Dynamically Identifying Regenerative Cycles in Simulation-Based Optimization Algorithms for Markov Chains *
Technical Report SIE-020003

Enrique Campos-Náñez and Stephen D. Patek
Department of Systems and Information Engineering
University of Virginia
{ec3z,patek}@virginia.edu

June 18, 2003

Abstract

Simulation-based algorithms for maximizing the average reward of a parameterized Markov chain often rely upon the existence of a state which is recurrent for all choices of parameter values. For example, in the "batch" simulation-based algorithm of Marbach and Tsitsiklis [28], a given recurrent state i^* is used to mark the onset of regenerative cycles within a simulation of the process, and the data collected in each cycle give rise to asymptotically unbiased estimates of the gradient of the average reward of the process. The question of *which* recurrent state should serve as i^* is a very important practical consideration in applications. Even when all states of the process are recurrent, some states tend to be visited more often than others, and lengthy renewal cycles tend to result in high variance estimates of the gradient. An appropriate choice of i^* is especially difficult when the steady state distribution of the process depends strongly on the parameters of the underlying Markov chain. To address this difficulty, we analyze a recently-proposed mechanism for adjusting i^* dynamically (i^*-adaptation [8]) as applied to the "batch" simulation-based algorithm of [28]. We show that the desirable convergence properties of the original algorithm are retained with i^*-adaptation, namely the almost sure convergence of the parameter vector to a critical point, and we present an academic example which illustrates that i^*-adaptation may significantly expand the range of applications of the original methodology.

1 Introduction

Simulation-based algorithms for tuning the parameters of Markov chains have generated a lot of interest recently [1, 12, 14, 15, 16, 21, 22, 27, 28, 32, 33]. In this paper, we focus on irreducible Markov chains with average reward criteria,

*This work is supported in part by a grant from the National Science Foundation (ECS-9875688 (CAREER)) and by a scholarship from Consejo Nacional de Ciencia y Tecnología (CONACYT-68428).

and we address algorithms, such as the "batch" simulation-based algorithm of Marbach and Tsitsiklis [28], that (1) exploit the regenerative structure of the problem in computing estimates of the gradient of the objective function and (2) update recursively the parameters of the Markov chain in the direction of the estimate. Typically, in algorithms of this sort, the variance of the gradient estimate grows with the length of the underlying regenerative cycles of the process, so that a very important practical consideration is the selection of the state i^* used to mark renewal epochs. Choosing a state that is infrequently visited can make it very difficult to set algorithmic parameters, such as the stepsize rule, and typically results in slow convergence, rendering the algorithm impractical even though in theory it converges with probability one to a point where the gradient of the objective function is zero. We address these performance issues by analyzing an adaptive rule, i^*-adaptation, for dynamically adjusting i^* as the batch simulation-based algorithm of [28] evolves. The i^*-adaptation rule was introduced and experimentally tested in [8]. In this paper, we give a formal proof that the modified version of the batch simulation-based procedure (with i^*-adaptation) retains the desirable property of the original algorithm that the parameter vector of the Markov chain converges almost surely to a critical point where the gradient of the objective function is zero.

To set the stage for this paper more formally, we consider a discrete-time Markov chain with finite state space $S = \{1, 2, \ldots, N\}$, where the transition probabilities and transition rewards, controlled by a set of parameters $\theta \in \Re^K$, are denoted by $p_{ij}(\theta)$, and $g_i(\theta)$, respectively. Let $P(\theta) = [p_{ij}(\theta)]_{ij}$, and let $\mathcal{P} = \{P(\theta)|\theta \in \Re^K\}$. We make the following structural assumptions.

Assumption 1 *The matrices $P \in \bar{\mathcal{P}}$ are irreducible and aperiodic, where $\bar{\mathcal{P}}$ is the closure of $\mathcal{P}$.*

Assumption 2 *For every $i, j \in S$ the functions $p_{ij}(\theta)$ and $g_i(\theta)$ are bounded, twice differentiable, and have a bounded first and second derivatives for all $\theta \in \Re^K$.*

Assumption 3 *For every i and j, there exists a bounded function $L_{ij}(\theta)$ such that*

$$\nabla p_{ij}(\theta) = p_{ij}(\theta) L_{ij}(\theta), \quad \forall \theta \in \Re^K.$$

Our objective is to maximize the average reward of the process, defined by

$$\lambda(\theta) = \lim_{T \to \infty} \frac{1}{T} E_\theta \left[\sum_{k=0}^{T} g_{i_k}(\theta) \middle| i_0 = i \right].$$

Note that under Assumption 1 the average reward objective $\lambda(\theta)$ is well-defined and independent of the initial state, and can be calculated by $\sum_{i \in S} \pi_i(\theta) g_i(\theta)$, where $\{\pi_i(\theta) \ ; \ i \in S\}$ is the steady-state distribution associated with $P(\theta)$. Assumptions 1 and 2 also imply that $\lambda(\theta)$ is twice differentiable, and has bounded first and second derivatives.

In Section 2 of this paper, we review relevant literature, including an extensive review of the batch simulation-based algorithm developed by Marbach and Tsitsiklis in [28]. In Section 3, we discuss some practical issues associated with the batch simulation-based algorithm. Particularly, in considering a simple academic example, we focus on the question of what is an appropriate choice of i^*? In Section 4, we motivate the i^*-adaptation rule for dynamically adjusting the recurrent state i^*, and we formally describe its application to the batch simulation-based algorithm of [28]. In Section 5, under an additional technical assumption, we establish the almost sure convergence of the batch simulation-based algorithm with i^*-adaptation to a critical point. In Section 6, we illustrate the convergence properties of the algorithm for the academic example of Section 3 and make concluding remarks.

2 Background

2.1 Literature Review

Here, we briefly review some of the literature that addresses computational issues faced in optimizing the average-reward of finite-state Markov chains. The classical optimization framework is that of Markov decision processes and dynamic programming [2, 29, 31, 34], where the objective is to compute an optimal policy (i.e. mapping from states to actions). In this context, the well-known "curse of dimensionality" prevents the direct application of dynamic programming techniques in problems with large numbers of states, say more than several thousands. Over the years there have been many proposals for algorithms based on approximate dynamic programming, including reinforcement learning [1, 21, 22, 32, 33] and neuro-dynamic programming [4], where the main computational effort is to obtain first an approximation to the optimal value function of the process and then use the approximation to derive a near-optimal policy. Approximate dynamic programming offers a generic methodology that can be applied in diverse applications, especially in problems where the form of the optimal policy is unknown.

