorname{mse}(h^n(x)) = \operatorname{var} H(x)/n$ for all $x \in \mathbb{X}$.

PROPOSITION 5.5: *Suppose that (5.7) holds. If*

$$\sum_{j=0}^{\infty} \frac{1}{n_j} < \infty,$$

then (4.2) holds.

In concert with Theorem 4.2, Proposition 5.5 gives easily verified sufficient conditions for SBPI to converge almost surely to an optimal policy. The condition (5.2) basically states that the run lengths must grow "fast enough" that we eventually do not erroneously step out of the set of optimal policies. Note that this condition is *not* satisfied by taking n_j to be constant, no matter how large; that is, "fixed run lengths are not enough."

The assumption that the forward coupling time possesses a finite moment generating function in a neighborhood of the origin allows us to weaken condition (5.2). In particular, if for every x and d there exists $\theta_{x,d} > 0$ so that

$$Ee^{\theta H_d(x)} < \infty \qquad \text{for } |\theta| < \theta_{x,d}, \tag{5.9}$$

then a standard large deviations result (e.g., [9, Thm. 6, p. 281]), ensures that for any $\bar{\delta} > 0$, there exists $0 < \gamma_{x,d} < 1$ so that $P(|h_d^n(x) - h_d(x)| > \bar{\delta}) \leq 2\gamma_{x,d}^n$. Provided the state and action spaces are finite, we can modify the derivation of (5.3) to conclude that there exist constants $c > 0$ and $0 < \gamma < 1$ so that

$$P(\|h_d^n - h_d\| > \bar{\delta}) < c\gamma^n. \tag{5.10}$$

Combining this with Lemma 5.1, we obtain the following result.

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

PROPOSITION 5.6: *Suppose that (5.9) holds. If γ satisfies (5.10) and*

$$\sum_{j=0}^{\infty} \gamma^{n_j} < \infty, \tag{5.11}$$

then (4.2) holds.

It is unlikely that one will be able to identify γ satisfying (5.10) a priori, but observe that we could also employ the stronger condition that (5.11) holds with $\gamma = \nu$ for all $\nu \in (0,1)$, which is a condition that does not rely on a priori knowledge. For example, if $n_j = j$, then this condition holds, but (5.2) does not. In this case, stronger moment assumptions do indeed yield a "payoff."

5.2. An Estimate of the Relative Value Function

The estimator of the bias constructed in Section 5.1 relies on obtaining a sample from the stationary distribution of the chain. Perfect sampling may be employed for any recurrent, aperiodic, finite-state-space Markov chain to obtain such samples. However, the computational effort required to perform perfect sampling can grow exorbitantly as the state space increases in size. Motivated by this observation, we show how to construct a simulation estimator of a different solution to the AEE.

Fix $x^* \in \mathbb{X}$ and let $\tau = \inf\{n \geq 0 : X_n = x^*\}$ denote the hitting time of x^*. A solution to the AEE is then given by the relative value function

$$h(x) = E^x \sum_{i=0}^{\tau-1} [r(X_i) - g].$$

(Again, we are suppressing dependence on the policy d in our notation. Note, also, that we are *redefining* h within this subsection.) We can construct an estimator of $h(x)$ as follows.

The gain g can be estimated via

$$g^n = \frac{1}{n} \sum_{i=0}^{n-1} r(X_i),$$

with an arbitrary initial distribution on the chain. Once this point estimate is obtained, for all $x \in \mathbb{X}$, $h(x)$ may be estimated by $h^{m,n}(x)$, where $h^{m,n}(x)$ is the sample mean of m conditionally independent (given g^n) replicates of

$$H^n(x) = \sum_{j=0}^{\tau-1} [r(X_j^x) - g^n].$$

Remark 5.7: The estimator $h^{m,n}$ of h is typically biased because of the presence of bias in the estimator g^n of g. If g^n is an unbiased estimator of g, then $h^{m,n}$ is an unbiased estimator of the relative value function h (see also [12,13]).

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

We then obtain the following result on convergence of SBPI. Let m_j and n_j represent the run lengths used to estimate $h = h_{d_j}$ on the jth iteration of SBPI. For the following result, we need not use the same "reference" state x^* at each iteration of SBPI.

PROPOSITION 5.8: *If*

$$\sum_{j=0}^{\infty} \max\left\{\frac{1}{n_j}, \frac{1}{m_j}\right\} < \infty$$

and the estimator $h^{m,n}$ of h outlined above is used in SBPI, then (4.2) holds.

