

# Recursive Learning Automata Approach to Markov Decision Processes

Hyeong Soo Chang, Michael C. Fu, Jiaqiao Hu, and Steven I. Marcus

## Abstract

We present a sampling algorithm, called “Recursive Automata Sampling Algorithm” (RASA), for control of finite horizon Markov decision processes. By extending in a recursive manner the learning automata Pursuit algorithm of Rajaraman and Sastry [5] designed for solving stochastic optimization problems, RASA returns an estimate of both the optimal action from a given state and the corresponding optimal value. Based on the finite-time analysis of the Pursuit algorithm, we provide an analysis for the finite-time behavior of RASA. Specifically, for a given initial state, we derive the following probability bounds as a function of the number of samples: (i) a lower bound on the probability that RASA will sample the optimal action; and (ii) an upper bound on the probability that the deviation between the true optimal value and the RASA estimate exceeds a given error.

## Index Terms

Markov decision process, learning automata, sampling.

H.S. Chang is with the Department of Computer Science and Engineering at Sogang University, Seoul 121-742, Korea, and can be reached by e-mail at [hschang@sogang.ac.kr](mailto:hschang@sogang.ac.kr).

M.C. Fu is with the Robert H. Smith School of Business and Institute for Systems Research at the University of Maryland, College Park, USA, and can be reached by email at [mfu@rhsmith.umd.edu](mailto:mfu@rhsmith.umd.edu).

J. Hu and S.I. Marcus are with the Department of Electrical and Computer Engineering and Institute for Systems Research at the University of Maryland, College Park, USA, and can be reached by e-mail at [{jqhu,marcus}@eng.umd.edu](mailto:{jqhu,marcus}@eng.umd.edu).

This work was supported in part by the National Science Foundation under Grant DMI-0323220, in part by the Air Force Office of Scientific Research under Grant FA95500410210, and in part by the Department of Defense.

A preliminary version of this paper appeared in the *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005.

## I. INTRODUCTION

Consider the following finite horizon ( $H < \infty$ ) Markov decision processes (MDP) model:

$$x_{i+1} = f(x_i, a_i, w_i) \text{ for } i = 0, 1, 2, \dots, H-1,$$

where  $x_i$  is a random variable ranging over a (possibly infinite) state set  $X$  giving the state at stage  $i$ ,  $a_i$  is the control to be chosen from a finite action set  $A$  at stage  $i$ ,  $w_i$  is a random disturbance uniformly and independently selected from  $[0,1]$  at stage  $i$ , representing the uncertainty in the system, and  $f : X \times A \times [0,1] \rightarrow X$  is the next-state function.

Let  $\Pi$  be the set of all possible nonstationary Markovian policies  $\pi = \{\pi_i | \pi_i : X \rightarrow A, i = 0, \dots, H-1\}$ . Defining the optimal reward-to-go value function for state  $x$  in stage  $i$  by

$$V_i^*(x) = \sup_{\pi \in \Pi} E_w \left[ \sum_{t=i}^{H-1} R(x_t, \pi_t(x_t), w_t) \middle| x_i = x \right],$$

where  $x \in X, w = (w_i, w_{i+1}, \dots, w_{H-1}), w_j \sim U(0,1), j = i, \dots, H-1$ , with a bounded nonnegative reward function  $R : X \times A \times [0,1] \rightarrow \mathcal{R}^+$ ,  $V_H^*(x) = 0$  for all  $x \in X$ , and  $x_t = f(x_{t-1}, \pi_{t-1}(x_{t-1}), w_{t-1})$  a random variable denoting the state at stage  $t$  following policy  $\pi$ , we wish to compute  $V_0^*(x_0)$  and obtain an optimal policy  $\pi^* \in \Pi$  that achieves  $V_0^*(x_0), x_0 \in X$ .  $R$  is bounded such that  $R_{\max} := \sup_{x \in X, a \in A, w \in [0,1]} R(x, a, w) < \infty$ , and for simplicity it is assumed that every action in  $A$  is admissible at every state and the same random number is associated with the reward and transition functions.

This paper presents a sampling algorithm, called ‘‘Recursive Automata Sampling Algorithm’’ (RASA), for estimating  $V_0^*(x_0)$  and obtaining an approximately optimal policy. RASA extends in a recursive manner (for sequential decisions) the Pursuit algorithm by Rajaraman and Sastry [5] in the context of solving MDPs. Because it uses sampling, the algorithm complexity is independent of the state space size.

The Pursuit algorithm is based on well-known learning automata [8] for solving (non-sequential) stochastic optimization problems. A learning automaton is associated with a finite set of actions (candidate solutions) and updates a probability distribution over the set by iterative interaction with an environment and takes (samples) an action according to the newly updated distribution. The environment provides a certain reaction (reward) to the action taken by the automaton, where the reaction is random and the distribution is unknown to the automaton. The automaton’s aim is to learn to choose the optimal action that yields the highest average reward.

In the Pursuit algorithm, the automaton *pursues* the current best optimal action by increasing the probability of sampling that action while decreasing the probabilities of all other actions.

As learning automata are well-known *adaptive* decision-making devices operating in unknown random environments, RASA’s sampling process of taking an action at the sampled state is adaptive at each stage. At each sampled state at a stage, a fixed sampling budget is allocated among feasible actions, and the budget is used with the current probability estimate for the optimal action. RASA builds a sampled tree in a recursive manner to estimate the optimal value  $V_0^*(x_0)$  at an initial state  $x_0$  in a bottom-up fashion and makes use of an adaptive sampling scheme of the action space while building the tree. Based on the finite-time analysis of the Pursuit algorithm, we analyze the finite-time behavior of RASA, providing in terms of the sampling parameters of RASA for a given  $x_0$ : (i) a lower bound on the probability that the optimal action is sampled, and (ii) an upper bound on the probability that the difference between the estimate of  $V_0^*(x_0)$  and the true value of  $V_0^*(x_0)$  exceeds a given error.