As an alternative approach, it is possible to specify a parameterized class of policies and reduce the computational problem to one of searching the parameter space for extrema. Such an approach, often referred to as "search in policy space," can simplify the problem [20] and is natural in many applications where the structure of the policy is set in advance. In this paper, as in [28], we consider Markov reward processes where the set of policies is parametrized by a set of continuous variables, and where the objective function can be evaluated as a continuous and differentiable function of those parameters. Although it is difficult to obtain a closed form expression for the gradient, it is possible in practice to obtain "noisy" estimates of the gradient and use them to guide a stochastic approximations type search procedure [23, 24, 25, 26, 30].

Estimates of the gradient are often obtained by sampling the function around the current parameter values, as in "finite-

differences" methods [26]. This approach can be computationally expensive. An alternative source of gradient estimates is through the analysis of sample-path realizations under given parameter settings. This was originally proposed by [20] and was further refined in [10, 17, 18, 19] in a methodology called infinitesimal perturbation analysis. If regenerative structure is available, it is possible to obtain unbiased estimates by conditional Monte Carlo techniques, as in [11, 13, 14, 15].

For Markov reward processes, it is possible to further improve the estimation of the gradient of the objective function by means of the so-called likelihood ratio method [9, 15, 16]. In this method, sample paths are observed between visits to a singled out state (i^*), and likelihood information is used to obtain an unbiased estimate of the gradient. A similar approach, which produces an asymptotically unbiased estimate of the gradient after observing just one regenerative cycle, is presented in [28], which we discuss more thoroughly in the following subsection.

2.2 A "Batch" Simulation-Based Algorithm from [28]

In [28], Marbach and Tsitsiklis developed the following method of constructing estimates of $\nabla\lambda(\theta)$. Given $\theta \in \Theta$, along with an estimate $\tilde{\lambda} \in \Re$ of the average reward $\lambda(\theta)$, suppose that $\{i_0 = i^*, i_1, i_2, \ldots, i_T = i^*\}$ is a sample trajectory of the Markov chain under θ, where process is initialized at $i_0 = i^*$ and $T > 0$ is the time index of the first return to i^*. Consider the function

$$F(\theta, \lambda) = \sum_{n=0}^{T-1} \tilde{v}_{i_n}(\theta, \tilde{\lambda}) L_{i_{n-1} i_n}(\theta) + \nabla g_{i_n}(\theta), \tag{1}$$

where

$$\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = \begin{cases} \sum_{k=n}^{T-1} \left(g_{i_k}(\theta) - \tilde{\lambda} \right), & n = 1, \ldots, T-1 \\ 0, & n = 0 \end{cases}$$

is an estimate of the differential reward to return to i^*, and the likelihood functions $L_{ij}(\theta)$ are defined in Assumption 3. (Note that $\tilde{v}_{i^*}(\theta, \tilde{\lambda})$ does not need to be estimated, since $E[\tilde{v}_{i^*}(\theta, \lambda(\theta))] = 0$.) A key result from [28] is that

$$E_\theta[F(\theta, \tilde{\lambda})] = E_\theta[T] \nabla\lambda(\theta) + G(\theta)(\lambda(\theta) - \tilde{\lambda}), \tag{2}$$

where $E_\theta[\cdot]$ denotes the expectation with respect to fixed θ and

$$G(\theta) = E_\theta \left[\sum_{n=1}^{T-1} (T - n) L_{i_{n-1} i_n}(\theta) \right]. \tag{3}$$

From Eqn. (2), we see that except for the error term $G(\theta)(\lambda(\theta) - \tilde{\lambda})$ the expected direction of $F(\theta, \tilde{\lambda})$ is the same as that of the gradient $\nabla\lambda(\theta)$.

As discussed in [28], the gradient estimate of Eqn. (1) can be used in a stochastic approximation optimization procedure by

1. starting with an initial parameter vector θ_0 and average reward estimate $\tilde{\lambda}_0$, then
2. obtaining (e.g. through simulation) a sample trajectory of the Markov chain under θ_0 starting from $i_0 = i^*$ and ending when the system reaches i^* again for the first time, and then
3. evaluating an estimate of the gradient estimate according to Eqn. (1) and computing a new parameter vector θ_1 and average reward estimate $\tilde{\lambda}_1$ using a gradient search type update, and continuing ad infinitum.

To formalize this procedure, let

$$\{i_{t_m} = i^*, i_{t_m+1}, \ldots, i_{t_{m+1}} = i^*\}$$

denote the $(m+1)$-st sample regenerative cycle, realized under θ_m. Let $T_m = t_{m+1} - t_m$ denote the length of the cycle. Motivated from Eqn. (1), let

$$F(\theta_m, \tilde{\lambda}_m) = \sum_{n=t_m}^{t_{m+1}-1} \tilde{v}_{i_n}(\theta_m, \tilde{\lambda}_m) L_{i_{n-1}i_n}(\theta_m) + \nabla g_{i_n}(\theta_m),$$

where

$$\tilde{v}_{i_n}(\theta_m, \tilde{\lambda}_m) = \begin{cases} \sum_{k=n}^{t_{m+1}-1} \left(g_{i_k}(\theta_m) - \tilde{\lambda}_m \right), & n = 1, \ldots, T_m - 1 \\ 0, & n = 0. \end{cases}$$

From [28],

$$E_{\theta_m}[F(\theta_m, \tilde{\lambda}_m)] = E_{\theta_m}[T_m] \nabla \lambda(\theta_m) + G(\theta_m) \left(\lambda(\theta_m) - \tilde{\lambda}_m \right). \tag{4}$$

The parameter vector and average reward estimate are updated as follows.

$$\begin{aligned} \theta_{m+1} &= \theta_m + \gamma_m F(\theta_m, \tilde{\lambda}_m), \\ \tilde{\lambda}_{m+1} &= \tilde{\lambda}_m + \gamma_m \eta \sum_{n=t_m}^{t_{m+1}-1} \left(g_{i_n}(\theta_m) - \tilde{\lambda}_m \right), \end{aligned}$$

where γ_m is a stepsize parameter such that $\sum_m \gamma_m = \infty$, and $\sum_m \gamma_m^2 < \infty$, and $\eta > 0$ is a scale parameter. One of the main results of [28] is that $\nabla \lambda(\theta_m) \to 0$, as $m \to \infty$, with probability 1.[1]

[1] Marbach and Tsitsiklis established this result under a weaker (more general) version of Assumption 1. Whereas our Assumption 1 requires *all* states to be recurrent for all $\theta \in \Theta$, [28] only requires that there exists a state (i^*) which is recurrent under all $\theta \in \Theta$. We use the more restrictive assumption in Section 4, where our i^*-adaptation procedure essentially requires that any state can serve as a candidate value of i^*.

