OPERATIONS RESEARCH
Vol. 53, No. 1, January–February 2005, pp. 126–139
ISSN 0030-364X | EISSN 1526-5463 | 05 | 5301 | 0126

informs®
DOI 10.1287/opre.1040.0145

An Adaptive Sampling Algorithm for Solving Markov Decision Processes

Hyeong Soo Chang
Department of Computer Science and Engineering, Sogang University, Seoul, Korea,
hschang@ccs.sogang.ac.kr

Michael C. Fu
Robert H. Smith School of Business, and Institute for Systems Research, University of Maryland,
College Park, Maryland 20742, mfu@rhsmith.umd.edu

Jiaqiao Hu, Steven I. Marcus
Department of Electrical and Computer Engineering, and Institute for Systems Research, University of Maryland,
College Park, Maryland 20742 {jqhu@glue.umd.edu, marcus@eng.umd.edu}

Based on recent results for multiarmed bandit problems, we propose an adaptive sampling algorithm that approximates the optimal value of a finite-horizon Markov decision process (MDP) with finite state and action spaces. The algorithm adaptively chooses which action to sample as the sampling process proceeds and generates an asymptotically unbiased estimator, whose bias is bounded by a quantity that converges to zero at rate $(\ln N)/N$, where N is the total number of samples that are used per state sampled in each stage. The worst-case running-time complexity of the algorithm is $O((|A|N)^H)$, independent of the size of the state space, where $|A|$ is the size of the action space and H is the horizon length. The algorithm can be used to create an approximate receding horizon control to solve infinite-horizon MDPs. To illustrate the algorithm, computational results are reported on simple examples from inventory control.

Subject classifications: dynamic programming/optimal control: Markov finite state.
Area of review: Stochastic Models.
History: Received June 2002; revision received May 2003; accepted November 2003.

1. Introduction

In this paper, we propose an "adaptive" sampling algorithm that approximates the optimal value to break the well-known *curse of dimensionality* in solving finite-horizon Markov decision processes (MDPs). The algorithm is aimed at solving MDPs with large state spaces and relatively smaller action spaces. The approximate value computed by the algorithm not only converges to the true optimal value but also does so in an "efficient" way. The algorithm adaptively chooses which action to sample as the sampling process proceeds, and the estimate produced by the algorithm is asymptotically unbiased, with the worst-case bias bounded by a quantity that converges to zero at rate[1] of $O(\sum_{i=0}^{H-1}(\ln N_i)/N_i)$, where H is the length of the horizon and N_i is the total number of samples that are used per state sampled in stage i. The logarithmic bound in the numerator is achievable uniformly over time. Given that the action space size is $|A|$, the worst-case running-time complexity of the algorithm is $O((|A| \max_{i=0,\ldots,H-1} N_i)^H)$, which is independent of the state space size but is dependent on the size of the action space, due to the requirement that each action be sampled at least once at each sampled state.

The idea behind the adaptive sampling algorithm is based on the expected *regret* analysis of the multiarmed bandit problem developed by Lai and Robbins (1985). In particular, we exploit the recent finite-time analysis work by Auer et al. (2002) that elaborated Agrawal (1995). The goal of the multiarmed bandit problem is to play as often as possible the machine that yields the highest (expected) reward. The regret quantifies the exploration/exploitation dilemma in the search for the true "optimal" machine, which is unknown in advance. During the search process, we wish to explore the reward distribution of different machines while also frequently playing the machine that is empirically best thus far. The regret is the expected loss due to not always playing the true optimal machine. Lai and Robbins (1985) showed that for an optimal strategy the regret grows at least logarithmically in the number of machine plays, and recently Auer et al. (2002) showed that the logarithmic regret is also achievable uniformly over time with a simple and efficient sampling algorithm for arbitrary reward distributions with bounded support. We incorporate their results into a sampling-based process for finding an optimal action in a state for a single stage of an MDP by appropriately converting the definition of regret into the difference between the true optimal value and the approximate value yielded by the sampling process. We then extend the one-stage sampling process into multiple stages in a recursive manner, leading to a multistage (sampling-based) approximation algorithm for solving MDPs.

To the best of our knowledge, this is the first work applying the theory of the multiarmed bandit problem to derive a provably convergent algorithm for solving general finite-horizon MDPs. The closest related works are probably those of Agrawal et al. (1989) and Graves and Lai (1997). Agrawal et al. considered a controlled Markov chain problem with finite state and action spaces for infinite-horizon average reward, where transition probabilities and initial distribution are parameterized by an unknown parameter θ belonging to some known finite parameter space, and each Markov chain induced from each fixed parameter is irreducible and aperiodic. They assume that the unique optimal stationary policy is known for the (infinite-horizon) average reward under a recurrence condition for each θ. Although they consider a finite-horizon loss function defined over all θ based on the regret of Lai and Robbins (1985), they regard the optimal stationary policy for the average reward as an approximation for an optimal nonstationary policy that minimizes the loss for the finite horizon. By then using the optimal stationary policy for the average reward for each θ, they develop an adaptive but rather complex policy (see Lai and Robbins 1985, §III.B), the performance of which is bounded in terms of the horizon size of the loss function, which vanishes as the size increases. That is, the adaptive policy is "asymptotically efficient" and works well for all θ such that the loss associated with the adaptive policy is equal to the lower bound on the loss function asymptotically (as the scheme is applied over infinite number of time steps). The adaptiveness comes from the use of the multiarmed bandit theory for the stationary control laws. In other words, the arm corresponds to a particular stationary law or policy, but not a particular action in the action space. Graves and Lai (1997) deal with a more general case but use essentially the same framework.

The rest of this paper is organized as follows. In §2, we give the necessary background and an intuitive description of the adaptive sampling algorithm, present a formal description of the algorithm, and discuss how to create an (approximate) receding horizon control (Hernández-Lerma and Lasserre 1990) via the sampling algorithm to solve MDPs in an "on-line" manner in the context of "planning" for infinite-horizon criteria. In §3, we provide the proofs for the convergence and the convergence rate of the worst-case bias. In §4, we provide some computational results on examples from inventory control. Two additional estimators using the same adaptive sampling framework of the algorithm are proposed and compared numerically with the original estimator. In §5, we conclude this paper with some remarks.

2. Adaptive Sampling Algorithm

2.1. Background

Consider a finite-horizon MDP $M = (X, A, P, R)$ with finite state space X, finite action space A with $|A| > 1$, nonnegative bounded reward function R such that R: $X \times A \to \mathscr{R}^+$, and transition function P that maps a state and action pair to a probability distribution over X. We denote the probability of transitioning to state $y \in X$ when taking action a in state $x \in X$ by $P(x,a)(y)$. For simplicity, we assume that every action is admissible in every state.

Let Π be the set of all possible nonstationary Markovian policies $\pi = \{\pi_t | \pi_t\colon X \to A, t \geqslant 0\}$. Our goal is to estimate the optimal discounted total reward (thereby obtaining an approximate optimal policy) for horizon length H, discount factor γ, and initial state x_0. Defining the optimal reward-to-go value function for state x in stage i by

$$V_i^*(x) = \sup_{\pi\in\Pi} E\left[\sum_{t=i}^{H-1} \gamma^t R(x_t, \pi_t(x_t)) \,\middle|\, x_i = x\right],$$
$$x \in X, 0 < \gamma \leqslant 1,\ i = 0, \ldots, H-1,$$

with $V_H^*(x) = 0$ for all $x \in X$ and x_t a random variable denoting the state at time t following policy π, we wish to estimate $V_0^*(x_0)$. Throughout the paper, we assume that γ is fixed. It is well known (see, e.g., Bertsekas 1995) that V_i^* can be written recursively as follows: for all $x \in X$ and $i = 0, \ldots, H-1$,

$$V_i^*(x) = \max_{a\in A}(Q_i^*(x,a)),$$

where

$$Q_i^*(x,a) = R(x,a) + \gamma \sum_{y\in X} P(x,a)(y) V_{i+1}^*(y).$$

Suppose we estimate $Q_i^*(x,a)$ by a sample mean $\hat{Q}_i(x,a)$ for each action $a \in A$, where

$$\hat{Q}_i(x,a) = R(x,a) + \gamma \frac{1}{N_{a,i}^x} \sum_{y\in S_a^x} \hat{V}_{i+1}^{N_{i+1}}(y), \qquad (1)$$

where S_a^x is the *multiset* (which means the set may include repeated members, i.e., the same element more than once) of (independently) sampled next states according to the distribution $P(x,a)$, and $|S_a^x| = N_{a,i}^x \geqslant 1$ for all $x \in X$ and such that $\sum_{a\in A} N_{a,i}^x = N_i$ for a fixed $N_i \geqslant |A|$ for all $x \in X$, and $\hat{V}_{i+1}^{N_{i+1}}(y)$ is an estimate of the unknown $V_{i+1}^*(y)$. Note that the number of next-state samples depends on the state x, action a, and stage i. Suppose also that we estimate the optimal value of $V_i^*(x)$ by

$$\hat{V}_i^{N_i}(x) = \sum_{a\in A} \frac{N_{a,i}^x}{N_i} \hat{Q}_i(x,a).$$

This leads to the following recursion:

$$\hat{V}_i^{N_i}(x) := \sum_{a\in A} \frac{N_{a,i}^x}{N_i}\left(R(x,a) + \gamma \frac{1}{N_{a,i}^x} \sum_{y\in S_a^x} \hat{V}_{i+1}^{N_{i+1}}(y)\right),$$
$$i = 0, \ldots, H-1,$$

with $\hat{V}_H^{N_H}(x) = 0$ for all $x \in X$ and any $N_H > 0$.