The book by Poznyak et al. [3] provides a good survey on using learning automata for solving controlled (ergodic) Markov chains in a model-free reinforcement learning (RL) framework for a loss function defined on the chains. Each automaton is associated with a state and updates certain functions (including the probability distribution over the action space) at each iteration of various learning algorithms. Wheeler and Narendra [9] consider controlling ergodic Markov chains for infinite horizon average reward within a similar RL framework. As far as the authors are aware, for all of the learning automata approaches proposed in the literature, the number of automata grows with the size of the state space, and thus the corresponding updating procedure becomes cumbersome, with the exception of Wheeler and Narendra’s algorithm, where only the currently visited automaton (state) updates relevant functions. Santharam and Sastry [6] provide an adaptive method for solving infinite horizon MDPs via a stochastic neural network model in the RL framework. For each state, there are  $|A|$ -units of nodes associated with the estimates of “state-action values.” As in [9], the probability distribution over the action space at an observed state is updated with an observed payoff. Even though they provide a probabilistic performance guarantee for their approach, the network size grows with the sizes of the state and action spaces.

Jain and Varaiya [2] provide a *uniform* bound on the empirical performance of policies within a simulation model of (partially observable) MDPs, which would be useful in many contexts. However, the main interest here is an algorithm and its sampling complexity for estimating the

*optimal* value/policy, and not the value function over the *entire* set of feasible policies.

The adaptive multi-stage sampling (AMS) algorithm in [1] also uses adaptive sampling in a manner similar to RASA, but the sampling is based on the exploration-exploitation process of multi-armed bandits from regret analysis, and the performance analysis of AMS is given in terms of the expected bias. In contrast, RASA's sampling is based on the probability estimate for the optimal action, and probability bounds are provided for the performance analysis of RASA.

This paper is organized as follows. We present RASA in Section II and analyze it in Section III. We conclude our paper in Section IV.

## II. ALGORITHM

It is well known (see, e.g., [4]) that  $V_i^*$  can be written recursively as follows: for all  $x \in X$  and  $i = 0, \dots, H - 1$ ,

$$\begin{aligned} V_i^*(x) &= \max_{a \in A} Q_i^*(x, a), \\ Q_i^*(x, a) &= E_w[R(x, a, w) + V_{i+1}^*(f(x, a, w))], \end{aligned}$$

$w \sim U(0, 1)$ , and  $V_H^*(x) = 0, x \in X$ . We provide below a high-level description of RASA to estimate  $V_0^*(x_0)$  for a given initial state  $x_0$  via this set of equations. The inputs to RASA are the stage  $i$ , a state  $x \in X$ , and sampling parameters  $K_i > 0$ ,  $\mu_i \in (0, 1)$ , and the output of RASA is  $\hat{V}_i^{K_i}(x)$ , the estimate of  $V_i^*(x)$ , where  $\hat{V}_H^{K_H}(x) = V_H^{K_H}(x) = 0 \ \forall K_H, x \in X$ . Whenever  $\hat{V}_{i'}^{K_{i'}}(y)$  (for stage  $i' > i$  and state  $y$ ) is encountered in the **Loop** portion of RASA, a recursive call to RASA is required (at (1)). The initial call to RASA is done with stage  $i = 0$ , the initial state  $x_0$ ,  $K_0$ , and  $\mu_0$ , and every sampling is independent of previous samplings.

Basically, RASA builds a sampled tree of depth  $H$  with the root node being the initial state  $x_0$  at stage 0 and a branching factor of  $K_i$  at each level  $i$  (level 0 corresponds to the root). The root node  $x_0$  corresponds to an automaton and initializes the probability distribution over the action space  $P_{x_0}$  as the uniform distribution (refer to **Initialization** in RASA). The  $x_0$ -automaton then samples an action and a random number (an action and a random number together corresponding to an edge in the tree) independently w.r.t. the current probability distribution  $P_{x_0}(k)$  and  $U(0, 1)$ , respectively, at each iteration  $k = 0, \dots, K_0 - 1$ . If action  $a(k) \in A$  is sampled, the count variable  $N_{a(k)}^0(x_0)$  is incremented. From the sampled action  $a(k)$  and the random number  $w_k$ ,

the automaton obtains an environment response

$$R(x_0, a(k), w_k) + \hat{V}_1^{K_1}(f(x_0, a(k), w_k)).$$

Then by averaging the responses obtained for each action  $a \in A$  such that  $N_a^0(x_0) > 0$ , the  $x_0$ -automaton computes a sample mean for  $Q_0^*(x_0, a)$ :

$$\hat{Q}_0^{K_0}(x_0, a) = \frac{1}{N_a^0(x_0)} \sum_{j: a(j)=a} R(x_0, a, w_j) + \hat{V}_1^{K_1}(f(x_0, a, w_j)),$$

where  $\sum_{a \in A} N_a^0(x_0) = K_0$ . At each iteration  $k$ , the  $x_0$ -automaton obtains an estimate of the optimal action by taking the action that achieves the current best  $Q$ -value (cf. (2)) and updates the probability distribution  $P_{x_0}(k)$  in the direction of the current estimate of the optimal action  $\hat{a}^*$  (cf. (3)). In other words, the automaton *pursues* the current best action, and hence the nonrecursive one-stage version of this algorithm is called the Pursuit algorithm [5]. After  $K_0$  iterations, RASA estimates the optimal value  $V_0^*(x_0)$  by  $\hat{V}_0^{K_0}(x_0) = \hat{Q}_0^{K_0}(x_0, \hat{a}^*)$ .