3 The Effect of Lengthy Regenerative Cycles

A very important practical consideration in using Eqn. (1) to estimate the gradient of $\lambda(\theta)$ is that lengthy sample regenerative cycles (i.e. samples $\{i_0, \ldots, i_T\}$ with T large) tend to produce high variance estimates $F(\theta, \tilde{\lambda})$. (Recall that the estimate is constructed as a summation of random variables, one for each stage of the regenerative cycle.) For the simulation-based algorithm of [28], large variances generally make it difficult to select appropriate values for the stepsize rule $\{\gamma_m; m \geq 0\}$ and the scale factor η. The essential tradeoff is between the ability of the optimization procedure to quickly identify extrema (calling for "large" values of γ_m and η) and the ability to average-out noise (calling for "small" values of γ_m and η). Marbach and Tsitsiklis have addressed this issue by introducing some variants on their "batch" method, including (i) an algorithm which allows for premature termination of regenerative cycles (by defining a *set* of terminal states $I^* \supset i^*$) and (ii) an algorithm based on introducing a discount factor (see [27]). Both of these variants introduce significant bias into the gradient estimate, resulting ultimately in the failure to converge to a critical point. In [28], Marbach and Tsitsiklis describe another algorithm based on Eqn. (1) in which updates to the parameter vector are made after *each* state transition. Unfortunately this algorithm requires a bias-correction update upon each transition to i^*, and the problem of large variances associated with lengthy regenerative cycles is only temporarily avoided.

In this paper we address the problem of long regenerative cycles by considering the question: What is an appropriate choice for i^*? Certainly, if all states are recurrent under all $\theta \in \Theta$ (as in Assumption 1), then any state $i \in S$ can serve as i^*. On the other hand, some states tend to be visited more often than others. As a rule of thumb we would choose i^* to the be the state $i \in S$ with the largest steady state probability $\pi_i(\theta)$. Unfortunately, the steady state distribution depends on θ, which changes over time as the optimization procedure evolves, and this rule of thumb leads to a "chicken and the egg" problem where the best choice of i^* depends upon the unknown critical point θ^* to which we will eventually converge.

As an illustration, consider the following numerical example involving a Markov chain with state space $S = \{0, 1, 2, \ldots, N\}$, transition probabilities

$$p_{ij}(\theta) = \begin{cases} \frac{(N-i)\theta}{(N-i)\theta+\mu} & \text{if } j = i+1, \\ \frac{\mu}{(N-i)\theta+\mu} & \text{if } j = i-1, \text{ and } i > 0, \\ 0, & \text{otherwise,} \end{cases}$$

and transition rewards

$$g_i(\theta) = (1-\theta)\frac{(N-i)\theta}{(N-i)\theta+\mu},$$

where $\theta \in (a, b)$ is the control parameter. We consider the case where $N = 100$, $\mu = 25$, $a = .05$ and $b = .95$. Note that, except for the requirement that θ be taken from $\Re$, Assumptions 1–3 hold. Since θ is restricted to an open interval, it can

be parametrized as $\theta(u) = a + (\arctan(u)/\pi + 1/2)\,(b-a)$, with $u \in \Re$. However, in our numerical results we optimize over (a, b) directly, choosing the stepsize γ_m and scale parameter η so that θ_m lies within the open interval. The optimal parameter value (computed off-line) is $\theta^* = .2473$. Figure 1 shows a contour plot of the the steady-state distribution of the Markov chain with state number $i = 0, \ldots, 100$ on the y-axis and parameter $\theta \in (.05, .95)$ on the x-axis. Notice that for $\theta \approx .05$, the states numbered $0, \ldots, 5$ are the most likely states (i.e. the states with the highest steady state probabilities). On the other hand, for $\theta \approx .95$, the states numbered $65, \ldots, 85$ are most likely. Now consider what happens when we apply the batch simulation-based method of [28]. If we set $\theta_0 \approx .95$, then $i^* = 75$ would be reasonable: sample trajectories starting and ending at $i^* = 75$ will be short on average, and low variance estimates of $\lambda(\theta_m)$ will quickly push θ_m toward .2473. Unfortunately, as $\theta_m \to .2473$, the state $i^* = 75$ becomes more and more unlikely. This, in turn, causes the sample regenerative cycles to become longer and longer on average, amplifying the noise observed in the evolution of θ_m. Figure 2 illustrates what the practitioner will typically observe under these circumstances. The figure shows the sample evolution of θ_m for four initial conditions $(\theta_0, i^*) \in \{(.90, 75), (.90, 5), (.1, 75), (.1, 5)\}$ with $\gamma_m = \frac{1}{(1000+m)100}$ and $\eta = 100$. Notice that except for the lucky choice of $i^* = 5$ and $\theta_0 = .1$ the algorithm fails to make much progress toward $\theta^* = .2473$, even after 10^6 simulated state transitions. While Figure 2 only illustrates sample trajectories for a specific example, we have found this behavior to be quite common: (i) the algorithm can get "stuck" simulating long regenerative cycles and/or (ii) when updates do occur they can be quite noisy. Thus, despite the convergence result of [28], the algorithm can fail (in a practical sense) because of the inappropriate choice of i^*. We point out that queueing applications are particularly prone to this type of failure. Specifically, in queuing applications it is often natural to choose i^* to be the "system empty" state, but unfortunately "system empty" can be an infrequently visited state under optimal loading conditions.

4 The i^*-adaptation procedure

To address the problem of lengthy regenerative cycles, we analyze a rule for dynamically adjusting the value of i^* as the simulation-based algorithm evolves. The technique, called i^**-adaptation*, was introduced and experimentally tested in [8] and applies generally to optimization algorithms that use likelihood ratio methods to estimate $\nabla\lambda(\theta)$. The philosophy of i^*-adaptation is that *if, as the simulation based algorithm evolves, the prevailing state i^* becomes infrequently observed, we should identify a better candidate value for i^* and adjust its value as needed.*

While we provide a formal implementation of this philosophy in the sequel (cf. Algorithm 1 below), we first discuss some of the issues that must be addressed in this approach. The basic methodology is to choose an integer $\tau > 0$ and use this as a threshold length of regenerative cycles by (i) terminating (and ignoring) sample regenerative cycles whose lengths