In the above definition, the total number of sampled (next) states is $O(N^H)$ with $N = \max_{i=0,\dots,H-1} N_i$, which is independent of the state space size. One approach is to select "optimal" values of $N_{a,i}^{x'}$ for $i = 0, \dots, H-1$, $a \in A$, and $x' \in X$, such that the expected error between the values of $\hat{V}_0^{N_0}(x)$ and $V_0^*(x)$ is minimized, but this problem would be difficult to solve. So instead we seek the values of $N_{a,i}^{x'}$ for $i = 0, \dots, H-1$, $a \in A$, and $x' \in X$ such that the expected difference is *bounded* as a function of $N_{a,i}^{x'}$ and N_i, $i = 0, \dots, H-1$, and that the bound (from above and from below) goes to zero as N_i, $i = 0, \dots, H-1$, go to infinity. We propose an "adaptive" allocation rule (sampling algorithm) that adaptively chooses which action to sample, updates the value of $N_{a,i}^{x'}$ as the sampling process proceeds, and achieves convergence such that as $N_i \to \infty$ for all $i = 0, \dots, H-1$, $E[\hat{V}_0^{N_0}(x)] \to V_0^*(x)$, and is efficient in the sense that the worst possible bias is bounded by a quantity that converges to zero at rate $O(\sum_i (\ln N_i)/N_i)$, and the logarithmic bound in the numerator is achievable uniformly over time.

As mentioned before, the main idea behind the adaptive allocation rule is based on a simple interpretation of the regret analysis of the multiarmed bandit problem, a well-known model that captures the exploitation/exploration trade-off. An M-armed bandit problem is defined by random variables $K_{i,n}$ for $1 \leqslant i \leqslant M$ and $n \geqslant 1$, where successive plays of machine i yield "rewards" $K_{i,1}, K_{i,2}, \dots,$ which are independent and identically distributed according to an unknown but fixed distribution δ_i with unknown expectation μ_i. The rewards across machines are also independently generated. Let $T_i(n)$ be the number of times machine i has been played by an algorithm during the first n plays. Define the *expected regret* $\rho(n)$ of an algorithm after n plays by

$$\rho(n) = \mu^* n - \sum_{i=1}^{M} \mu_i E[T_i(n)] \quad \text{where } \mu^* := \max_i \mu_i.$$

Lai and Robbins (1985) characterized an "optimal" algorithm such that the best machine, which is associated with μ^*, is played exponentially more often than any other machine, at least asymptotically. That is, they showed that playing machines according to an (asymptotically) optimal algorithm leads to[1] $\rho(n) = \Theta(\ln n)$ as $n \to \infty$ under mild assumptions on the reward distributions. Unfortunately, obtaining an optimal algorithm (proposed by Lai and Robbins) can sometimes be very difficult, so Agrawal (1995) derived a set of simple algorithms that achieve the asymptotic logarithmic regret behavior, using a form of *upper confidence bounds*. During the plays, we are temped to take the machine with the maximum current sample mean—exploitation. But the sample mean $\hat{\mu}_i(\bar{n})$ for the machine i is just an estimate that contains uncertainty, where $\bar{n}$ is the number of overall plays so far. To account for this, we add a function $\sigma_i(\bar{n})$ such that $\hat{\mu}_i(\bar{n}) - \sigma_i(\bar{n}) \leqslant \mu_i < \hat{\mu}_i(\bar{n}) + \sigma_i(\bar{n})$ with high probability, where $\hat{\mu}_i(\bar{n}) + \sigma_i(\bar{n})$ is the upper confidence bound (see Agrawal 1995 for a substantial discussion). Then the width of the confidence bound gives us guidance for exploration. Indeed, the use of the upper confidence bound leads us to trade off between exploitation and exploration, giving a criterion of which of the two between exploitation and exploration to be selected. Agrawal's algorithm is to choose the machine with the highest upper confidence bound at each play over time. For bounded rewards, Auer et al. (2002) propose simple upper confidence-bound based algorithms that achieve the logarithmic regret uniformly over time, rather than only asymptotically, and our sampling algorithm primarily builds on their results.

For an intuitive description of the allocation rule, consider first only the one-stage approximation. That is, we assume for now that we know $V_1^*(x)$ for all $x \in X$. To estimate $V_0^*(x)$, obviously we need to estimate $Q_0^*(x, a^*)$, where $a^* \in \arg\max_{a \in A}(Q_0^*(x, a))$. The search for a^* corresponds to the search for the best machine in the multiarmed bandit problem. We start by sampling each possible action once at x, which leads to the next state according to $P(x, a)$ and reward $R(x, a)$. We then iterate as follows (see **Loop** in Figure 1). The next action to sample is the one that achieves the maximum among the current estimates of $Q_0^*(x, a)$ plus its current upper confidence bound (see Equation (3)), where the estimate $\hat{Q}_0(x, a)$ is given by the immediate reward plus the *sample mean* of V_1^*-values at the *sampled next states that have been sampled so far* (see Equation (4)).

Among the N_0 samples for state x, $N_{a,0}^x$ denotes the number of samples using action a. If the sampling is done appropriately, we might expect that $N_{a,0}^x / N_0$ provides a good estimate of the likelihood that action a is optimal in state x, because in the limit as $N_0 \to \infty$, the sampling scheme should lead to $N_{a^*,0}^x / N_0 \to 1$ if a^* is the unique optimal action, or if there are multiple optimal actions, say a set A^*, then $\sum_{a \in A^*} N_{a,0}^x / N_0 \to 1$, i.e., $\{N_{a,0}^x / N_0\}_{a \in A}$ should converge to a probability distribution concentrated on the set of optimal actions. For this reason, we use a weighted (by $N_{a,0}^x / N_0$) sum of the currently estimated value of $Q_0^*(x, a)$ over A to approximate $V_0^*(x)$ (see Equation (5)). Ensuring that the weighted sum concentrates on a^* as the sampling proceeds will ensure that in the limit the estimate of $V_0^*(x)$ converges to $V_0^*(x)$.

2.2. Algorithm Description

We now provide a high-level description of the adaptive multistage sampling (AMS) algorithm to estimate $V_0^*(x)$ for a given state x in Figure 1. The inputs to AMS are a state $x \in X$, $N_i \geqslant |A|$, and stage i, and the output of AMS is $\hat{V}_i^{N_i}(x)$, the estimate of $V_i^*(x)$. Whenever we encounter $\hat{V}_k^{N_k}(y)$ for a state $y \in X$ and stage k in the **Initialization** and **Loop** portions of the AMS algorithm, we need to call AMS recursively (at Equations (2) and (4)). The initial call to AMS is done with $i = 0$, the initial state x_0, and N_0,

Figure 1. Adaptive multistage sampling algorithm (AMS) description.

Adaptive Multistage Sampling (AMS)

- **Input:** a state $x \in X$, $N_i \geqslant |A|$, and stage i. **Output:** $\hat{V}_i^{N_i}(x)$.
- **Initialization:** Sample each action $a \in A$ sequentially once at state x and set

$$\hat{Q}_i(x,a) = \begin{cases} 0 & \text{if } i = H \text{ and go to } \textbf{Exit}, \\ R(x,a) + \gamma \hat{V}_{i+1}^{N_{i+1}}(y) & \text{if } i \neq H, \end{cases} \tag{2}$$

where y is the sampled next state with respect to $P(x,a)$, and set $\bar{n} = |A|$.

- **Loop:** Sample an action a^* that achieves

$$\max_{a\in A}\left(\hat{Q}_i(x,a) + \sqrt{\frac{2\ln \bar{n}}{N_{a,i}^x}}\right), \tag{3}$$

where $N_{a,i}^x$ is the number of times action a has been sampled so far, and $\bar{n}$ is the overall number of samples done so far for this stage, and $\hat{Q}_i$ is defined by

$$\hat{Q}_i(x,a) = R(x,a) + \gamma \frac{1}{N_{a,i}^x} \sum_{y \in S_a^x} \hat{V}_{i+1}^{N_{i+1}}(y), \tag{4}$$

where S_a^x is the set of sampled next states so far with $|S_a^x| = N_{a,i}^x$ with respect to the distribution $P(x,a)$.

— Update $N_{a^*,i}^x \leftarrow N_{a^*,i}^x + 1$ and $S_{a^*}^x \leftarrow S_{a^*}^x \cup \{y'\}$, where y' is the newly sampled next state by a^*.

— Update $\hat{Q}_i(x,a^*)$ with the $\hat{V}_{i+1}^{N_{i+1}}(y')$ value.

— $\bar{n} \leftarrow \bar{n} + 1$. If $\bar{n} = N_i$, then exit **Loop**.