PROOF: We will show that the mean squared error of $h^{m,n}(x)$ is of the order $\max\{m^{-1}, n^{-1}\}$ so that the result will follow immediately from Corollary 5.2. We append the suffix (i) to quantities to indicate that they relate to the ith independent replication, so that $\tau(i)$ is the time to hit x^* in the ith independent replication of H^n and so forth. We then have that

$$\begin{aligned}
\mathrm{mse}(h^{m,n}(x)) &= E[h^{m,n}(x) - h(x)]^2 \\
&= E\left[\frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j=0}^{\tau(i)-1}[r(X_j^x(i)) - g] + \sum_{j=0}^{\tau(i)-1}(g - g^n)\right) - h(x)\right]^2 \\
&= E\left[\frac{1}{m}\left(\sum_{i=1}^{m} D_i(x)\right) + (g - g^n)\bar{\tau}(m)\right]^2 \\
&\le 2E\left[\frac{1}{m}\sum_{i=1}^{m} D_i(x)\right]^2 + 2E[(g - g^n)^2\bar{\tau}(m)^2], \qquad \textbf{(5.12)}
\end{aligned}$$

where $D_i(x) = \sum_{j=0}^{\tau(i)-1}[r(X_j^x(i)) - g] - h(x)$, and $\bar{\tau}(m)$ is the sample mean of $\tau(i)$, $i = 1, \ldots, m$. Equation (5.12) follows because $(a + b)^2 \le 2a^2 + 2b^2$ for any real numbers a and b.

Now, because of the recurrent and finite-state-space assumptions, τ has a moment generating function and, hence, moments of all orders under E^x for any $x \in \mathbb{X}$. In particular, $E^x\tau^2 < \infty$ for all $x \in \mathbb{X}$. It then follows, as in Lemma 5.4, that $E^x D_1(x)^2 < \infty$ since r is bounded. Since $E^x D_1(x) = 0$, the first term in (5.12) equals $2E^x D_1(x)^2/m$.

Turning to the second term in (5.12), $\bar{\tau}(m)$ is independent of g^n, so that it is equal to $2\mathrm{mse}(g^n)E^x\bar{\tau}_m^2$. In addition,

$$E^x\bar{\tau}_m^2 = \frac{E^x\tau^2}{m} + \frac{m(m-1)(E^x\tau)^2}{m^2},$$

and $E^x\tau^2 < \infty$, so that $E^x\bar{\tau}_m^2$ is bounded. Therefore, the proof will be complete once we show that $\mathrm{mse}(g^n) \le c/n$ for some $c > 0$. Let μ denote the initial distribution on the chain when computing g^n. Theorem 6.5 of [4] gives that $\mathrm{var}_\mu(g^n) \le c'/n$ for some $c' > 0$ (for any initial distribution), so it remains to establish that the bias of

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

$g^n, E^\mu g^n - g \le c''/\sqrt{n}$, where E^μ denotes the expectation operator on the path space of the chain with initial distribution μ. Let

$$b(x) = \sum_{n=0}^{\infty} E^x[r(X_n) - g],$$

which is finite for all $x \in \mathbb{X}$, since $\mathbb{X}$ is finite and the MDP is recurrent. Now, observe that

$$\begin{aligned}
E^\mu g^n - g &= \sum_{x\in\mathbb{X}} \mu(\{x\}) E^x \frac{1}{n} \sum_{i=0}^{n-1} [r(X_i) - g] \\
&= \sum_{x\in\mathbb{X}} \mu(\{x\}) \frac{1}{n} \sum_{i=0}^{n-1} E^x[r(X_i) - g] \\
&= \sum_{x\in\mathbb{X}} \mu(\{x\}) \left[\frac{1}{n} \sum_{i=0}^{\infty} E^x[r(X_i) - g] - \frac{1}{n} \sum_{i=n}^{\infty} E^x[r(X_i) - g] \right] \\
&= \sum_{x\in\mathbb{X}} \mu(\{x\}) \left[\frac{1}{n} b(x) + o(n^{-1}) \right] \\
&= \frac{1}{n} \sum_{x\in\mathbb{X}} \mu(\{x\}) b(x) + o(n^{-1}),
\end{aligned}$$

where $o(a_n)$ denotes a sequence $\{w_n\}$ with the property that $w_n/a_n \to 0$ as $n \to \infty$. Thus, the bias in g^n is bounded by c''/n for some c'', and the proof is complete. ■

5.3. A Ratio Estimator

The estimator constructed in Section 5.2 relies on a preliminary simulation to estimate the gain g. One might prefer an estimator that can be obtained from a single simulation run, and in this subsection, we consider such an estimator.