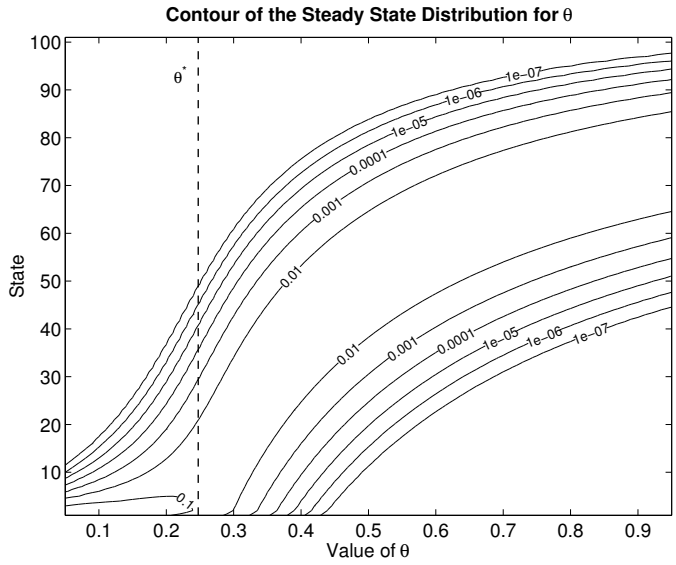

Figure 1: Contours of the steady state distribution for the example of Section 3.

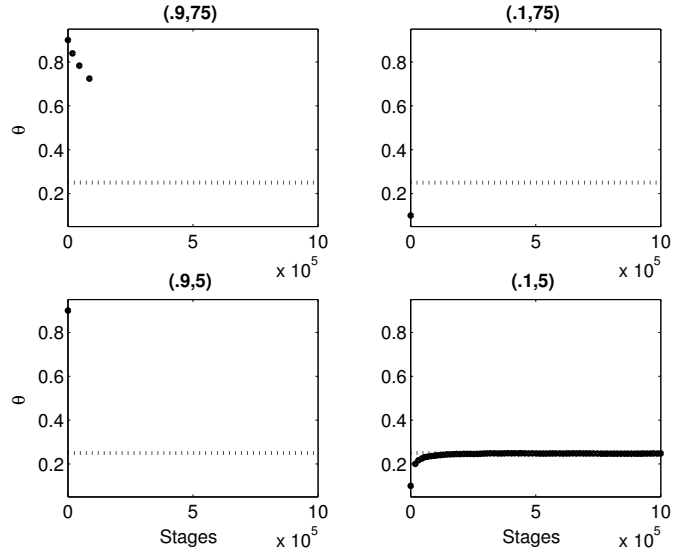

Figure 2: Sample evolution of the batch simulation-based algorithm of [28] for four initial conditions $(\theta_0, i^*) \in \{(.90, 75), (.90, 5), (.1, 75), (.1, 5)\}$. [Note that, except for the case $(.1, 5)$, the algorithm gets stuck simulating long regenerative cycles. For the case $(.90, 5)$ only 2 cycles were observed (very early in the process). For the case $(.1, 75)$ no cycles were observed.].

T exceed τ and (ii) choosing new values for i^* whenever we act to break cycles this way. The first issue to consider in this scheme is: How should we pick new values for i^*? Since all states are recurrent (by Assumption 1), any state $i \in S$ can serve as i^*. However, rather than selecting the new i^* at random, we prefer to choose deliberately a state that is likely to be visited frequently under the prevailing set of parameters θ. To this end, suppose the most recently observed sample regenerative cycle is given by $\{i_0 = i^*, i_1, \ldots, i_T\}$, where $T > \tau$. A reasonable choice for the new value of i^* is the observed state of the system when the threshold τ was surpassed, namely i_τ. (If the threshold τ is large enough, then we may take i_τ as an estimate of the most likely state (i.e. the state with the highest steady-state probability) under θ.) In Algorithm 1 below, we use $i^* \leftarrow i_\tau$ as the rule for selecting the new value for i^* when we break regenerative cycles.

The second issue that must be addressed in implementing the i^*-adaptation philosophy is: how should we select the threshold length τ? As discussed in [8], the answer turns out to be that *we cannot use any fixed value of* τ. A fixed threshold length τ will cause all regenerative cycles of length $T > \tau$ to be ignored as the simulation-based algorithm evolves, and this introduces a bias into the calculations as the estimates of $\nabla\lambda(\theta)$ are averaged over many regenerative cycles. Thus, if we are to enjoy any potential benefit from i^*-adaptation and retain the desirable convergence properties of [28], we are compelled to force the threshold length τ to increase over time. In Algorithm 1 below, we use the rule $\tau \leftarrow \tau + 1$ whenever a regenerative cycle is observed whose length T exceeds τ.

Algorithm 1 (Batch Simulation-Based Algorithm with i^*-Adaptation) *Given scale parameter $\eta > 0$ and stepsizes γ_m such that $\sum_m \gamma_m = \infty$, and $\sum_m \gamma_m^2 < \infty$, recursively compute $\{(\theta_m, i_m^*, \tau_m, t_m) \,:\, m = 0, 1, \ldots\}$ as follows.*

1. **(Initialization)** *Set $\theta_0 \in \Theta$, $\tau_0 > 0$, $i_0^* = i_0$, and $t_0 = 0$.*

2. **(Iteration $\mathbf{m+1}$)** *Simulate the $(m+1)$-st regenerative cycle: $\{i_{t_m}, i_{t_m+1}, \ldots, i_{t_{m+1}}\}$, where $t_{m+1} = t_m + \min\{\tau_m, T_m\}$, with T_m being the random number of transitions for a complete cycle (i.e. ending upon the first return to i_m^*) holding θ_m fixed.*

   (a) **(Broken Cycle)** *If $i_{t_{m+1}} \neq i_m^*$, then set $i_{m+1}^* = i_{t_{m+1}}$, $\tau_{m+1} = \tau_m + 1$, $\theta_{m+1} = \theta_m$, and $\tilde{\lambda}_{m+1} = \tilde{\lambda}_m$. Otherwise,*

   (b) **(Normal Cycle)** *if $i_{t_{m+1}} = i_m^*$, then update the parameter vector and the average reward estimate according to*

$$\theta_{m+1} = \theta_m + \gamma_m F(\theta_m, \tilde{\lambda}_m), \tag{5}$$

$$\tilde{\lambda}_{m+1} = \tilde{\lambda}_m + \eta\gamma_m \sum_{n=t_m}^{t_{m+1}-1} \Big(g_{i_n}(\theta_m) - \tilde{\lambda}_m \Big). \tag{6}$$

   Finally, set $i_{m+1}^* = i_m^*$ and $\tau_{m+1} = \tau_m$.