- **Exit:** Set $\hat{V}_i^{N_i}(x)$ such that

$$\hat{V}_i^{N_i}(x) = \begin{cases} \displaystyle\sum_{a\in A} \frac{N_{a,i}^x}{N_i} \hat{Q}_i(x,a) & \text{if } i = 0, \ldots, H-1, \\ 0 & \text{if } i = H. \end{cases} \tag{5}$$

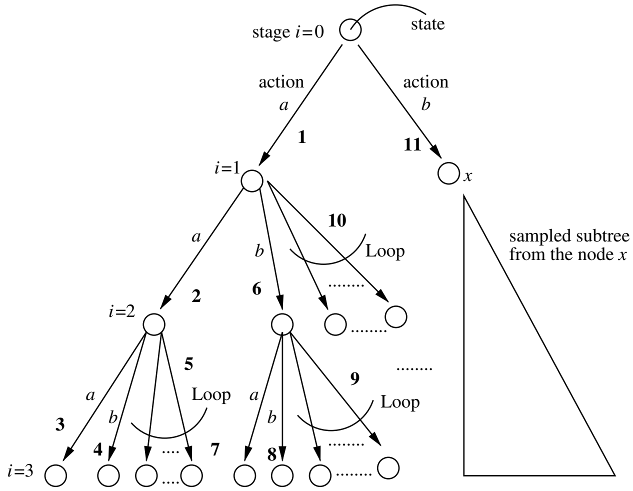
and return $\hat{V}_i^{N_i}(x)$.

and every sampling is done independently of the previous samplings. To help understand how the recursive calls are made sequentially, in Figure 2 we graphically illustrate the sequence of calls with two actions and $H = 3$ for the **Initialization** portion.

The AMS algorithm is a recursive extension of the UCB1 algorithm given in Auer et al. (2002) in the context of the MDP framework. It is based on the index-based policy of Agrawal (1995), where the index for an action is given by the sum of the current estimate of the true Q-value for the action plus a term that relates the size of the upper confidence bound.

The running time complexity of the AMS algorithm is $O((|A|N)^H)$ with $N = \max_i N_i$. To see this, let M_i be the number of recursive calls made to compute $\hat{V}_i^{N_i}$ in the *worst* case. At stage i, AMS makes at most $M_i = |A| N_i M_{i+1}$ recursive calls (in **Initialization** and **Loop**), leading to $M_0 = O((|A|N)^H)$. In contrast, backward induction has $O(H|A||X|^2)$ running-time complexity (see, e.g., Blondel and Tsitsiklis 2000). Therefore, the main benefit of AMS is independence from the state space size, but this comes at the expense of exponential (versus linear, for backward induction) dependence on both the action space and the horizon length.

Figure 2. Sequence of the recursive calls made in **Initialization** of the AMS algorithm.



Note. Each circle corresponds to a state and each arrow with noted action signifies a sampling (and a recursive call). The bold-face number near each arrow is the sequence number for the recursive calls made. For simplicity, the entire **Loop** process is signified by one call number.

2.3. Creating an Online Stochastic Policy

Once armed with an algorithm that estimates the optimal value for finite-horizon problems, we can create a nonstationary stochastic policy in an online manner in the context of "planning" (see, e.g., Kearns et al. 2001). Suppose that at time $t \geqslant 0$, we are at state $x \in X$. We evaluate each action's utility as follows:

$$R(x,a) + \gamma \frac{1}{N_{\{a,t\}^*}} \sum_{y\in S_a^x} \hat{V}_{t+1}^{N_{t+1}}(y), \quad a \in A, \tag{6}$$

where we apply the AMS algorithm at the sampled next states for the stage $t+1$ and we replace the horizon $H-1$ by $t+H$ in the definition of $\hat{V}_{t+1}^{N_{t+1}}(y)$. We simply take the action that achieves the maximum utility. The resulting stochastic policy yields an (approximate) receding H-horizon control (Hernández-Lerma and Lasserre 1990) for the infinite horizon problem. We remark that the use of common random numbers (see, e.g., Law and Kelton 2000) across actions in the utility measures given by Equation (6) should reduce the variance in the spirit of "differential training" in the rollout algorithm (Bertsekas 1997).

3. Convergence Analysis

In this section, we prove the convergence of the AMS algorithm and show that the worst possible bias converges to zero at rate $O(\sum_{i=0}^{H-1} (\ln N_i)/N_i)$.

THEOREM 3.1. *Let $R_{\max} = \max_{x,a} R(x, a)$ and assume that $R_{\max} \leqslant 1/H$. Suppose AMS is run with the input N_i for stage $i = 0, \ldots, H-1$ and an arbitrary initial state $x \in X$. Then,*

$$\lim_{N_0\to\infty}\lim_{N_1\to\infty}\cdots\lim_{N_{H-1}\to\infty} E\big[\hat{V}_0^{N_0}(x)\big] = V_0^*(x).$$

PROOF. We start with a convergence result for the one-stage approximation. Consider the one-stage sampling algorithm (OSA) in Figure 3 with a *stochastic value function U* defined over X. $U(x)$ for $x \in X$ is a *nonnegative random variable* with *unknown* distribution and bounded above for all $x \in X$. We denote $U(x)$ as a (random) sample from the unknown distribution associated with $U(x)$. As before, every sampling is done independently, and we are assuming that there is a black box that returns $U(x)$ once x is given to the black box. Let

$$U_{\max} = \max_{x,a}\bigg(R(x, a) + \gamma\sum_{y\in X} P(x, a)(y)E[U(y)]\bigg),$$

and assume for the moment that $U_{\max} \leqslant 1$.

Figure 3. One-stage sampling algorithm (OSA) description.

One-stage Sampling Algorithm (OSA)

- **Input:** a state $x \in X$ and $n \geqslant |A|$.
- **Initialization:** Sample each action $a \in A$ once at state x and set

$$\tilde{Q}(x, a) = R(x, a) + \gamma U(y),$$

where y is the sampled next state with respect to $P(x, a)$, and set $\bar{n} = |A|$.

- **Loop:** Sample an action a^* that achieves

$$\max_{a\in A}\Bigg(\tilde{Q}(x, a) + \sqrt{\frac{2\ln\bar{n}}{T_a^x(\bar{n})}}\Bigg),$$

where $T_a^x(\bar{n})$ is the number of times action a has been sampled so far at state x, $\bar{n}$ is the overall number of samples done so far, and $\tilde{Q}$ is defined by

$$\tilde{Q}(x, a) = R(x, a) + \gamma\frac{1}{T_a^x(\bar{n})}\sum_{y\in\Lambda_a^x(\bar{n})} U(y),$$

where $\Lambda_a^x(\bar{n})$ is the set of sampled next states so far with $|\Lambda_a^x(\bar{n})| = T_a^x(\bar{n})$ with respect to the distribution $P(x, a)$.

— Update $T_{a^*}^x(\bar{n}) \leftarrow T_{a^*}^x(\bar{n}) + 1$ and $\Lambda_{a^*}^x(\bar{n}) \leftarrow \Lambda_{a^*}^x(\bar{n}) \cup \{y'\}$, where y' is the newly sampled next state by a^*.

— Update $\tilde{Q}(x, a^*)$ with $U(y')$.

— $\bar{n} \leftarrow \bar{n} + 1$. If $\bar{n} = n$, then exit **Loop**.

- **Exit:** Set $\tilde{V}^n$ such that

$$\tilde{V}^n(x) = \sum_{a\in A}\frac{T_a^x(n)}{n}\tilde{Q}(x, a). \tag{7}$$

We state a key lemma that will be used to prove the convergence of the AMS algorithm.

LEMMA 3.1. *Given a stochastic value function U defined over X with $U_{\max} \leqslant 1$, suppose we run OSA with the input n. Define for all $x \in X$*

$$V(x) = \max_{a\in A}\Big(R(x, a) + \gamma\sum_{y\in X} P(x, a)(y)E[U(y)]\Big).$$

Then, for all $x \in X$,

$$E[\tilde{V}^n(x)] \to V(x) \quad \textit{as } n \to \infty.$$

PROOF OF LEMMA 3.1. Fix a state $x \in X$ and index each action in $|A|$ by numbers from 1 to $|A|$. Consider an $|A|$-armed bandit problem where each a is a gambling machine. Successive plays of machine a yield "bandit rewards," which are independent and identically distributed according to an unknown distribution δ_a with unknown expectation

$$Q(x, a) = R(x, a) + \gamma\sum_{y\in X} P(x, a)(y)E[U(y)],$$

and are independent across machines or actions.

The term $T_a^x(n)$ signifies the number of times machine a has been played (or action a has been sampled) by OSA during the n plays. Define the *expected regret $\rho(n)$* of OSA after n plays by

$$\rho(n) = V(x)n - \sum_{a=1}^{|A|} Q(x, a)E[T_a^x(n)], \qquad \text{where } V(x) = \max_{a\in A} Q(x, a).$$

Applying Theorem 1 from Auer et al. (2002) gives the following bound on $\rho(n)$.