The overall estimate procedure is replicated recursively at those sampled states (corresponding to nodes in level 1 of the tree)  $f(x_0, a(k), w_k)$ -automata, where the estimates returned from those states  $\hat{V}_1^{K_1}(f(x_0, a(k), w_k))$  comprise the environment responses for the  $x_0$ -automaton.

### Recursive Automata Sampling Algorithm (RASA)

- **Input:** stage  $i \neq H$ , state  $x \in X$ ,  $K_i > 0$ ,  $\mu_i \in (0, 1)$ .  
(If  $i = H$ , then **Exit** immediately, returning  $V_H^{K_H}(x) = 0$ .)
- **Initialization:** Set  $P_x(0)(a) = 1/|A|$ ,  $N_a^i(x) = 0$ ,  $M_i^{K_i}(x, a) = 0 \quad \forall a \in A; k = 0$ .
- **Loop** until  $k = K_i$ :
  - Sample  $a(k) \sim P_x(k)$ ,  $w_k \sim U(0, 1)$ .
  - Update  $Q$ -value estimate for  $a = a(k)$  only:

$$M_i^{K_i}(x, a(k)) \leftarrow M_i^{K_i}(x, a(k)) + R(x, a(k), w_k) + \hat{V}_{i+1}^{K_{i+1}}(f(x, a(k), w_k)), \quad (1)$$

$$N_{a(k)}^i(x) \leftarrow N_{a(k)}^i(x) + 1,$$

$$\hat{Q}_i^{K_i}(x, a(k)) \leftarrow \frac{M_i^{K_i}(x, a(k))}{N_{a(k)}^i(x)}.$$

- Update optimal action estimate: (ties broken arbitrarily)

$$\hat{a}^* = \arg \max_{a \in A} \hat{Q}_i^{K_i}(x, a). \quad (2)$$

- Update probability distribution over action space:

$$P_x(k+1)(a) \leftarrow (1 - \mu_i)P_x(k)(a) + \mu_i I\{a = \hat{a}^*\} \quad \forall a \in A, \quad (3)$$

where  $I\{\cdot\}$  denotes the indicator function.

- $k \leftarrow k + 1$ .

- **Exit:** Return  $\hat{V}_i^{K_i}(x) = \hat{Q}_i^{K_i}(x, \hat{a}^*)$ .

The running time complexity of RASA is  $O(K^H)$  with  $K = \max_i K_i$ , independent of the state space size. (For some performance guarantees, the value  $K$  depends on the size of the action space. See the next section.)

### III. ANALYSIS OF RASA

All of the estimated  $\hat{V}^{K_i}$  and  $\hat{Q}^{K_i}$ -values in the current section refer to the values at the **Exit** in the algorithm. The following lemma provides a probability bound on the estimate of the  $Q^*$ -value relative to the true  $Q^*$ -value when the estimate of the  $Q^*$ -value is obtained under the assumption that the optimal value for the remaining horizon is known (so that the recursive call is not required).

*Lemma 3.1:* (cf. [5, Lemma 3.1]) Given  $\delta \in (0, 1)$  and positive integer  $N$  such that  $6 \leq N < \infty$ , consider running the one-stage nonrecursive RASA obtained by replacing (1) by

$$M_i^{K_i}(x, a) \leftarrow M_i^{K_i}(x, a) + R(x, a, w_k) + V_{i+1}^*(f(x, a, w_k)) \quad (4)$$

with  $k > \bar{\lambda}(N, \delta)$  and  $0 < \mu_i < \bar{\mu}_i(N, \delta)$ , where

$$\bar{\lambda}(N, \delta) = \left\lceil \frac{2N}{\ln l} \ln \left[ \frac{Nl}{\ln l} \left( \frac{N}{\delta} \right)^{\frac{1}{N}} \right] \right\rceil, \quad \bar{\mu}_i(N, \delta) = 1 - 2^{-1/\bar{\lambda}(N, \delta)}, \quad \text{and} \quad l = \frac{2|A|}{2|A| - 1}.$$

Then for each action  $a \in A$ , we have

$$P \left( \sum_{j=0}^k I\{a(j) = a\} \leq N \right) < \delta.$$

*Theorem 3.1:* Let  $\{X_i, i = 1, 2, \dots\}$  be a sequence of i.i.d. non-negative uniformly bounded random variables, with  $0 \leq X_i \leq D$  and  $E[X_i] = \mu \forall i$ , and let  $M \in \mathcal{Z}^+$  be a positive integer-valued random variable bounded by  $L$ . If the event  $\{M = n\}$  is independent of  $\{X_{n+1}, X_{n+2}, \dots\}$ , then for any given  $\epsilon > 0$  and  $n \in \mathcal{Z}^+$ , we have

$$P \left( \left| \frac{1}{M} \sum_{i=1}^M X_i - \mu \right| \geq \epsilon, M \geq n \right) \leq 2e^{-n \left( \frac{D+\epsilon}{D} \ln \frac{D+\epsilon}{D} - \frac{\epsilon}{D} \right)}.$$

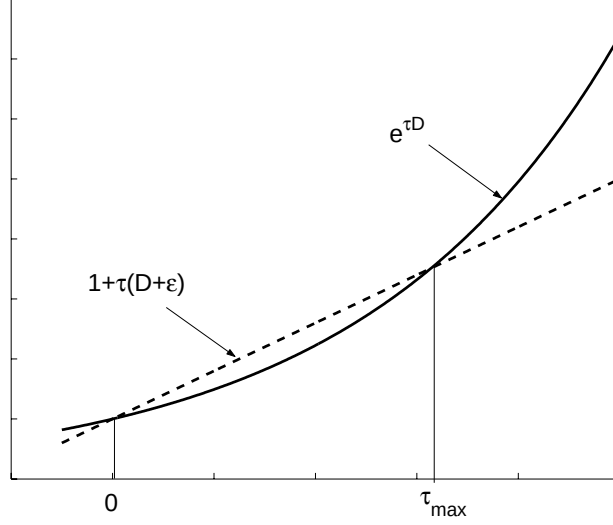


Fig. 1. A sketch of the function  $f_1(\tau) = e^{\tau D}$  and the function  $f_2(\tau) = 1 + \tau(D + \epsilon)$ .