Let $T(0) = 0$, and for $n \ge 0$, define $T(n+1) = \inf\{k > T(n) : X_k = x^*\}$ to be the consecutive hitting times of the distinguished state $x^* \in \mathbb{X}$. Suppose that $X_0 = x^*$, so that the times $T(0), T(1), \ldots$, divide the sample path of $X = \{X_k : k \ge 0\}$ into i.i.d. regenerative cycles.

Let

$$g^n = \frac{\sum_{i=0}^{T(n)-1} r(X_k)}{T(n)}$$

be an estimate of the gain based on n regenerative cycles. For $i \ge 1$, define

$$W_i(x) = \sum_{j=T(i-1)}^{T(i)-1} (r(X_j) - g^n)\, N_x(j, i)$$

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

and

$$C_i(x) = \sum_{j=T(i-1)}^{T(i)-1} I(X_j = x),$$

where

$$N_x(j,i) = \sum_{k=T(i-1)}^{j} I(X_k = x)$$

is the number of visits to state x by time j within the ith regenerative cycle. Here, $C_i(x)$ gives the number of visits to state x in the ith regenerative cycle. The expression for $W_i(x)$ will be discussed further.

Define the estimator $h^n(x)$ by

$$h^n(x) = \frac{\overline{W}_n(x)}{\bar{C}_n(x)}$$

if $\bar{C}_n(x) > 0$, and 1 otherwise, where $\overline{W}_n(x)$ and $\bar{C}_n(x)$ denote sample means of $W_1(x),\ldots,W_n(x)$ and $C_1(x),\ldots,C_n(x)$, respectively. In contrast to previous sections where n represented the simulation run length in terms of the number of transitions, n represents the number of completed regenerative cycles here.

Here, $h^n(x)$ is used as an estimator of $h(x)$, where

$$h(x) = E^x \sum_{k=0}^{\tau-1} [r(X_k) - g]$$

is the expected cumulative cost (centered by the gain g) until hitting the state x^* starting from state x. (Recall that $\tau = \inf\{k \geq 0 : X_k = x^*\}$ is the hitting time of state x^*.)

To see why h^n is a reasonable estimator of h, note that

$$\begin{aligned} W_i(x) &= \sum_{j=T(i-1)}^{T(i)-1} (r(X_j) - g^n) \sum_{k=T(i-1)}^{j} I(X_k = x) \\ &= \sum_{k=T(i-1)}^{T(i)-1} I(X_k = x) \sum_{j=k}^{T(i)-1} (r(X_j) - g^n), \end{aligned}$$

so that $W_i(x)$ is a sum of terms, each starting in state x and representing the cumulative centered cost until the end of the current regenerative cycle when the chain hits state x^*. Thus, $W_i(x)$ is a sum of (dependent) terms, each of which is a reasonable estimator of $h(x)$. The estimator given by Eq. (18) in [6] is similar to h^n as defined above, but it only counts the *first* time (if any) in each regenerative cycle that the chain visits state x. The following proposition establishes that h^n has mean squared error of order n^{-1}.

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

PROPOSITION 5.9: *Suppose that the chain $X = \{X_n : n \geq 0\}$ is irreducible. Then, the* $\text{mse}(h^n(x)) \leq cn^{-1}$ *for some* $c < \infty$. *Hence, if* n_j *denotes the number of regenerative cycles used in SBPI iteration* j, *then*

$$\sum_{j=0}^{\infty} \frac{1}{n_j} < \infty$$

implies that (4.2) holds.

PROOF: We only need to establish the mean squared error result, since an application of Corollary 5.2 then completes the proof.

Let us fix $x \in \mathbb{X}$ and then drop the dependence in our notation on x, so that $C_i = C_i(x)$. For $i \geq 1$, define $\eta_i = T(i) - T(i-1)$ to be the length of the ith regenerative cycle, and note that η_i represents the hitting time of state x^* in a finite-state-space irreducible Markov chain. Therefore, η_i has a finite moment generating function in a neighborhood of the origin, $i \geq 1$. Since $0 \leq C_i \leq \eta_i$, C_i also has a finite moment generating function in a neighborhood of the origin, $i \geq 1$. Furthermore, if we define the i.i.d. sequence $\{V_i : i \geq 1\}$ by

$$V_i = W_i + (g^n - g) \sum_{j=T(i-1)}^{T(i)-1} N_x(j,i) = \sum_{j=T(i-1)}^{T(i)-1} [r(X_j) - g] N_x(j,i),$$

then since $|V_i| \leq 2\|r\|\eta_i^2$, V_i also has a finite moment generating function in a neighborhood of the origin.