THEOREM 3.2. *For all $|A| > 1$, if OSA is run on $|A|$ machines having arbitrary bandit reward distribution $\delta_1, \ldots, \delta_{|A|}$ with $U_{\max} \leqslant 1$,*

$$\rho(n) \leqslant \sum_{a:Q(x,a)<V(x)}\bigg[\frac{8\ln n}{V(x) - Q(x, a)} + \bigg(1 + \frac{\pi^2}{3}\bigg)(V(x) - Q(x, a))\bigg],$$

where $Q(x, a)$ is the expected value of bandit rewards with respect to δ_a.

See Auer et al. (2002) for a proof of Theorem 3.2. Observe that $\max_a(V(x) - Q(x, a)) \leqslant U_{\max}$. Let $\phi(x) = \{a \mid Q(x, a) < V(x),\, a \in A\}$, i.e., the set of nonoptimal actions for x. Define $\alpha(x)$ for $\phi(x) \neq \varnothing$, such that

$$\alpha(x) = \min_{a\in\phi(x)}(V(x) - Q(x, a)), \tag{8}$$

and note that $0 < \alpha(x) \leqslant U_{\max}$. Define

$$\tilde{V}(x) = \sum_{a=1}^{|A|}\frac{T_a^x(n)}{n}Q(x, a).$$

Applying Theorem 3.2, we have

$$\begin{aligned}0 \leqslant V(x) - E[\tilde{V}(x)] = \frac{\rho(n)}{n} &\leqslant \frac{8(|A|-1)\ln n}{n\alpha(x)} \\ &\quad + \left(1+\frac{\pi^2}{3}\right)\cdot\frac{(|A|-1)U_{\max}}{n} \\ &\leqslant \frac{C_1 \ln n}{n} + \frac{C_2}{n},\end{aligned}$$

for some constants C_1 and C_2. Note that because X is finite, there exists a constant $C > 0$ such that $0 < C \leqslant \min_{x\in X} \alpha(x)$ and also that $\rho(n) = 0$ if $\phi(x) = \varnothing$. From the definition of $\tilde{V}^n(x)$ given by Equation (7), it follows that

$$\begin{aligned}V(x)-E[\tilde{V}^n(x)] &= V(x)-E[\tilde{V}(x)-\tilde{V}(x)+\tilde{V}^n(x)] \\ &= V(x)-E[\tilde{V}(x)] \\ &\quad +E\left[\sum_{a\in A}\frac{T_a^x(n)}{n}(Q(x,a)-\tilde{Q}(x,a))\right].\end{aligned} \tag{9}$$

Letting $n\to\infty$, the first term $V(x)-E[\tilde{V}(x)]$ is bounded by zero from below with convergence rate of $O((\ln n)/n)$ by Equation (9). We show now that the second expectation term is zero.

Let $Y_j \sim \{P(x,a)\}$ denote the (i.i.d.) jth next state sampled from the same starting state x with same action a. Then, $T_a^x(n)$ for every finite n is a *stopping time* (see, e.g., Ross 1995, p. 104) for $\{Y_j\}$, because $T_a^x(n) \leqslant n < \infty$ and the event $\{T_a^x(n) = k\}$ is independent of $\{Y_{k+1},\ldots\}$. Let $\mu_a(x) = E[U(Y_j)]$. Then,

$$\begin{aligned}&E\left[\sum_{a\in A}\frac{T_a^x(n)}{n}(Q(x,a)-\tilde{Q}(x,a))\right] \\ &\quad = E\left[\sum_{a\in A}\frac{T_a^x(n)}{n}\left(R(x,a)+\gamma\mu_a(x)-R(x,a)\right.\right. \\ &\qquad\qquad \left.\left.-\gamma\frac{1}{T_a^x(n)}\sum_{j=1}^{T_a^x(n)}U(Y_j)\right)\right] \\ &\quad = \frac{\gamma}{n}\left(\sum_{a\in A}E[T_a^x(n)]\mu_a(x)-\sum_{a\in A}E\left[\sum_{j=1}^{T_a^x(n)}U(Y_j)\right]\right)=0,\end{aligned}$$

by applying Wald's equation.

Because

$$V(x)-E[\tilde{V}^n(x)] = V(x)-E[\tilde{V}(x)],$$

the convergence follows directly from Equation (9).

Therefore, because x was chosen arbitrarily, we have that for all $x\in X$,

$$E[\tilde{V}^n(x)] \to V(x) \quad \text{as } n\to\infty,$$

which concludes the proof of Lemma 3.1. □

We now return to the AMS algorithm. From the definition of $\hat{V}_{H-1}^{N_{H-1}}$,

$$\begin{aligned}\hat{V}_{H-1}^{N_{H-1}}(x) &= \sum_{a\in A}\frac{N_{a,H-1}^x}{N_{H-1}}\left(R(x,a)+\gamma\frac{1}{N_{a,H-1}^x}\sum_{y\in S_a^x}\hat{V}_H^{N_H}(y)\right) \\ &\leqslant \sum_{a\in A}\frac{N_{a,H-1}^x}{N_{H-1}}(R_{\max}+\gamma\cdot 0) = R_{\max}, \quad x\in X.\end{aligned}$$

Similarly, for $\hat{V}_{H-2}^{N_{H-2}}$, we have that

$$\begin{aligned}\hat{V}_{H-2}^{N_{H-2}}(x) &= \sum_{a\in A}\frac{N_{a,H-2}^x}{N_{H-2}}\left(R(x,a)+\gamma\frac{1}{N_{a,H-2}^x}\sum_{y\in S_a^x}\hat{V}_{H-1}^{N_{H-1}}(y)\right) \\ &\leqslant \sum_{a\in A}\frac{N_{a,H-2}^x}{N_{H-2}}(R_{\max}+\gamma R_{\max}) = R_{\max}(1+\gamma), \quad x\in X.\end{aligned}$$

Continuing this backward, we have for all $x\in X$ and $i = 0,\ldots,H-1$,

$$\hat{V}_i^{N_i}(x) \leqslant R_{\max}\sum_{j=0}^{H-i-1}\gamma^j \leqslant R_{\max}(H-i) \leqslant 1,$$

where the last inequality comes from the assumption that $R_{\max}H \leqslant 1$.

Therefore, from Lemma 3.1 with $U_{\max} = R_{\max}(H-i) \leqslant 1$, we have for $i = 0,\ldots,H-1$, and for arbitrary $x\in X$,

$$E[\hat{V}_i^{N_i}(x)] \xrightarrow{N_i\to\infty} \max_{a\in A}\left(R(x,a)+\gamma\sum_{y\in X}P(x,a)(y)E[\hat{V}_{i+1}^{N_{i+1}}(y)]\right).$$

But for arbitrary $x\in X$, because $\hat{V}_H^{N_H}(x) = V_H^*(x) = 0$, $x\in X$,

$$E[\hat{V}_{H-1}^{N_{H-1}}(x)] \xrightarrow{N_{H-1}\to\infty} V_{H-1}^*(x),$$

which in turn leads to $E[\hat{V}_{H-2}^{N_{H-2}}(x)] \to V_{H-2}^*(x)$ as $N_{H-2}\to\infty$ for arbitrary $x\in X$, and by an inductive argument, we have that

$$\lim_{N_0\to\infty}\lim_{N_1\to\infty}\cdots\lim_{N_{H-1}\to\infty}E[\hat{V}_0^{N_0}(x)] = V_0^*(x) \quad \text{for all } x\in X,$$

which concludes the proof of Theorem 3.1. □

We now argue that the worst possible bias by AMS is bounded by a quantity that converges to zero at rate $O(\sum_{i=0}^{H-1}(\ln N_i)/N_i)$. Let $B(X)$ be the space of real-valued bounded measurable functions on X endowed with the supremum norm $\|\Phi\| = \sup_x|\Phi(x)|$ for $\Phi\in B(X)$. We define an operator T: $B(X)\to B(X)$ as

$$T(\Phi)(x) = \max_{a\in A}\left\{R(x,a)+\gamma\sum_{y\in X}P(x,a)(y)\Phi(y)\right\}, \quad \Phi\in B(X),\ x\in X. \tag{10}$$

Table 1. Value function estimate for the inventory control example case (i) as a function of the number of samples at each state: $H = 3$, $M = 20$, $x_0 = 5$, $D_t \sim DU(0, 9)$, $q = 10$, $h = 1$.

(K, p)	Optimal	N	Estimator 1 (std err)	Estimator 2 (std err)	Estimator 3 (std err)
$K = 0$	10.440	4	15.03 (0.29)	9.13 (0.21)	9.56 (0.32)
$p = 1$	$s = 0$	8	12.82 (0.16)	10.21 (0.10)	10.30 (0.10)
		16	11.75 (0.09)	10.33 (0.08)	10.38 (0.08)
		32	11.23 (0.06)	10.45 (0.06)	10.49 (0.06)
$K = 0$	24.745	4	30.45 (0.87)	19.98 (0.79)	20.48 (0.82)
$p = 10$	$s = 6$	8	28.84 (0.49)	23.09 (0.55)	23.68 (0.52)
		16	26.69 (0.38)	23.88 (0.44)	23.94 (0.45)
		32	26.12 (0.14)	24.73 (0.19)	24.74 (0.18)
$K = 5$	10.490	4	18.45 (0.29)	10.23 (0.21)	10.41 (0.22)
$p = 1$	$s_1 = 0$	8	14.45 (0.15)	10.59 (0.10)	10.62 (0.10)
	$s_2 = 0$	16	12.48 (0.10)	10.51 (0.10)	10.52 (0.10)
	$s_3 = 0$	32	11.47 (0.07)	10.46 (0.06)	10.46 (0.06)
$K = 5$	31.635	4	37.52 (0.98)	26.42 (0.88)	26.92 (0.89)
$p = 10$	$s_1 = 6$	8	36.17 (0.43)	30.13 (0.49)	30.41 (0.51)
	$s_2 = 6$	16	33.81 (0.40)	30.76 (0.43)	30.80 (0.43)
	$s_3 = 5$	32	33.11 (0.16)	31.62 (0.22)	31.64 (0.22)

Note. Each entry represents the mean based on 30 independent replications (standard error in parentheses).