*Proof:* Define  $\Lambda_D(\tau) := \frac{e^{D\tau} - 1 - \tau D}{D^2}$ , and let  $\tau_{\max}$  be a constant satisfying  $\tau_{\max} \neq 0$  and  $1 + (D + \epsilon)\tau_{\max} - e^{D\tau_{\max}} = 0$  (see Figure 1).

Let  $Y_k = \sum_{i=1}^k (X_i - \mu)$ . It is easy to see that the sequence  $\{Y_k\}$  forms a martingale w.r.t  $\{\mathcal{F}_k\}$ , where  $\mathcal{F}_j$  is the  $\sigma$ -field generated by  $\{Y_1, \dots, Y_j\}$ . Therefore, for any  $\tau > 0$ ,

$$\begin{aligned} P\left(\frac{1}{M} \sum_{i=1}^M X_i - \mu \geq \epsilon, M \geq n\right) &= P(Y_M \geq M\epsilon, M \geq n) \\ &= P(\tau Y_M - \Lambda_D(\tau) \langle Y \rangle_M \geq \tau M\epsilon - \Lambda_D(\tau) \langle Y \rangle_M, M \geq n), \end{aligned}$$

where

$$\langle Y \rangle_n = \sum_{j=1}^n E[(\Delta Y_j)^2 | \mathcal{F}_{j-1}], \quad \Delta Y_j = Y_j - Y_{j-1}.$$

Now for any  $\tau \in (0, \tau_{\max})$ , and for any  $n_1 \geq n_0$ , where  $n_0, n_1 \in \mathbb{Z}^+$ , we have

$$\begin{aligned} \tau(n_1 - n_0)\epsilon &\geq \frac{e^{D\tau} - 1 - \tau D}{D^2} (n_1 - n_0) D^2 \\ &\geq \Lambda_D(\tau) \left[ \sum_{j=1}^{n_1} E[(\Delta Y_j)^2 | \mathcal{F}_{j-1}] - \sum_{j=1}^{n_0} E[(\Delta Y_j)^2 | \mathcal{F}_{j-1}] \right], \end{aligned}$$

which implies that

$$\tau n_1 \epsilon - \Lambda_D(\tau) \langle Y \rangle_{n_1} \geq \tau n_0 \epsilon - \Lambda_D(\tau) \langle Y \rangle_{n_0}, \quad \forall \tau \in (0, \tau_{\max}).$$

Thus for all  $\tau \in (0, \tau_{\max})$ ,

$$\begin{aligned}
P\left(\frac{1}{M} \sum_{i=1}^M X_i - \mu \geq \epsilon, M \geq n\right) &\leq P(\tau Y_M - \Lambda_D(\tau) \langle Y \rangle_M \geq \tau n \epsilon - \Lambda_D(\tau) \langle Y \rangle_n, M \geq n) \\
&\leq P(\tau Y_M - \Lambda_D(\tau) \langle Y \rangle_M \geq \tau n \epsilon - \Lambda_D(\tau) n D^2, M \geq n) \\
&= P(e^{\tau Y_M - \Lambda_D(\tau) \langle Y \rangle_M} \geq e^{\tau n \epsilon - n \Lambda_D(\tau) D^2}, M \geq n). \tag{5}
\end{aligned}$$

It can be shown that (cf. e.g., Lemma 1 in [7], pp.505) the sequence  $\{Z_t(\tau) = e^{\tau Y_t - \Lambda_D(\tau) \langle Y \rangle_t}, t \geq 1\}$  with  $Z_0(\tau) = 1$  forms a non-negative supermartingale. It follows that

$$\begin{aligned}
(5) &\leq P(e^{\tau Y_M - \Lambda_D(\tau) \langle Y \rangle_M} \geq e^{\tau n \epsilon - n \Lambda_D(\tau) D^2}) \\
&\leq P\left(\sup_{0 \leq t \leq L} Z_t(\tau) \geq e^{\tau n \epsilon - n \Lambda_D(\tau) D^2}\right) \\
&\leq \frac{E[Z_0(\tau)]}{e^{\tau n \epsilon - n \Lambda_D(\tau) D^2}} \text{ by maximal inequality for supermartingales [7]} \\
&= e^{-n(\tau \epsilon - \Lambda_D(\tau) D^2)}.
\end{aligned}$$

By using a similar argument, we can also show that

$$P\left(\frac{1}{M} \sum_{i=1}^M X_i - \mu \leq -\epsilon, M \geq n\right) \leq e^{-n(\tau \epsilon - \Lambda_D(\tau) D^2)}. \tag{6}$$

Thus by combining (5) and (6), we have

$$P\left(\left|\frac{1}{M} \sum_{i=1}^M X_i - \mu\right| \geq \epsilon, M \geq n\right) \leq 2e^{-n(\tau \epsilon - \Lambda_D(\tau) D^2)}. \tag{7}$$

Finally, we optimize the right-hand-side of (7) over  $\tau$ . It is easy to verify that the optimal  $\tau^*$  is given by  $\tau^* = \frac{1}{D} \ln \frac{D+\epsilon}{D} \in (0, \tau_{\max})$  and

$$\tau^* \epsilon - \Lambda_D(\tau^*) D^2 = \frac{D+\epsilon}{D} \ln \frac{D+\epsilon}{D} - \frac{\epsilon}{D} > 0.$$