For notational convenience, define

$$D_i = \sum_{j=T(i-1)}^{T(i)-1} N_x(j,i).$$

Note that

$$nE(h^n - h)^2 = nE\left[\frac{\bar{W}_n}{\bar{C}_n} - \frac{EV_1}{EC_1}\right]^2 I(\bar{C}_n > 0) + n(1-h)^2 P(\bar{C}_n = 0). \qquad \textbf{(5.13)}$$

Now, the chain is irreducible and recurrent, so that all states are visited infinitely often. In particular, then, $EC_1 > 0$ and $P(C_1 = 0) < 1$. It follows that $n(1-h)^2 \times P(\bar{C}_n = 0) = n(1-h)^2 P(C_1 = 0)^n \to 0$ as $n \to \infty$ and is, therefore, $o(1)$. To establish the result, it remains to show that the first term in (5.13) is bounded.

The identity $(a+b+c)^2 \leq 3a^2 + 3b^2 + 3c^2$ allows us to conclude that

$$\begin{aligned}
nE(h^n - h)^2 &= nE\left[\frac{\bar{V}_n}{\bar{C}_n} + (g - g^n)\frac{\bar{D}_n}{\bar{C}_n} - \frac{EV_1}{EC_1}\right]^2 I(\bar{C}_n > 0) + o(1) \\
&= nE\left[\frac{\bar{V}_n - EV_1}{\bar{C}_n} + EV_1\left(\frac{1}{\bar{C}_n} - \frac{1}{EC_1}\right) + (g - g^n)\frac{\bar{D}_n}{\bar{C}_n}\right]^2 I(\bar{C}_n > 0) + o(1) \\
&\leq 3nE\left[\frac{\bar{V}_n - EV_1}{\bar{C}_n}\right]^2 I(\bar{C}_n > 0) + 3n(EV_1)^2 E\left[\frac{1}{\bar{C}_n} - \frac{1}{EC_1}\right]^2 I(\bar{C}_n > 0) \\
&\quad + 3nE\left[(g - g^n)\frac{\bar{D}_n}{\bar{C}_n}\right]^2 I(\bar{C}_n > 0) + o(1). \qquad \textbf{(5.14)}
\end{aligned}$$

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

The remainder of the proof consists of showing that each of the terms in (5.14) is bounded in n.

Note that

$$
\begin{aligned}
&3nE\left[\frac{\bar{V}_n - EV_1}{\bar{C}_n}\right]^2 I(\bar{C}_n > 0) \\
&\quad = 3nE\left[\frac{\bar{V}_n - EV_1}{\bar{C}_n}\right]^2 I(\bar{C}_n \in (0, EC_1/2]) + 3nE\left[\frac{\bar{V}_n - EV_1}{\bar{C}_n}\right]^2 I(\bar{C}_n > EC_1/2) \\
&\quad \le 3n^3 E[(\bar{V}_n - EV_1)^2 I(\bar{C}_n \in (0, EC_1/2])] + \frac{12n}{(EC_1)^2} E[(\bar{V}_n - EV_1)^2 I(\bar{C}_n > EC_1/2)] \\
&\quad \le 3n^3 E[(\bar{V}_n - EV_1)^4]^{1/2} P(\bar{C}_n \in (0, EC_1/2])^{1/2} + \frac{12n}{(EC_1)^2} E(\bar{V}_n - EV_1)^2 \\
&\quad = 3E[(\bar{V}_n - EV_1)^4]^{1/2} n^3 P(\bar{C}_n \in (0, EC_1/2])^{1/2} + \frac{12 \operatorname{var} V_1}{(EC_1)^2}.
\end{aligned}
$$

The first inequality follows since $\bar{C}_n \ge n^{-1}$ on the event $\bar{C}_n \in (0, EC_1/2]$. To see why, note that $\bar{C}_n > 0$ implies that $C_i > 0$ for at least one i, and C_i is integer-valued. The second inequality is a consequence of the Cauchy–Schwarz inequality. Finally, by direct calculation, we can show that $E[(\bar{V}_n - EV_1)^4]$ is bounded in n (in fact, it is of the order n^{-2}), and since $P(\bar{C}_n \in (0, EC_1/2])$ converges to 0 exponentially fast, $n^3 P(\bar{C}_n \in (0, EC_1/2])^{1/2}$ is also bounded in n. Thus, we have shown that the first term in (5.14) is bounded in n. A similar approach can be used to show that the remaining terms in (5.14) are also bounded in n, and so the proof is complete. ■