Define $\Psi_i \in B(X)$ such that $\Psi_i(x) = E[\hat{V}_i^{N_i}(x)]$ for all $x \in X$ and $i = 0, \ldots, H-1$ and $\Psi_H(x) = V_H^*(x) = 0$, $x \in X$. In the proof of Lemma 3.1 (see Equation (9)), we showed that for $i = 0, \ldots, H-1$,

$$T(\Psi_{i+1})(x) - \Psi_i(x) \leqslant O\left(\frac{\ln N_i}{N_i}\right), \qquad x \in X.$$

Therefore, we have

$$T(\Psi_1)(x) - \Psi_0(x) \leqslant O\left(\frac{\ln N_0}{N_0}\right), \qquad x \in X. \tag{11}$$

and

$$\Psi_1(x) \geqslant T(\Psi_2)(x) - O\left(\frac{\ln N_1}{N_1}\right), \qquad x \in X. \tag{12}$$

Applying the T-operator to both sides of Equation (12) and using the monotonicity property of T (see, e.g., Bertsekas 1995), we have

$$T(\Psi_1)(x) \geqslant T^2(\Psi_2)(x) - O\left(\frac{\ln N_1}{N_1}\right), \qquad x \in X. \tag{13}$$

Therefore, combining Equations (11) and (13) yields

$$T^2(\Psi_2)(x) - \Psi_0(x) \leqslant O\left(\frac{\ln N_0}{N_0} + \frac{\ln N_1}{N_1}\right), \qquad x \in X.$$

Repeating this argument yields

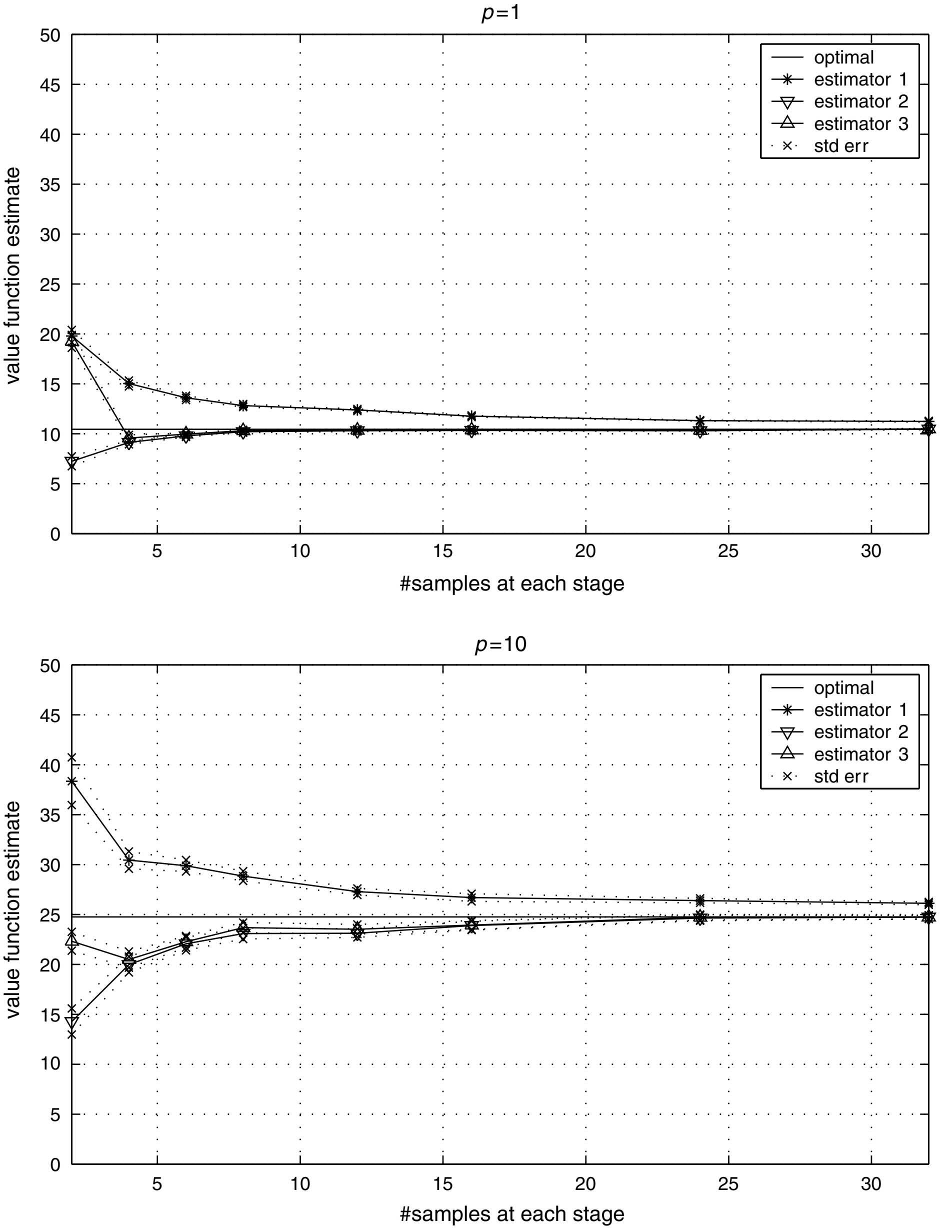
$$T^H(\Psi_H)(x) - \Psi_0(x) \leqslant O\left(\sum_{i=0}^{H-1} \frac{\ln N_i}{N_i}\right), \qquad x \in X. \tag{14}$$

Table 2. Value function estimate for the inventory control example case (ii) as a function of the number of samples at each state: $H = 3$, $M = 20$, $x_0 = 5$, $D_t \sim DU(0, 9)$, $h = 1$.

(K, p)	Optimal	N	Estimator 1 (std err)	Estimator 2 (std err)	Estimator 3 (std err)
$K = 0$	7.500	21	24.06 (0.16)	3.12 (0.17)	9.79 (0.21)
$p = 1$	$S = 4$	25	22.05 (0.12)	5.06 (0.12)	6.28 (0.19)
		30	20.36 (0.11)	5.91 (0.09)	6.47 (0.09)
		35	18.82 (0.11)	6.26 (0.10)	6.62 (0.11)
$K = 0$	13.500	21	29.17 (0.21)	6.04 (0.30)	13.69 (0.46)
$p = 10$	$S = 9$	25	28.08 (0.21)	9.28 (0.23)	12.06 (0.29)
		30	27.30 (0.19)	11.40 (0.20)	13.28 (0.23)
		35	26.06 (0.16)	12.23 (0.18)	13.07 (0.16)
$K = 5$	10.490	21	33.05 (0.12)	8.73 (0.21)	18.62 (0.44)
$p = 1$	$s_1 = 0, S_1 = 0$	25	29.99 (0.10)	10.96 (0.11)	11.79 (0.16)
	$s_2 = 0, S_2 = 0$	30	27.45 (0.10)	11.22 (0.05)	11.52 (0.07)
	$s_3 = 0, S_3 = 0$	35	25.33 (0.09)	10.96 (0.06)	11.12 (0.07)
$K = 5$	25.785	21	39.97 (0.22)	17.78 (0.49)	26.76 (0.52)
$p = 10$	$s_1 = 6, S_1 = 9$	25	39.01 (0.19)	22.68 (0.26)	25.09 (0.33)
	$s_2 = 6, S_2 = 9$	30	38.03 (0.16)	24.35 (0.17)	25.45 (0.27)
	$s_3 = 6, S_3 = 9$	35	36.89 (0.12)	24.71 (0.23)	25.51 (0.28)

Note. Each entry represents the mean based on 30 independent replications (standard error in parentheses).

Figure 4. Convergence of value function estimate for the inventory control example case (i) $q = 10$ as a function of the number of samples at each stage: $H = 3, M = 20, x_0 = 5, D_t \sim DU(0, 9), h = 1, K = 0$.



Observe that $T^H(\Psi_H)(x) = V_0^*(x)$, $x \in X$. Rewriting Equation (14), we finally have

$$V_0^*(x) - E[\hat{V}_0^{N_0}(x)] \leqslant O\left(\sum_{i=0}^{H-1} \frac{\ln N_i}{N_i}\right), \quad x \in X,$$

and we know that $V_0^*(x) - E[\hat{V}_0^{N_0}(x)] \geqslant 0$, $x \in X$. Therefore, it implies that the worst possible bias is bounded by the quantity that converges to zero at rate $O(\sum_{i=0}^{H-1}(\ln N_i)/N_i)$.