Hence Theorem 3.1 follows. ■

*Lemma 3.2:* Consider the nonrecursive RASA obtained by replacing (1) by

$$M_i^{K_i}(x, a) \leftarrow M_i^{K_i}(x, a) + R(x, a, w_k) + V_{i+1}^*(f(x, a, w_k)). \tag{8}$$

Assume  $K_i > \lambda(\epsilon, \delta)$  and  $0 < \mu_i < \mu_i^*(\epsilon, \delta)$ , where

$$\lambda(\epsilon, \delta) = \left\lceil \frac{2M_{\epsilon, \delta}}{\ln l} \ln \left[ \frac{l M_{\epsilon, \delta}}{\ln l} \left( \frac{2M_{\epsilon, \delta}}{\delta} \right)^{\frac{1}{M_{\epsilon, \delta}}} \right] \right\rceil \tag{9}$$

with  $M_{\epsilon, \delta} = \max\{6, \lceil \frac{R_{\max} H \ln(4/\delta)}{(R_{\max} H + \epsilon) \ln((R_{\max} H + \epsilon)/R_{\max} H) - \epsilon} \rceil\}$ ,  $l = 2|A|/(2|A| - 1)$ , and  $\mu_i^*(\epsilon, \delta) = 1 - 2^{-\frac{1}{K_i}}$ . Consider a fixed  $i$ ,  $x \in X$ ,  $\epsilon > 0$ , and  $\delta \in (0, 1)$ . Then, for all  $a \in A$  at the **Exit** step,

$$P\left(|\hat{Q}_i^{K_i}(x, a) - Q_i^*(x, a)| \geq \epsilon\right) < \delta.$$

*Proof:* For any action  $a \in A$ , let  $I_j(a)$  be the iteration at which action  $a$  is chosen for the  $j$ th time, let  $\hat{Q}_{i,k}^{K_i}(x, a)$  be the current estimate of  $Q_i^*(x, a)$  at the  $k$ th iteration, and let  $N_a^{i,k}(x)$  be the number of times action  $a$  is sampled up to the  $k$ th iteration at  $x$ , i.e.,  $N_a^{i,k}(x) = \sum_{j=0}^k I\{a(j) = a\}$ . By RASA, the estimation  $\hat{Q}_{i,k}^{K_i}(x, a)$  is given by (cf. (8))

$$\hat{Q}_{i,k}^{K_i}(x, a) = \frac{1}{N_a^{i,k}(x)} \sum_{j=1}^{N_a^{i,k}(x)} (R(x, a, w_{I_j(a)}) + V_{i+1}^*(f(x, a, w_{I_j(a)}))). \quad (10)$$

Since the sequence of random variables  $\{w_{I_j(a)}, j \geq 1\}$  are i.i.d., a straightforward application of Theorem 3.1 yields

$$P\left(|\hat{Q}_{i,k}^{K_i}(x, a) - Q_i^*(x, a)| \geq \epsilon, N_a^{i,k}(x) \geq N\right) \leq 2e^{-N\left(\frac{R_{\max} H + \epsilon}{R_{\max} H} \ln \frac{R_{\max} H + \epsilon}{R_{\max} H} - \frac{\epsilon}{R_{\max} H}\right)}. \quad (11)$$

Define the events

$$\mathcal{A}_k = \{|\hat{Q}_{i,k}^{K_i}(x, a) - Q_i^*(x, a)| \geq \epsilon\} \quad \text{and} \quad \mathcal{B}_k = \{N_a^{i,k}(x) \geq N\}.$$

By the law of total probability,

$$P(\mathcal{A}_k) = P(\mathcal{A}_k \cap \mathcal{B}_k) + P(\mathcal{A}_k | \mathcal{B}_k^c) P(\mathcal{B}_k^c) \leq P(\mathcal{A}_k \cap \mathcal{B}_k) + P(\mathcal{B}_k^c).$$

Taking  $N = \left\lceil \frac{R_{\max} H \ln(4/\delta)}{(R_{\max} H + \epsilon) \ln((R_{\max} H + \epsilon)/R_{\max} H) - \epsilon} \right\rceil$ , we get from (11) that  $P(\mathcal{A}_k \cap \mathcal{B}_k) \leq \frac{\delta}{2}$ . On the other hand, by Lemma 3.1

$$P(\mathcal{B}_k^c) = P(N_a^{i,k}(x) < N) < \frac{\delta}{2} \quad \text{for } k > \bar{\lambda}(N, \delta/2) \text{ and } 0 < \mu_i < 1 - 2^{-1/\bar{\lambda}(N, \delta/2)}.$$

Therefore  $P(\mathcal{A}_{K_i}) = P\left(|\hat{Q}_i^{K_i}(x, a) - Q_i^*(x, a)| \geq \epsilon\right) < \delta$  for  $K_i > \lambda(\epsilon, \delta)$  and  $0 < \mu_i < \mu_i^*(\epsilon, \delta)$ , where

$$\begin{aligned} \lambda(\epsilon, \delta) &= \left\lceil \frac{2M_{\epsilon, \delta}}{\ln l} \ln \left[ \frac{lM_{\epsilon, \delta}}{\ln l} \left( \frac{2M_{\epsilon, \delta}}{\delta} \right)^{\frac{1}{M_{\epsilon, \delta}}} \right] \right\rceil \\ M_{\epsilon, \delta} &= \max \left\{ 6, \left\lceil \frac{R_{\max} H \ln(4/\delta)}{(R_{\max} H + \epsilon) \ln((R_{\max} H + \epsilon)/R_{\max} H) - \epsilon} \right\rceil \right\} \\ \mu_i^*(\epsilon, \delta) &= 1 - 2^{-\frac{1}{K_i}} < 1 - 2^{-\frac{1}{\lambda(\epsilon, \delta)}}. \end{aligned}$$

Since  $a \in A$  is arbitrary, the proof is complete. ■

We now make an assumption *for the purpose of the analysis*. The assumption states that at each stage, the optimal action is unique at each state. In other words, the given MDP has a unique optimal policy. We will give a remark on this at the concluding section.