5 Main Result

In this section we show that with the addition of the i^*-adaptation rule the batch simulation-based algorithm of [28] still converges with probability one to a critical point θ^* where $\nabla\lambda(\theta^*) = 0$. The main analytical issue to resolve is whether the bias introduced by Step 2.(a) of Algorithm 1 becomes asymptotically negligible as the algorithm proceeds. Intuitively, given that $\tau_m \to \infty$, the algorithm behaves asymptotically like the original batch simulation-based algorithm of [28]. Thus, the fact that Algorithm 1 retains the convergence properties of [28] may not be surprising. On the other hand, the question remains whether $\tau_m \to \infty$ *fast enough* for this intuition to hold true. In general, the analysis of Algorithm 1 is complicated by the fact that its structure changes over time as new values of i^* cause the algorithm to interpret regenerative cycles differently as the parameter vector θ evolves.

Our analysis requires the following technical assumption.

Assumption 4 *There exists* $\bar{\rho} > 0$ *such that* $\inf_{\theta, i^*, i \neq i^*} 1 - P_{ii^*}(\theta) \geq \bar{\rho}$.

This assumption is similar in spirit to the "observability" assumption commonly used in the theory of adaptive Markov chains (see e.g. [6, 7], in particular Assumption 1 in [6]). We now state our main result.

Theorem 1 *Under Assumptions 1–4, Algorithm 1 with the stepsize rule* $\gamma_m = a/(b+m)$ *produces a sequence of parameter vectors* θ_m *that converges with probability one to a critical point where the gradient of the objective function is zero, i.e.* $\nabla\lambda(\theta_m) \to 0$*, for all* $a, b > 0$.

The proof of this result is somewhat involved and is given throughout the remainder of this section. Since the general strategy of the proof closely follows the proof of the corresponding result in [28] (cf. Proposition 3), we focus here on the additional arguments that have to be made due to the fact that i^* can be adjusted dynamically as the simulation-based algorithm proceeds. We defer proofs of the intermediate results to the appendices.

To begin, let $r_m = (\theta_m, \tilde{\lambda}_m)$ and note that Algorithm 1 can be rewritten in the form

$$r_{m+1} = r_m + \gamma_m h(r_m, i_m^*) + \varepsilon_m + \varepsilon'_m,$$

where

1. $h(r_m, i_m^*) = E\left[H(r_m, i_m^*) \mid i_{t_m} = i_m^*\right]$ is the expected update assuming that the $(m+1)$-st cycle is complete, i.e.

$$H(r_m, i_m^*) = \begin{bmatrix} F(r_m) \\ \eta \sum_{n=t_m}^{t_m+T_m-1} \left(g_{i_n}(\theta_m) - \tilde{\lambda}_m \right) \end{bmatrix},$$

where $F(r_m) = F(\theta_m, \tilde{\lambda}_m)$, as defined in Eqn. (1) with the additional consideration that i_m^* is used to mark regenerative cycles.

2. $\varepsilon_m = \gamma_m(H(r_m, i_m^*) - h(r_m, i_m^*))$ is the error introduced by the gradient estimate assuming that the $(m+1)$-st cycle is complete, and

3. $\varepsilon'_m = \gamma_m H(r_m, i_m^*)\mathbf{1}_{\{i_{t_{m+1}} \neq i_m^*\}}$ is the bias term that corrects for the possibility that the $(m+1)$-st cycle is incomplete.

The general idea of the proof is to take advantage of the observation (drawn from [28]) that

$$h(r_m, i_m^*) = \begin{bmatrix} E_{\theta_m, i_m^*}[T_m]\nabla\lambda(\theta_m) + G(\theta_m, i_m^*)(\lambda(\theta_m) - \tilde{\lambda}_m) \\ \eta E_{\theta_m, i_m^*}[T](\lambda(\theta_m) - \tilde{\lambda}_m) \end{bmatrix}, \tag{7}$$

where $E_{\theta_m, i_m^*}[T_m]$ is the expected length of a cycle (under θ_m) that starts and ends at i_m^* and $G(\theta_m, i_m^*)$ is defined as in Eqn. (3) with the additional consideration that i_m^* is used to mark regenerative cycles. Ultimately, Eqn. (7) implies that Algorithm 1 can be interpreted as a noisy version of gradient descent, and the main work of the proof is to analyze the bias terms ε_m and ε'_m. To this end, notice that, while $E[\varepsilon_m | \mathcal{F}_m] = 0$ (where $\mathcal{F}_m = \{\theta_0, \tilde{\lambda}_0, \tau_0, i_0^*, i_0, \ldots, \theta_m, \tilde{\lambda}_m, \tau_m, i_m^*, i_{t_m}\}$), the new bias term ε'_m can have non-zero expectation. It is in characterizing ε'_m that this proof differs substantially from the proof of the corresponding result in [28].

As a first step, we state some preliminary properties, similar to the ones stated in Lemma 2 in [28].

1. For some $\rho, \bar{\rho} \in (0,1)$ and $C > 0$, we have

$$\bar{\rho}^{k+1} \leq P_{\theta, i^*}(T \geq k) \leq C\rho^k, \qquad \forall\, \theta \in \Theta,\ i^* \in S. \tag{8}$$

2. $E_{\theta, i^*}[T]$, and $E_{\theta, i^*}[T^2]$ are bounded functions of both $\theta \in \Theta$ and $i^* \in S$.

3. $h(r, i^*)$ and $G(\theta, i^*)$ are bounded functions of $\theta \in \Theta$, $i^* \in S$, and $\tilde{\lambda}$.

4. The sequence $\tilde{\lambda}_m$ is bounded.

Generally, these results can be proved as in [28]. The one exception is to show that $\bar{\rho}^{k+1} \leq P_{\theta, i^*}(T \geq k)$. However, this follows from Assumption 4, since the probability that a realization $i_1, \ldots, i_k$ does not return to i^* can be bounded by

$$P_{\theta, i^*}(T > k) = \prod_{n=0}^{k}(1 - P_{i_n, i^*}(\theta)) \geq \bar{\rho}^k.$$

The following proposition, established in Appendix A.1, verifies that the extra bias term ε'_m introduced by i^*-adaptation is asymptotically negligible.