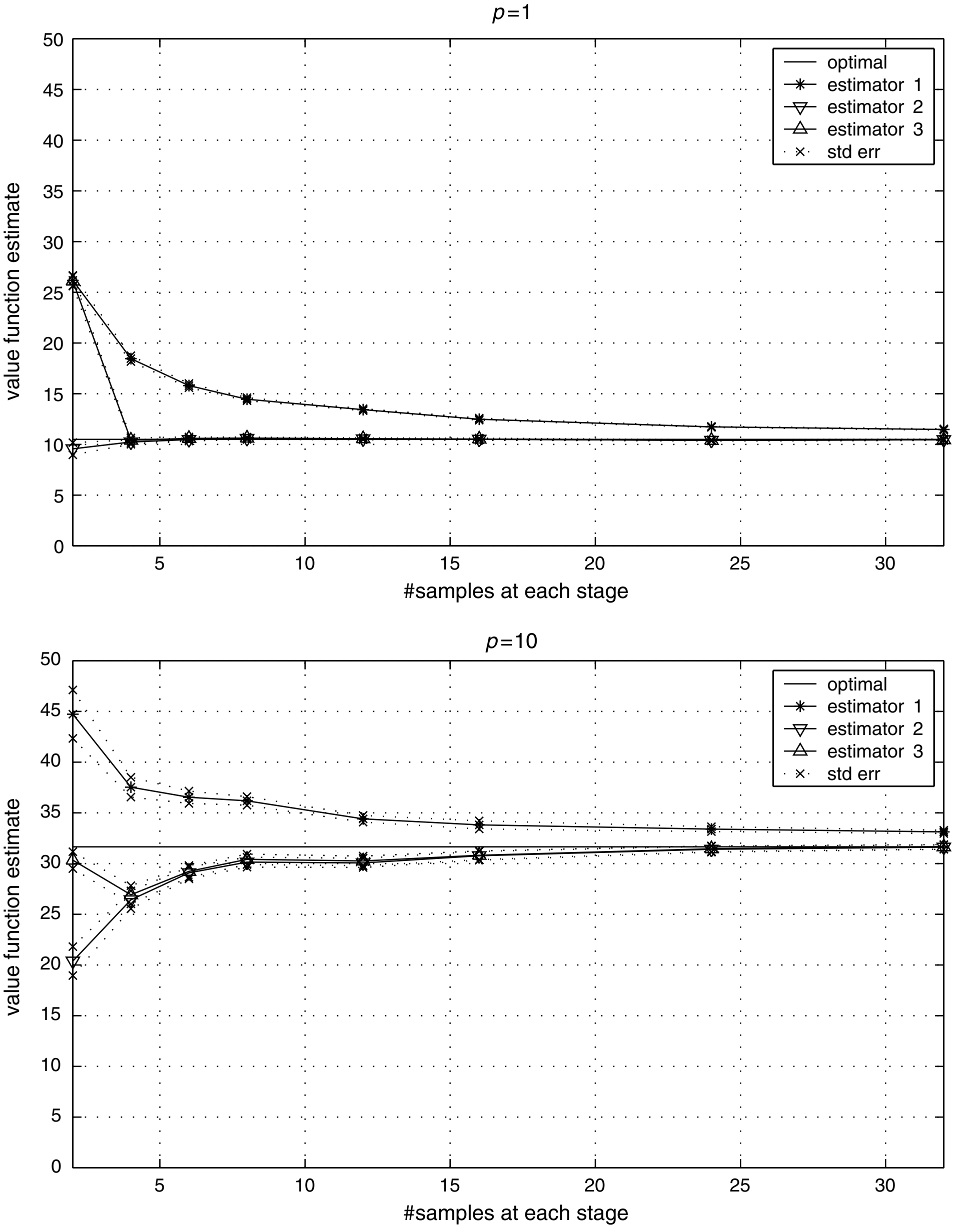
Remark 1. We can relax the assumption $R_{\max} \leqslant 1/H$, by a normalization of the given reward function. The upper bound in Theorem 3.2 for $\rho(n)$ needs to be modified with a different bounded constant from $1 + \pi^2/3$, with support in $[0, R_{\max}H]$ rather than in $[0, 1]$. This can be achieved by the Hoeffding inequality (Hoeffding 1963):

$$P\left\{\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| > \epsilon\right\} \leqslant 2\exp\left(-\frac{2\epsilon^2 n^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right), \qquad (15)$$

where X_i are independent random variables with bounded support $[a_i, b_i]$ and finite common mean μ. Therefore, the assumption of the support in $[0, 1]$ is not crucial (Cesa-Bianchi and Fisher 1998).

Remark 2. Our proofs for the convergence can be extended to infinite state space X as long as we can ensure

Figure 5. Convergence of value function estimate for the inventory control example case (i) $q = 10$ as a function of the number of samples at each stage: $H = 3$, $M = 20$, $x_0 = 5$, $D_t \sim DU(0, 9)$, $h = 1$, $K = 5$.



that there exists a constant $C > 0$ such that $0 < C \leqslant \inf_{x \in X} \alpha(x)$.

4. A Numerical Example

To illustrate the algorithm, we consider some computational experiments on a simple example: a finite-horizon inventory control problem with lost sales. The objective is to find the (nonstationary) policy to minimize expected costs, which comprise holding, order, and penalty costs. Demand is a discrete random variable. Given an inventory level, orders are placed and received, demand is realized, and the new inventory level for the period is calculated, on which costs are charged.

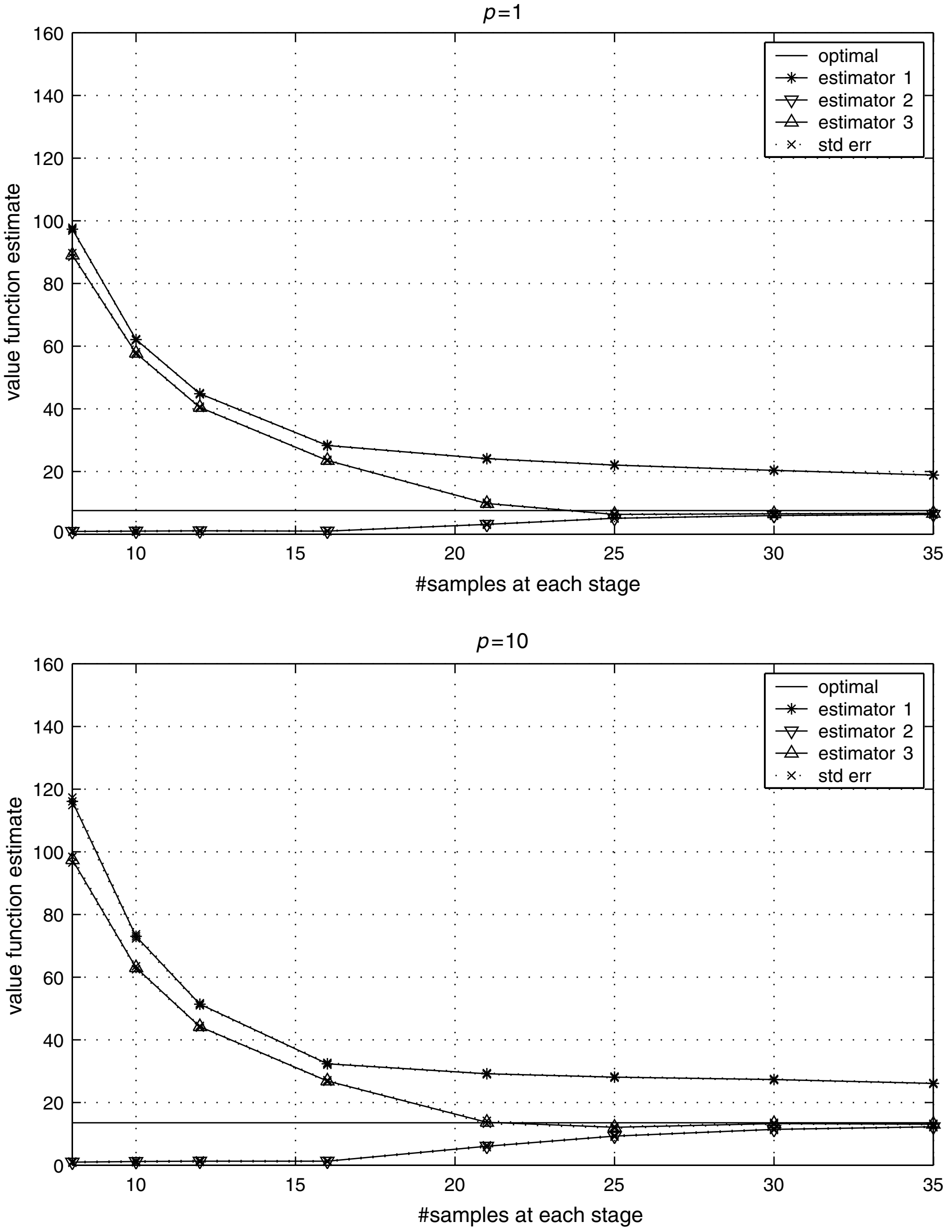
Let D_t denote the demand in period t, x_t the inventory level at the end of period t (which is the inventory at the beginning of period $t+1$), a_t the order amount in period t, p the per period per unit demand lost penalty cost, h the per period per unit inventory holding cost, K the fixed (set-up) cost per order, and M the maximum inventory level (storage capacity), i.e., $x_t \in \{0, 1, \ldots, M\}$. Then the state transition follows the dynamics:

$$x_{t+1} = (x_t + a_t - D_t)^+.$$

The objective function is the expectation of the total cost given by

$$\sum_{t=1}^{H} [h(x_t + a_t - D_t)^+ + p(D_t - x_t - a_t)^+ + K \cdot \mathbf{1}\{a_t > 0\}],$$

Figure 6. Convergence of value function estimate for the inventory control example case (ii) as a function of the number of samples at each stage: $H = 3, M = 20, x_0 = 5, D_t \sim DU(0, 9), h = 1, K = 0$.