*Assumption 3.1:* For all  $x \in X$  and  $i = 0, 1, \dots, H - 1$ ,

$$\theta_i(x) := Q_i^*(x, a^*) - \max_{a \neq a^*} Q_i^*(x, a) > 0,$$

where  $V_i^*(x) = Q_i^*(x, a^*)$ .

Define  $\theta := \inf_{x \in X, i=0, \dots, H-1} \theta_i(x)$ . Given  $\delta_i \in (0, 1), i = 0, \dots, H - 1$ , define

$$\rho := (1 - \delta_0) \prod_{i=1}^{H-1} (1 - \delta_i)^{\prod_{j=1}^i K_j}. \quad (12)$$

*Lemma 3.3:* Assume that Assumption 3.1 holds. Select  $K_i > \lambda(\frac{\theta}{2^{i+2}}, \delta_i)$  and  $0 < \mu_i < \mu_i^* = 1 - 2^{-\frac{1}{K_i}}$  for a given  $\delta_i \in (0, 1), i = 0, \dots, H - 1$ . Then under RASA with  $\rho$  in (12).

$$P \left( \left| \hat{V}_0^{K_0}(x_0) - V_0^*(x_0) \right| > \frac{\theta}{2} \right) < 1 - \rho.$$

*Proof:* Let  $X_s^i$  be the set of sampled states in  $X$  by the algorithm at stage  $i$ . Suppose for a moment that for all  $x \in X_s^{i+1}$ , with some  $K_{i+1}, \mu_{i+1}$ , and a given  $\delta_{i+1} \in (0, 1)$ ,

$$P \left( \left| \hat{V}_{i+1}^{K_{i+1}}(x) - V_{i+1}^*(x) \right| > \frac{\theta}{2^{i+2}} \right) < \delta_{i+1}. \quad (13)$$

Consider for  $x \in X_s^i$ ,

$$\tilde{Q}_i^{K_i}(x, a) = \frac{1}{N_a^i(x)} \sum_{j=1}^{N_a^i(x)} R(x, a, w_j^a) + V_{i+1}^*(f(x, a, w_j^a)),$$

where  $\{w_j^a\}, j = 1, \dots, N_a^i(x)$  refers to the sampled random number sequence for the sample execution of the action  $a$  in the algorithm. We have that for any sampled  $x \in X_s^i$  at stage  $i$ ,

$$\hat{Q}_i^{K_i}(x, a) - \tilde{Q}_i^{K_i}(x, a) = \frac{1}{N_a^i(x)} \sum_{j=1}^{N_a^i(x)} \left( \hat{V}_{i+1}^{K_{i+1}}(f(x, a, w_j^a)) - V_{i+1}^*(f(x, a, w_j^a)) \right).$$

Then under the supposition that (13) holds, for all  $a \in A$  at any sampled  $x \in X_s^i$  at stage  $i$ ,

$$P \left( \left| \hat{Q}_i^{K_i}(x, a) - \tilde{Q}_i^{K_i}(x, a) \right| \leq \frac{\theta}{2^{i+2}} \right) \geq (1 - \delta_{i+1})^{N_a^i(x)} \geq (1 - \delta_{i+1})^{K_{i+1}}. \quad (14)$$

This is because if for all of the next states sampled by taking  $a$ ,  $|V_{i+1}^{K_{i+1}}(f(x, a, w_j^a)) - V_{i+1}^*(f(x, a, w_j^a))| \leq \epsilon$  for  $\epsilon > 0$ , then

$$\frac{1}{N_a^i(x)} \sum_{j=1}^{N_a^i(x)} \left| V_{i+1}^{K_{i+1}}(f(x, a, w_j^a)) - V_{i+1}^*(f(x, a, w_j^a)) \right| \leq \epsilon,$$

which further implies

$$\frac{1}{N_a^i(x)} \left| \sum_{j=1}^{N_a^i(x)} V_{i+1}^{K_{i+1}}(f(x, a, w_j^a)) - V_{i+1}^*(f(x, a, w_j^a)) \right| \leq \epsilon,$$

and therefore

$$P \left( |\hat{Q}_i^{K_i}(x, a) - \tilde{Q}_i^{K_i}(x, a)| \leq \epsilon \right) \geq \prod_{j=1}^{N_a^i(x)} P \left( |V_{i+1}^{K_{i+1}}(f(x, a, w_j^a)) - V_{i+1}^*(f(x, a, w_j^a))| \leq \epsilon \right).$$

From Lemma 3.2, for all  $a \in A$ , with  $K_i > \lambda(\frac{\theta}{2^{i+2}}, \delta_i)$  and  $\mu_i \in (0, 1 - 2^{-\frac{1}{K_i}})$  for  $\delta_i \in (0, 1)$ ,

$$P \left( \left| \tilde{Q}_i^{K_i}(x, a) - Q_i^*(x, a) \right| > \frac{\theta}{2^{i+2}} \right) < \delta_i, x \in X_s^i. \quad (15)$$

Combining (14) and (15),

$$P \left( \left| \hat{Q}_i^{K_i}(x, a) - Q_i^*(x, a) \right| \leq \frac{\theta}{2^{i+2}} + \frac{\theta}{2^{i+2}} \right) \geq (1 - \delta_i)(1 - \delta_{i+1})^{K_{i+1}},$$

and this yields that under the supposition of (13), for any  $x \in X_s^i$ ,

$$P \left( |\hat{Q}_i^{K_i}(x, a) - Q_i^*(x, a)| \leq \frac{\theta}{2^{i+1}} \right) \geq (1 - \delta_i)(1 - \delta_{i+1})^{K_{i+1}}.$$