Proposition 1 *Under Assumptions 1–4, if $\gamma_m = \frac{a}{b+m}$, then*

$$\sum_m \|\varepsilon'_m\| < \infty, \quad \textit{with probability 1.}$$

One consequence of Proposition 1 is that $\varepsilon'_m \to 0$ with probability 1. Following the proof in [28], we can also conclude that $\sum_m \|\varepsilon_m\|$ converges with probability 1. And hence $\varepsilon_m \to 0$ almost surely. These facts, together with the fact that $h(r_m, i^*)$ is bounded, allow us conclude that

$$\lim_{m\to\infty} \theta_{m+1} - \theta_m = 0, \qquad \lim_{m\to\infty} \lambda(\theta_{m+1}) - \lambda(\theta_m) = 0, \qquad \lim_{m\to\infty} \tilde{\lambda}_{m+1} - \tilde{\lambda}_m = 0.$$

The following proposition formalizes the notion that asymptotically the parameters θ_m are adjusted according to a noisy version of gradient ascent. The proof, given in Appendix A.2, proceeds by showing that both $\lambda(\theta_m)$ and $\tilde{\lambda}_m$ converge to the same value as $m \to \infty$.

Proposition 2 *Under the same assumptions as Proposition 1, the update equation for the parameter vector θ_m is of the form*

$$\theta_{m+1} = \theta_m + \gamma_m E_{\theta_m, i^*_m}[T](\nabla\lambda(\theta) + e_m) + \varepsilon_m + \varepsilon'_m,$$

where e_m converges to zero.

Finally, since ε_m and ε'_m are summable sequences, the remainder of the proof of Theorem 1 proceeds as in the proof of Proposition 3 in [28], choosing L to be an upper bound of $\|G(\theta, i^*)\|$ for all θ, i^*, instead of a bound on $\|G(\theta)\|$ for a single value of i^*.

If the process being optimized is a continuous-time Markov chain with maximum transition rate ν^*, it is possible to apply the i^*-adaptation mechanism to a uniformized (equivalent discrete-time) representation of the system. If the simulation of the process is only available as a continuous-time, discrete-event process, then the most natural implementation of the i^* adaptation mechanism is to (1) replace the threshold number of (discrete-time) stages τ with a threshold amount of continuous time $\bar{\tau}$ to wait before abandoning a regenerative cycle and (2) increase the value of the continuous time threshold upon processing each abandoned cycle. One issue to resolve here is that for any interval of continuous time there is a random number of state transitions in the associated uniformized representation of the system, and it is necessary to establish, as in

Appendix B, the convergence of the i^*-adaptation algorithm when the threshold number of stages τ is actually a Poisson random variable with mean $\nu^*\bar{\tau}$.

6 Discussion and Conclusions

We have analyzed a simple mechanism, i^*-adaptation, to improve the applicability of simulation-based algorithms for the optimization of average reward problems. The key idea is that, because the steady state distribution of the Markov chain changes over time (as the simulation-based algorithm evolves), a fixed rule for marking regenerative cycles can lead to a practical failure of the methodology even though eventual convergence is guaranteed with probability one. The i^*-adaptation scheme, as instantiated in Algorithm 1, addresses this issue by dynamically adjusting the value of the state used to mark regenerative cycles. We have shown that under our structural assumptions (Assumptions 1–4) Algorithm 1 retains the desirable convergence properties of [28], namely convergence with probability one to a critical point θ^* where $\nabla\lambda(\theta^*) = 0$. We point out that Algorithm 1 assumes a specific rule for increasing the cycle-breaking threshold τ, namely $\tau \leftarrow \tau + 1$. Interestingly, the proof of our main result can easily be extended to show convergence to a critical point for other rules such as the multiplicative rule $\tau \leftarrow \tau\beta$ with $\beta > 1$. Similarly, we believe that the result also holds for stepsize rules other than the specific stepsize rule $\gamma_m = a/(b+m)$, as long as $\sum_m \gamma_m = \infty$ and $\sum_m \gamma_m^2 < \infty$

Before concluding this paper, we return briefly to the numerical example of Section 3. First, we note that all Assumptions 1-4 hold in this case. Recall that this was a problem where the steady state distribution of the process depends strongly upon the single control parameter $\theta \in (.05, .95)$, as shown in Figure 1. Also, recall from Figure 2, that the original batch simulation-based algorithm of [28] performs inconsistently, depending upon the initial conditions $(i^*, \theta_0) \in \{(.9, 75), (.9, 5), (.1, 75), (.1, 5)\}$. In particular, except for the lucky choice of $i^* = 5$ and $\theta_0 = .1$, the algorithm fails to make much progress toward the unique critical point $\theta^* = .2473$, even after 10^6 simulated state transitions. In Figure 3, we show the sample evolution of Algorithm 1 for the same set of initial conditions. Notice that, regardless of where the algorithm starts, the i^* adaptation rule allows the algorithm to convergence robustly to $\theta^* = .2473$.

References

[1] A. G. Barto, S. J. Bradtke, and S. P. Singh. Learning to Act Using Real-Time Dynamic Programming. *Artificial Intelligence*, Special Volume: Computational Research on Interaction and Agency(72):81–138, 1995.

[2] D. Bertsekas. *Dynamic Programming and Optimal Control*, volume I and II. Athena Scientific, 1995.

[3] D. Bertsekas. *Dynamic Programming and Optimal Control*, volume II. Athena Scientific, 1995.

[4] D. Bertsekas and J. S. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

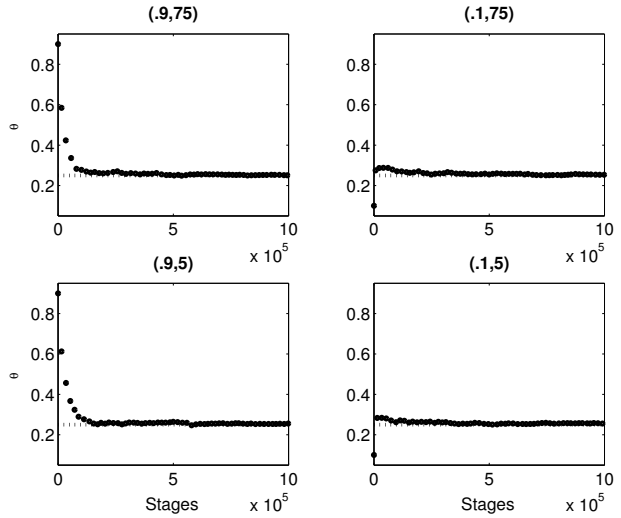

Figure 3: Sample evolution of Algorithm 1 (the batch simulation-based algorithm with i^*-adaptation) for the same initial conditions as in Figure 2.

[5] P. Billingsley. *Probability and Measure*. Wiley-Interscience, 3rd. edition, 1995.

[6] V. S. Borkar and P. Varaiya. Adaptive Control of Markov Chains, I: Finite Parameter Set. *IEEE Transactions on Automatic Control*, AC(24):953–957, 1979.

[7] V. S. Borkar and P. Varaiya. Identification and Adaptive Control of Markov Chains. *SIAM J. Control and Optimization*, 20(4):470–89, July 1982.