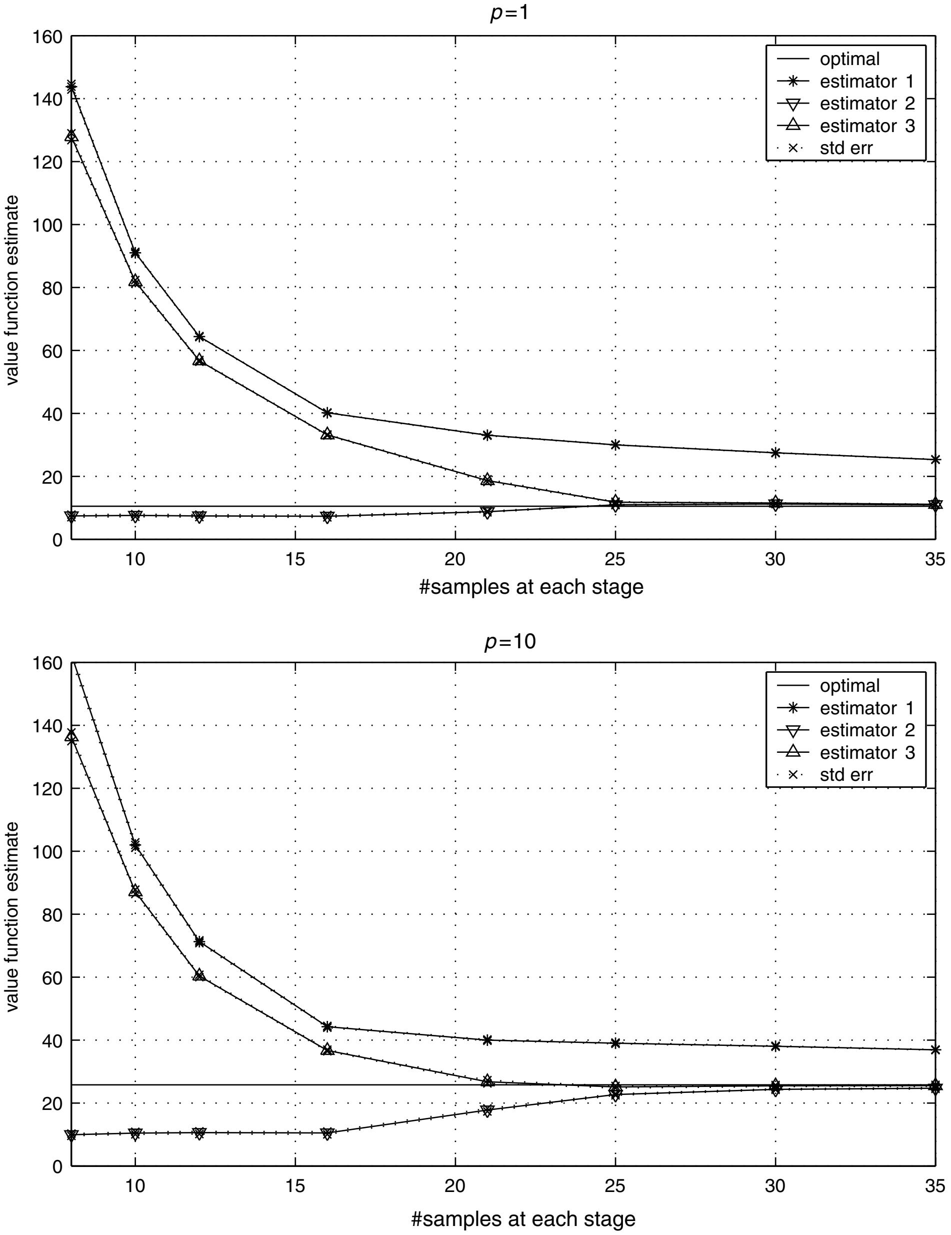
where x_0 is the starting inventory level, H is the number of periods (time horizon), and $\mathbf{1}\{\cdot\}$ denotes the indicator function. Note that we are ignoring per-unit order costs for simplicity.

We consider two versions: (i) fixed order amount q, and (ii) any (integral) order amount (up to capacity). In both cases, if the order amount would bring the inventory level above the inventory capacity M, then that order cannot be placed, i.e., that order amount action is not feasible in that state. In case (i), there are just two actions (order or no order), whereas in case (ii), the number of actions depends on the capacity limit.

Central to the context of the algorithm are that the underlying distribution is unknown and that only samples are available. Furthermore, there is no structural knowledge on the form of the optimal policy. However, the example selected here was chosen to be simple in order to allow for the optimal solution to be solved easily by standard techniques once the distribution is given, so that the performance of the algorithm could be evaluated.

In addition to the original estimator given by (5), we considered two alternative estimators. First, consider the estimator that replaces the weighted sum of the Q-value estimates in Equation (5) by the maximum of the estimates,

Figure 7. Convergence of value function estimate for the inventory control example case (ii) as a function of the number of samples at each stage: $H = 3, M = 20, x_0 = 5, D_t \sim DU(0, 9), h = 1, K = 5$.



i.e., for $i < H$,

$$\hat{V}_i^{N_i}(x) = \max_{a \in A} \hat{Q}_i(x, a),$$
$$\hat{Q}_i(x, a) = R(x, a) + \gamma \frac{1}{N_{a,i}^x} \sum_{y \in \Lambda_a^x} \hat{V}_{i+1}(y). \tag{16}$$

For the nonadaptive case, it can be shown that this estimator is also asymptotically unbiased, but with a finite-sample "optimistic" bias in the opposite direction as the original estimator (i.e., upward for maximization problems and downward for minimization problems such as the inventory control problem).

Next, consider an estimator that chooses the action that has been sampled the most thus far in order to estimate the value function. It can be easily shown that this estimator is less optimistic than the previous alternative, and so we combine it with the original estimator to obtain the following alternative estimator:

$$\bar{V}_i(x) = \max\left\{ \bar{Q}_i(x, a^*), \sum_{a \in A} \frac{N_{a,i}^x}{N_i} \bar{Q}_i(x, a) \right\}, \tag{17}$$

where

$$a^* = \arg\max_a \{N_{a,i}^x\},$$
$$\bar{Q}_i(x, a) = R(x, a) + \gamma \frac{1}{N_{a,i}^x} \sum_{y \in \Lambda_a^x} \bar{V}_{i+1}(y).$$

Intuitively, the rationale behind combining via the max operator, instead of just the straightforward $\bar{V}_i(x) = \bar{Q}_i(x, a^*)$, is that the estimator would be choosing the best between two possible estimates of the Q-function. Actually, a similar combination could also be used for (16), as well, to derive yet another possible estimator.

In actual implementation, a slight modification is required for this example because it is a minimization problem, whereas AMS was written for a maximization problem. Conceptually, the most straightforward way would be to just take the reward as the negative of the cost function. However, we instead leave the problem as a minimization, in which case we need to replace the "max" operator with the "min" operator and the addition with subtraction in Equations refeqn:first), (16), and (17), i.e., respectively,

$$\min_{a\in A}\left(\hat{Q}_i(x,a) - \sqrt{\frac{2\ln \bar{n}}{N^x_{a,i}}}\right),$$

$$\hat{V}_i^{N_i}(x) = \min_{a\in A} \hat{Q}_i(x,a),$$

$$\bar{V}_i(x) = \min\left\{\bar{Q}_i(x,a^*), \sum_{a\in A}\frac{N^x_{a,i}}{N_i}\bar{Q}_i(x,a)\right\}.$$

Note that the operator in defining $a^* = \arg\max_a\{N^x_{a,i}\}$ remains a maximization operation.

With $K = 0$ (no fixed order cost), the optimal order policy is easily solvable without dynamic programming, because the periods are decoupled, and the problem reduces to solving a single-period inventory optimization problem. In case (i), the optimal policy follows a threshold rule, in which an order is placed if the inventory is below a certain level; otherwise, no order is placed. The threshold (order point) is given by

$$s = \min_{x\geqslant 0}\{x\colon hE[(x+q-D)^+] + pE[(D-q-x)^+] \geqslant hE[(x-D)^+] + pE[(D-x)^+]\},$$

i.e., one orders in period t if $x_t < s$ (assuming that $x_t + q \leqslant M$; also, if the set is empty, then take $s = \infty$, i.e., an order will always be placed). In case (ii), the problem becomes a newsboy problem with a base-stock (order up to) solution given by

$$S = F^{-1}(p/(p+h)),$$

i.e., one orders $(S - x_t)^+$ in period t (with the implicit assumption that $S \leqslant M$).