Define the event

$$E(K_i) = \left\{ \sup_{j \geq K_i} \max_{a \in A} \left| \hat{Q}_i^j(x, a) - Q_i^*(x, a) \right| < \frac{\theta}{2} \right\}. \quad (16)$$

If the event  $E(K_i)$  occurs at the **Exit** step, then from the definition of  $\theta$  and the event  $E(K_i)$ ,  $\hat{Q}_i^{K_i}(x, a^*) > \hat{Q}_i^{K_i}(x, a)$  for all  $a \neq a^*$  with  $a^* = \arg \max_{a \in A} Q_i^*(x, a)$  (Refer the proof of Theorem 3.1 in [5]). Because from the definition of  $\hat{V}_i^{K_i}(x)$ ,  $\hat{V}_i^{K_i}(x) = \max_{a \in A} \hat{Q}_i^{K_i}(x, a) = \hat{Q}_i^{K_i}(x, a^*)$  and  $V_i^*(x) = Q_i^*(x, a^*)$ , with our choice of  $K_i > \lambda(\frac{\theta}{2^{i+2}}, \delta_i)$  and  $\mu_i \in (0, 1 - 2^{-\frac{1}{K_i}})$ ,

$$P \left( \left| \hat{V}_i^{K_i}(x) - V_i^*(x) \right| > \frac{\theta}{2^{i+1}} \right) < 1 - (1 - \delta_i)(1 - \delta_{i+1})^{K_{i+1}}$$

if for all  $x \in X_s^{i+1}$ , with some  $K_{i+1}, \mu_{i+1}$ , and a given  $\delta_{i+1} \in (0, 1)$ ,

$$P \left( \left| \hat{V}_{i+1}^{K_{i+1}}(x) - V_{i+1}^*(x) \right| > \frac{\theta}{2^{i+2}} \right) < \delta_{i+1}.$$

We now apply an inductive argument: because  $\hat{V}_H^{K_H}(x) = V_H^*(x) = 0, x \in X$ , with  $K_{H-1} > \lambda(\frac{\theta}{2^{H+1}}, \delta_{H-1}) \geq \lambda(\frac{\theta}{2^H}, \delta_{H-1})$  and  $\mu_{H-1} \in (0, 1 - 2^{-\frac{1}{K_{H-1}}})$ ,

$$P \left( \left| \hat{V}_{H-1}^{K_{H-1}}(x) - V_{H-1}^*(x) \right| > \frac{\theta}{2^H} \right) < \delta_{H-1}, x \in X_s^{H-1}.$$

It follows that with  $K_{H-2} > \lambda(\frac{\theta}{2^H}, \delta_{H-2})$  and  $\mu_{H-2} \in (0, 1 - 2^{-\frac{1}{K_{H-2}}})$ ,

$$P\left(\left|\hat{V}_{H-2}^{K_{H-2}}(x) - V_{H-2}^*(x)\right| > \frac{\theta}{2^{H-1}}\right) < 1 - (1 - \delta_{H-2})(1 - \delta_{H-1})^{K_{H-1}}, x \in X_s^{H-2}$$

and further follows that with  $K_{H-3} > \lambda(\frac{\theta}{2^{H-1}}, \delta_{H-3})$  and  $\mu_{H-3} \in (0, 1 - 2^{-\frac{1}{K_{H-3}}})$ ,

$$P\left(\left|\hat{V}_{H-3}^{K_{H-3}}(x) - V_{H-3}^*(x)\right| > \frac{\theta}{2^{H-2}}\right) < 1 - (1 - \delta_{H-3})(1 - \delta_{H-2})^{K_{H-2}}(1 - \delta_{H-1})^{K_{H-2}K_{H-1}}$$

for  $x \in X_s^{H-3}$ . Continuing this way, we have that

$$P\left(\left|\hat{V}_1^{K_1}(x) - V_1^*(x)\right| > \frac{\theta}{2^2}\right) < 1 - (1 - \delta_1)(1 - \delta_2)^{K_2}(1 - \delta_3)^{K_2K_3} \times \dots \times (1 - \delta_{H-1})^{K_2 \dots K_{H-1}}$$

for  $x \in X_s^1$ . Finally, with  $K_0 > \lambda(\frac{\theta}{4}, \delta_0)$  and  $\mu_0 \in (0, 1 - 2^{-\frac{1}{K_0}})$ ,

$$P\left(\left|\hat{V}_0^{K_0}(x_0) - V_0^*(x_0)\right| > \frac{\theta}{2}\right) < 1 - (1 - \delta_0)(1 - \delta_1)^{K_1} \times \dots \times (1 - \delta_{H-1})^{K_1 \dots K_{H-1}},$$

which completes the proof. ■

**Theorem 3.2:** Assume that Assumption 3.1 holds. Given  $\delta_i \in (0, 1), i = 0, \dots, H-1$ , select  $K_i > \lambda(\frac{\theta}{2^{i+2}}, \delta_i)$  and  $0 < \mu_i < \mu_i^* = 1 - 2^{-\frac{1}{K_i}}, i = 1, \dots, H-1$ . Then under RASA with  $\rho$  in (12), for all  $\epsilon \in (0, 1)$ ,

$$P(P_{x_0}(K_0)(a^*) > 1 - \epsilon) > 1 - \rho,$$

where  $a^* = \arg \max_{a \in A} Q_0^*(x_0, a)$  and  $K_0 > K_0^*$ , where

$$K_0^* = \kappa + \left\lceil \frac{\ln \frac{1}{\epsilon}}{\ln \frac{1}{1 - \mu_0^*}} \right\rceil,$$

with  $\kappa > \lambda(\frac{\theta}{4}, \delta_0)$  and  $0 < \mu_0 < \mu_0^* = 1 - 2^{-\frac{1}{\kappa}}$ .