6. CONCLUSIONS

We have analyzed the convergence of simulation-based policy iteration for average reward Markov decision processes with finite state and action spaces. By way of an example, we have shown that allowing simulation run lengths to grow to infinity does not, in general, suffice to ensure the almost-sure convergence of SBPI. Arguing from first principles, we have derived sufficient conditions for SBPI to be absorbed into the set of optimal decision rules with probability 1. Subsequently, we demonstrated how these general conditions can be applied to three different simulation estimators in order to obtain simple, easily verified conditions that ensure the desired almost-sure convergence of SBPI.

One might very well ask which of the three estimators of solutions to the AEE given in Section 5 one should use in a given situation. The estimator in Section 5.3 is perhaps the least complex to implement, as it can be applied based on a single simulated sample path. The estimator in Section 5.1 is desirable in that it produces unbiased estimates of the bias. It has the disadvantage that it requires "perfect sampling capability." If one is unwilling, or unable, to implement such a capability, then the estimator in Section 5.2 might be considered.

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press

Acknowledgments

The work of the first, second, and third authors was supported in part by National Science Foundation grant numbers DMI-0115385, DMI-0085165, and DMI-9908321, respectively.

References

1. Barto, A.G., Sutton, R.S., & Anderson, C.W. (1983). Neuron-like elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man and Cybernetics* 13: 835–846.
2. Bertsekas, D.P. (1995). *Dynamic programming and optimal control, Vol. II*. Belmont, MA: Athena Scientific.
3. Bertsekas, D.P. & Tsitsiklis, J.N. (1996). *Neuro-dynamic programming*. Belmont, MA: Athena Scientific.
4. Brémaud, P. (1999). *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. New York: Springer-Verlag.
5. Cao, X.-R. (1998). The relations among potentials, perturbation analysis, and Markov decision processes. *Discrete-Event Dynamic Systems: Theory and Applications* 8: 71–87.
6. Cao, X.-R. (1999). Single sample path-based optimization of Markov chains. *Journal of Optimization Theory and Applications* 100: 527–548.
7. Cao, X.-R. (2000). A unified approach to Markov decision problems and performance sensitivity analysis. *Automatica* 36: 771–774.
8. Cao, X.-R. & Chen, H.-F. (1997). Perturbation realization, potentials, and sensitivity analysis of Markov processes. *IEEE Transactions on Automatic Control* 42(10): 1382–1393.
9. Fristedt, B. & Gray, L. (1997). *A modern approach to probability theory*. Boston: Birkhauser.
10. Howard, R.A. (1960). *Dynamic programming and Markov processes*. New York: Wiley.
11. Konda, V.R. & Borkar, V.S. (1999). Actor-critic-type learning algorithms for Markov decision processes. *SIAM Journal on Control and Optimization* 38: 94–123.
12. Lewis, M.E. & Puterman, M.L. (2001). A probabilistic analysis of bias optimality in unichain Markov decision processes. *IEEE Transactions on Automatic Control* 46(1): 96–100.
13. Lewis, M.E. & Puterman, M.L. (2002). Bias optimality. In E. Feinberg & A. Shwartz (eds.), *Handbook of Markov decision processes: Methods and applications*. Boston: Kluwer Academic.
14. Propp, J.G. & Wilson, D.B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms* 9: 223–252.
15. Propp, J.G. & Wilson, D.B. (1998). How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph. *Journal of Algorithms* 27: 170–217.
16. Puterman, M.L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. New York: Wiley.
17. Resnick, S. (1992). *Adventures in stochastic processes*. Boston: Birkhauser.
18. Ross, S.M. (1996). *Stochastic processes*, 2nd ed. New York: Wiley.
19. Thorisson, H. (2000). *Coupling, stationarity, and regeneration*. New York: Springer-Verlag.
20. Tsitsiklis, J. (1994). Asynchronous stochastic approximation and Q-learning. *Machine Learning* 16: 195–202.
21. Watkins, C.J.C.H. & Dayan, P. (1992). Q-Learning. *Machine Learning* 8: 279–292.
22. Wilson, D.B. (2000). *Layered multishift coupling for use in perfect sampling algorithms (with a primer on CFTP)*. Fields Institute Communications, Vol. 26. Providence, RI: American Mathematical Society.

https://doi.org/10.1017/S0269964803172051 Published online by Cambridge University Press