For the $K > 0$ case (i), the optimal policy is again a threshold (order point) policy but the order point is nonstationary; whereas in case (ii), the optimal policy is of the (s, S) type, again nonstationary. To obtain the true solutions, standard backward induction was employed, using knowledge of the underlying demand distribution.

For the numerical experiments, we used the following parameter settings: horizon $H = 3$; capacity $M = 20$; initial inventory $x_1 = 5$; demand $D_t \sim DU(0, 9)$ (discrete uniform); holding cost $h = 1$; penalty cost $p = 1$ and $p = 10$; fixed order cost $K = 0$ and $K = 5$; fixed order amount for case (i): $q = 10$. Note that because the order quantity is greater than the maximum demand for our values of the parameters, i.e., $q > D_t$ always, placing an order guarantees no lost sales.

Tables 1 and 2 give the performances of these estimators for each of the respective cases (i) and (ii), including the optimal value and policy parameters. Figures 4 through 7 show the convergence of the estimates as a function of the number of samples at each stage for each of the respective cases (i) and (ii) considered. In each table and figure, estimator 1 stands for the original estimator using (5), and estimators 2 and 3 refer to the estimators using $\hat{V}(x)$ from (16) and $\bar{V}(x)$ from (17), respectively. The results indicate convergence of all three estimators, with the two alternative estimators providing superior empirical performance over the original estimator. We conjecture that this is because the original estimator's use of a weighted average is too conservative, thus leading to unnecessarily slow convergence. We suspect this would be the case for the nonadaptive sampling version using a weighted average estimator, too.

5. Conclusions and Future Research

We have proposed a framework for approximating the optimal value function of a finite-horizon, finite-action MDP using adaptive sampling based on multiarmed bandit models and have proved convergence and analyzed the convergence rate of an algorithm based on the framework, for the case of finite state spaces. Clearly, this very general framework will yield the most computational benefits in cases where the sampling cost is relatively expensive. Because an MDP involves *sequential* decision making, we anticipate that the potential savings from adopting an adaptive approach are substantial, since waste in sampling prospectively "poor" actions may be compounded, as contrasted, say, with a stochastic optimization problem in the setting where a single static objective function is sampled. We have in mind the setting of stochastic discrete-event simulation. For example, in a capacity planning model in manufacturing, the transitions and cost/rewards in the MDP model might correspond to outputs from a run of a large simulation model of a complex semiconductor fabrication facility, and the action might be a choice of whether or not to add long-term capacity by purchasing an expensive new piece of machinery. Preliminary numerical results on a simple example indicate that the approach is promising, but there is much left to investigate to make the implementation to specific large-scale applications more practically useful. Although we were able to prove convergence only for the finite state space setting, we believe the framework is applicable to infinite state spaces with the AMS algorithm unchanged, but the theoretical analysis might require a different approach. (Remark 2 gives a sufficient condition for convergence, but this condition is generally difficult to verify in practice.) An important future research topic is a comprehensive comparison—theoretical and/or empirical—of computational efficiency in terms of algorithm performance between the adaptive sampling framework proposed here and nonadaptive (fixed sampling) approaches.

We can also extend the AMS algorithm to include the case where the reward function is random. Again, the AMS algorithm would remain essentially identical, except that sampling would now include both the next state and the one-stage reward. Actually, this is the setting considered in the inventory control numerical examples (which could have been artificially converted into the deterministic reward framework by analytically computing the expected value for the cost function using the known demand distribution and only sampling from the distribution to generate the next state; this was, in fact, tested but not reported here, and the convergence was considerably faster, as might be expected). However, the convergence proof is likely to require more technical manipulations. Furthermore, the assumption of bounded rewards can be relaxed by using the result in Agrawal (1995). Even though the AMS algorithm will converge also in this case, unfortunately, we lose the property of the uniform logarithmic bound so that the convergence rate is expected to be very slow.

For problems where a relatively small set of states are likely to be revisited, it might be advantageous to store calculated values of $\hat{V}_i^{N_i}$ to avoid having to possibly recompute them, which could result in substantial savings for longer-horizon problems, because it would also avoid the costly recursive calls. The trade-off in additional required storage, possibly unmanageable for very large state spaces, would have to be evaluated against the estimated resultant gains in running time.

While conducting computational experiments on the inventory control example, we considered two alternative estimators, both of which empirically outperform the original weighted average estimator. It is conjectured that both of these alternatives are asymptotically unbiased, with the estimator given by (16) having an "optimistic" bias (i.e., high for maximization problems, low for minimization problems). If so, valid—albeit conservative—confidence intervals for the optimal value could also be easily derived by combining the two oppositely biased estimators. Such a result can be established for the nonadaptive versions of the estimators, but proving these results in our setting and characterizing the convergence rate of the estimator given by (16) in a similar manner as for the original estimator are still ongoing research problems.

Choosing an appropriate sample size is an important topic in practical application. The empirical performance of the two alternative estimators indicates that a heuristic stopping rule for choosing the number of samples at each stage could be based on these two estimates, which showed rapid convergence in the numerical examples. This convergence implies that in Equation (17), the first term in the "max" operator dominates the second term (i.e., the original estimator), and the actions that have been sampled the most of the times almost "always" yield the largest Q-function values; in other words, at this point estimators 2 and 3 are "almost" the same, so if they are biased in opposite directions, they must have reached a sample size at which they are "nearly" unbiased. Once this is the case, it may be preferable to perform more independent replications at a particular action than to sample more actions (larger N).

Endnote

1. Throughout the paper, the notation O indicates that for given two functions f and g, $f(n) = O(g(n))$ if $\limsup_{n\to\infty}(f(n)/g(n)) < \infty$, and the notation Θ is used if $f(n) = O(g(n))$ and $g(n) = O(f(n))$ (cf., Cormen et al. 1990). The O and Θ-notations are often called the asymptotic upper bound and the asymptotically tight bound, respectively, for the asymptotic running time of an algorithm.

Acknowledgments

This work was supported in part by the National Science Foundation under grant DMI-9988867, in part by the Air Force Office of Scientific Research under grant F496200110161, and in part by the Department of Defense under contract MDA 90499C2521. The work of Hyeong Soo Chang was also supported by the Sogang University research grants in 2003. The authors thank the associate editor and two anonymous referees for their many constructive comments and suggestions, which have led to an improved paper.

References

Agrawal, R. 1995. Sample mean based index policies with $O(\log n)$ regret for the multiarmed bandit problem. *Advances Appl. Probab.* **27** 1054–1078.

Agrawal, R., D. Teneketzis, V. Anantharam. 1989. Asymptotically efficient adaptive allocation schemes for controlled Markov chains: Finite parameter space. *IEEE Trans. Automat. Control* **34** 1249–1259.

Auer, P., N. Cesa-Bianchi, P. Fisher. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* **47** 235–256.

Bertsekas, D. P. 1995. *Dynamic Programming and Optimal Control*, Vols. 1 and 2. Athena Scientific, Belmont, MA.

Bertsekas, D. P. 1997. Differential training of rollout policies. *Proc. 35th Allerton Conf. Communication, Control, Comput.*, Allerton Park, IL, 913–922.

Blondel, V. D., J. Tsitsiklis. 2000. A survey of computational complexity results in systems and control. *Automatica* **36** 1249–1274.

Broadie, M., P. Glasserman. 1997. Pricing American-style securities using simulation. *J. Econom. Dynamics Control* **21** 1323–1352.

Cesa-Bianchi, N., P. Fisher. 1998. Finite-time regret bounds for the multiarmed bandit problem. *Proc. 15th Int. Conf. Machine Learning*. Morgan Kaufmann Publishers, San Francisco, CA, 101–108.

Cormen, T. H., C. E. Leiserson, R. L. Rivest. 1990. *Introduction to Algorithms*. MIT Press, Cambridge, MA.

Graves, T. L., T. L. Lai. 1997. Asymptotically efficient adaptive choice of control laws in controlled Markov chains. *SIAM J. Control Optim.* **35** 715–743.

Grimmett, G., D. Stirzaker. 2001. *Probability and Random Processes,* 3rd ed. Oxford University Press, New York.

Hernández-Lerma, O., J. B. Lasserre. 1990. Error bounds for rolling horizon policies in discrete-time Markov control processes. *IEEE Trans. Automat. Control* **35** 1118–1124.

Hoeffding, W. 1963. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.* **58** 13–30.

Kearns, M., Y. Mansour, A. Y. Ng. 2001. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning* **49** 193–208.

Lai, T., H. Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances Appl. Math.* **6** 4–22.

Law, A. M., W. D. Kelton. 2000. *Simulation Modeling and Analysis,* 3rd ed. McGraw-Hill, New York.

Ross, S. 1995. *Stochastic Process*, 2nd ed. John Wiley and Sons, New York.