*Proof:* Define the event  $E'(K_0) = \{P_{x_0}(K_0)(a^*) > 1 - \epsilon\}$  where  $a^* = \arg \max_{a \in A} Q_0^*(x_0, a)$ . Then,

$$P(P_{x_0}(K_0^*)(a^*) > 1 - \epsilon) = P(E'(K_0^*)) \geq P(E'(K_0^*)|E(\kappa))P(E(\kappa)),$$

where the event  $E$  is given in (16) with  $i = 0$ .

From the selection of  $\kappa > \lambda(\frac{\theta}{4}, \delta_0)$  with  $K_i > \lambda(\frac{\theta}{2^{i+2}}, \delta_i), i = 1, \dots, H-1$  and  $\mu_i$ 's for  $\delta_i \in (0, 1)$ ,  $P(E(\kappa)) \geq 1 - \rho$  by Lemma 3.3. With reasoning similar to that in the proof of Theorem 3.1 in [5],  $P(E'(l + \kappa)|E(\kappa)) = 1$  if  $l + \kappa > \lambda(\frac{\theta}{2}, \delta_0) + \lceil \frac{\ln \frac{1}{\epsilon}}{\ln \frac{1}{1 - \mu_0^*}} \rceil$ . ■

Based on the proof of Lemma 3.3, the following result follows immediately. We skip the details.

**Theorem 3.3:** Assume that Assumption 3.1 holds. Given  $\delta_i \in (0, 1), i = 0, \dots, H - 1$  and  $\epsilon \in (0, \theta]$ , select  $K_i > \lambda(\frac{\epsilon}{2^{i+2}}, \delta_i), 0 < \mu_i < \mu_i^* = 1 - 2^{-\frac{1}{K_i}}, i = 0, \dots, H - 1$ . Then under RASA with  $\rho$  in (12),

$$P\left(\left|\hat{V}_0^{K_0}(x_0) - V_0^*(x_0)\right| > \frac{\epsilon}{2}\right) < 1 - \rho.$$

#### IV. CONCLUDING REMARKS

From the statements of Lemma 3.3, Theorems 3.2 and 3.3, the performance of RASA depends on the value of  $\theta$ . If  $\theta_i(x)$  is very small or even 0 (failing to satisfy Assumption 3.1) for some  $x \in X$ , RASA will have a hard time distinguishing between the optimal action and the second best action or multiple optimal actions *if  $x$  is in the sampled tree of RASA*. In general, the larger  $\theta$  is, the more effective the algorithm will be (the smaller the sampling complexity). Therefore, in the actual implementation of RASA, if multiple actions' performances are very close after “enough” iterations in **Loop**, it would be advisable to keep only one action among the competitive actions (transferring the probability mass). The parameter  $\theta$  can thus be viewed as a measure of problem difficulty.

Furthermore, to achieve a certain approximation guarantee at the root level of the sampled tree (i.e., the quality of  $\hat{V}_0^{K_0}(x_0)$ ), we need a *geometric* increase in the accuracies of the optimal reward-to-go values for the sampled states at the lower levels, making it necessary that the total number of samples at the lower levels increase geometrically ( $K_i$  depends on  $2^{i+2}/\theta$ ). This is because the estimate error of  $V_i^*(x_i)$  for some  $x_i \in X$  affects the estimate of the sampled states in the higher levels in a recursive manner (the error in a level “adds up recursively”).

However, the probability bounds in Theorem 3.2 and 3.3 are obtained with coarse estimation of various parameters/terms. For example, we used the worst case values of  $\theta_i(x), x \in X, i = 0, \dots, H - 1$  and  $(R_{\max}H)^2$  for bounding  $\sup_{x \in X} V_i^*(x), i = 0, \dots, H - 1$ , and used conservative bounds in (14) and in relating the probability bounds for the estimates at the two adjacent levels. Considering this, the performance of RASA should probably be more effective in practice than the analysis indicates here.

## REFERENCES

- [1] H. S. Chang, M. C. Fu, J. Hu, and S. I. Marcus, "An adaptive sampling algorithm for solving Markov decision processes," *Operations Research*, vol. 53, no. 1, pp. 126–139, 2005.
- [2] R. Jain and P. Varaiya, "Simulation-based uniform value function estimates of discounted and average-reward MDPs," in *Proc. of the 43rd IEEE Conf. on Decision and Control*, pp. 4405–4410, 2004.
- [3] A. S. Poznyak, K. Najim, and E. Gomez-Ramirez, *Self-Learning Control of Finite Markov Chains*, Marcel Dekker, Inc. New York, 2000.
- [4] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.
- [5] K. Rajaraman and P. S. Sastry, "Finite time analysis of the pursuit algorithm for learning automata," *IEEE Trans. on Systems, Man, and Cybernetics*, Part B, vol. 26, no. 4, pp. 590–598, 1996.
- [6] G. Santharam and P. S. Sastry, "A reinforcement learning neural network for adaptive control of Markov chains," *IEEE Trans. on Systems, Man, and Cybernetics*, Part A, vol. 27, no. 5, pp. 588–600, 1997.
- [7] A.N. Shiryaev, *Probability*, Second Edition, Springer-Verlag, New York, 1995.
- [8] M. A. L. Thathachar and P. S. Sastry, "Varieties of learning automata: an overview," *IEEE Trans. on Systems, Man, and Cybernetics*, Part B, vol. 32, no. 6, pp. 711–722, 2002.
- [9] R. M. Wheeler, Jr. and K. S. Narendra, "Decentralized learning in finite Markov chains," *IEEE Trans. on Automatic Control*, vol. AC-31, no. 6, pp. 519–526, 1986.