[8] E. Campos-Nanez and S. D. Patek. On Improving the Performance of Simulation-Based Algorithms for Average Reward Processes with Application to Network Pricing. *Proceedings of the 2001 Winter Simulation Conference*, 2001.

[9] E. K. P. Chong and P. J. Ramadge. Optimal Load Sharing in Soft Real-Time Systems using Likelihood Ratios. *Journal of Optimization Theory and Applications*, 82(1):23–48, July 1994.

[10] E. K. P. Chong and P. J. Ramadge. Stochastic Optimization of Regenerative Systems using Infinitesimal Perturbation Analysis. *IEEE Transactions on Automatic Control*, 39(7):1400–1410, July 1994.

[11] M. Fu and J. Q. Hu. *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*. Kluwer Academic Publishers, 1997.

[12] M. C. Fu. Optimization Via Simulation: A Review. *Annals of Operations Research*, 53:199–248, 1994.

[13] M. C. Fu and Y. C. Ho. Using Perturbation Analysis for Gradient Estimation, Averaging, and Updating in a Stochastic Approximation Algorithm. *Proceedings of the 1988 Winter Simulation Conference*, pages 509–517, 1988.

[14] P. Glasserman and P. W. Glynn. Gradient Estimation for Regenerative Processes. *Winter Simulation Conference*, 1992.

[15] P. W. Glynn. Stochastic Approximation for Monte Carlo Optimization. *Proceedings of the 1986 Winter Simulation Conference*, 1986.

[16] P. W. Glynn. Likelihood Ratio Gradient Estimation: An Overview. *Proceedings of the 1987 Winter Simulation Conference*, 1987.

[17] Y. C. Ho. Performance Evaluation and Perturbation Analysis of Discrete Event Systems. *IEEE Transactions on Automatic Control*, 32:563–572, 1987.

[18] Y. C. Ho and X. R. Cao. Perturbation Analysis and Optimization of Queueing Networks. *J. of Optimization Theory and Applications*, 40:559–582, 1983.

[19] Y. C. Ho and X. R. Cao. *Perturbation Analysis of Discrete Event Dynamical Systems*. Kluwer, 1991.

[20] Y. C. Ho, M. A. Eyler, and T. T. Chien. A Gradient Technique for General Buffer Storage Design in a Serial Production Line. *J. Prod. Res.*, 17:557–580, 1979.

[21] L. P. Kaelbling. *Recent Advances in Reinforcement Learning*. Kluwer Academic, 1996.

[22] L. P. Kaelbling and M. L. Littman. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[23] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23:462–466, 1952.

[24] H. J. Kushner and D. S. Clark. *Stochastic Approximation for Constrained and Unconstrained Systems*. Springer-Verlag, 1978.

[25] H. J. Kushner and J. Yang. Stochastic approximation with averaging and feedback: Rapidly convergent on line algorithms. *IEEE Transactions on Automatic Control*, AC-40:24–34, 1995.

[26] H. J. Kushner and G. G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, 1997.

[27] P. Marbach and J. N. Tsitsiklis. Gradient-Based Optimization of Markov Reward Processes: Practical Variants. Technical report, Laboratory for Information and Decision Systems / MIT, March 2000.

[28] P. Marbach and J. N. Tsitsiklis. Simulation-Based Optimization of Markov Reward Processes. *IEEE Transactions on Automatic Control*, 46(2):191–209, February 2001.

[29] M. L. Puterman. *Markov decision processes : discrete stochastic dynamic programming* . John Wiley & Sons, 1994.

[30] H. Robbins and S. Monro. A Stochastic Approxmation Method. *Ann. Math. Statist.*, 22:400–407, 1951.

[31] S. Ross. *Introduction to Stochastic Dynamic Programming*. Academic Press, 1983.

[32] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[33] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. *Advances in Neural Information Processing Systems*, 12:1057–1063, 2000.

[34] D. J. White. *Dynamic Programming*. Holden-Day, 1969.

A Proofs of the Intermediate Results

A.1 Proof of Proposition 1

First, we analyze the expectation of the update direction, conditioned on the threshold τ being exceeded.

Lemma 1 *Under Assumptions 1–3, we have for all $\theta \in \Theta$, $i^* \in S$, and $\tau > 0$*

$$E_\theta[\,\|H(r,i^*)\| \;\mid\; T > \tau] \leq \frac{2CC_2\rho^{\tau+1}}{P_{\theta,i^*}(T>\tau)}\left[\frac{(\tau+1)^2}{1-\rho} + \frac{2(\tau+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3}\right]$$

for some constant $C_2 > 0$.

Proof For a fixed value of $\tilde{\lambda}$, we have from the definition of $F(r,i^*)$ and the boundedness of $g_i(\theta)$, $\nabla g_i(\theta)$, and $L_{ij}(\theta)$ that there exists a constant C_3 such that $\|L_{ij}(\theta)(g_i(\theta)-\tilde{\lambda})\| \leq C_3$ and another constant C_4 such that $\|\nabla g_i(\theta)\| \leq C_4$. Therefore,

$$\begin{aligned}
E_\theta\,[\,\|F(r,i^*)\| \mid T>\tau\,] &= E_\theta\left[\left\|\sum_{n=0}^{T-1}\left[\sum_{k=n}^{T-1}(g_{i_k}(\theta)-\tilde{\lambda})L_{i_{n-1}i_n}(\theta)\right] + \nabla g_{i_n}(\theta)\right\| \;\middle|\; T>\tau\right] \\
&\leq E_\theta\left[\left\|\sum_{n=0}^{T-1}\left[\sum_{k=n}^{T-1} C_3\right] + C_4\right\| T>\tau\right] \\
&\leq C_2 E_\theta[T^2 \mid T>\tau].
\end{aligned}$$

A similar bound can be found for the last component of $H(r,i^*)$. To simplify, we can say that exists some constant C_2 such that for all $\theta \in \Theta$, and $i^* \in S$ and $\tau > 0$,

$$\begin{aligned}
E_{\theta,i^*}[\,\|H(r,i^*)\| \;\mid\; T>\tau] &\leq C_2 E_{\theta,i^*}[T^2 \mid T>\tau] \\
&\leq C_2 \sum_{k=\tau+1}^{\infty} k^2 P_{\theta,i^*}(T \geq k \mid T>\tau) \\
&= \frac{C_2}{P_{\theta,i^*}(T>\tau)}\sum_{k=0}^{\infty}(k+\tau+1)^2 P_{\theta,i^*}(T \geq k+\tau+1) \\
&\leq \frac{CC_2\rho^{\tau+1}}{P_{\theta,i^*}(T>\tau)}\sum_{k=0}^{\infty}(\tau+1+k)^2\rho^k \\
&= \frac{CC_2\rho^{\tau+1}}{P_{\theta,i^*}(T>\tau)}\left[\frac{(\tau+1)^2}{1-\rho} + \frac{2(\tau+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3}\right]
\end{aligned}$$

where the last equality arises from the fact that it is possible to rewrite the sum $\sum_{k=0}^{\infty} k^2\rho^k = \frac{\rho(\rho+1)}{(1-\rho)^3}$. The constant C is defined in Eqn. (8). ■

Now, we obtain a bound on the expected bias that can be observed after n cycles have been broken (i.e. "broken" as in Step 2.(a) of Algorithm 1). Let $\{m_n\}$ be the subsequence of cycles that are broken, so that $\sum_n \varepsilon'_{m_n} = \sum_m \varepsilon'_m$.

Lemma 2 *Under Assumptions 1–4, if* $\gamma_m = \frac{a}{b+m}$*, we have*

$$E[\|\varepsilon'_{m_{n+1}}\| \mid \mathcal{F}_{m_n}] \leq -\frac{2aCC_2\rho^{\tau_{m_n}}}{(1-\bar{\rho}^{\tau_{m_n}})^{b+n}}\tau_{m_n} log(\bar{\rho}) \left[\frac{(\tau_{m_n}+1)^2}{1-\rho} + \frac{2(\tau_{m_n}+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3}\right],$$

for all $a, b > 0$.

Proof: To obtain a bound on $E[\|\varepsilon'_{m_{n+1}}\| \mid \mathcal{F}_{m_n}]$, we can condition on the number of normal (complete) cycles observed before the next cycle is dropped. Then, we obtain by using the bound derived in Lemma 1.

$$\begin{aligned}
E[\|\varepsilon'_{m_{n+1}}\| \mid \mathcal{F}_{m_n}] &= E[\gamma_{m_{n+1}}\|H(r_{m_{n+1}}, i^*_{m_{n+1}})\| \mid \mathcal{F}_{m_n}] \\
&= \sum_{k=1}^{\infty} \gamma_{m_n+k}\Bigg[\Big(\prod_{l=1}^{k-1} P(T_{m_n+l} \leq \tau_{m_n} \mid \mathcal{F}_{m_n+l-1})\Big) \\
&\qquad \times P(T_{m_n+k} > \tau_{m_n} \mid \mathcal{F}_{m_n+k-1}) \\
&\qquad\qquad \times E[\|H(r_{m_n+k}, i^*_{m_n})\| \mid \mathcal{F}_{m_n+k-1}, T_{m_n+k} > \tau_{m_n}]\Bigg] \\
&\leq \sum_{k=1}^{\infty} \gamma_{m_n+k}(1-\bar{\rho}^{\tau_{m_n}+1})^{k-1} \frac{P(T_{m_n+k} > \tau_{m_n} \mid \mathcal{F}_{m_n+k-1})}{P(T_{m_n+k} > \tau_{m_n} \mid \mathcal{F}_{m_n+k-1})} 2CC_2\rho^{\tau_{m_n}+1} \\
&\qquad \times \left[\frac{(\tau_{m_n}+1)^2}{1-\rho} + \frac{2(\tau_{m_n}+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3}\right],
\end{aligned}$$

where the value of C_2 reflects the result of Lemma 1 applied to the bounded sequence of average reward estimates $\{\tilde{\lambda}_m\}_{m=0}^{\infty}$. Then, using the stepsize $\gamma_m = \frac{a}{b+m}$, we obtain

$$\begin{aligned}
E[\|\varepsilon'_{m_{n+1}}\| \mid \mathcal{F}_{m_n}] &\leq 2CC_2\rho^{\tau_{m_n}+1}\left[\frac{(\tau_{m_n}+1)^2}{1-\rho} + \frac{2(\tau_{m_n}+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3}\right] \sum_{k=1}^{\infty} \frac{a(1-\bar{\rho}^{\tau_{m_n+1}})^{k-1}}{m_n+b+k} \\
&\leq 2CC_2\rho^{\tau_{m_n}+1}\left[\frac{(\tau_{m_n}+1)^2}{1-\rho} + \frac{2(\tau_{m_n}+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3}\right] \sum_{k=1}^{\infty} \frac{a(1-\bar{\rho}^{\tau_{m_n+1}})^{k-1}}{n+b+k} \\
&\leq -\frac{2aCC_2\rho^{\tau_{m_n}+1}}{(1-\bar{\rho}^{\tau_{m_n+1}})^{n+b}}\left[\frac{(\tau_{m_n}+1)^2}{1-\rho} + \frac{2(\tau_{m_n}+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3}\right] \tau_{m_n} log(\bar{\rho}).
\end{aligned}$$

The last relation is based on the fact that $\sum_{k=0}^{\infty} \frac{p^k}{k} = -log(1-p)$, for all $0 < p < 1$, and that

$$\begin{aligned}\sum_{k=1}^{\infty} \frac{p^k}{M+k} &= \frac{1}{p^M} \sum_{k=M}^{\infty} \frac{p^k}{k} \\ &= \frac{1}{p^M} \left[-log(1-p) - \sum_{k=1}^{M} \frac{p^k}{k} \right] \\ &\leq -\frac{1}{p^M} log(1-p),\end{aligned}$$

for all $0 < p < 1$, and all positive integers M. ■

From the rule for increasing the threshold τ (in Step 2.(a) of Algorithm 1), we observe that τ_{m_n} depends explicitly on the number of cycles n that have been broken and not on the cycle number m_n which was the n-th to be broken. Specifically, we have that $\tau_{m_n} = \tau_0 + n$. We will use this fact shortly to obtain a bound on the unconditional expected value of the bias that has accumulated up to the n-th broken cycle.

Defining $S_n = \sum_{k=0}^{n} \|\varepsilon'_{m_k}\|$, we see that the sequence $\{S_n\}_{n=1}^{\infty}$ is a submartingale. Moreover, from Lemma 2, we have that

$$S_n \leq E[S_{n+1} \mid \mathcal{F}_{m_n}] \leq S_n + \frac{2aCC_2\rho^{\tau_{m_n+1}}}{(1-\bar{\rho}^{\tau_{m_n}+1})^{n+b}} \tau_{m_n} log(\bar{\rho}) \left[\frac{(\tau_{m_n}+1)^2}{1-\rho} + \frac{2(\tau_{m_n}+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3} \right].$$

Plugging in $\tau_{m_n} = \tau_0 + n$ and summing the bound over $k = 1, \ldots, n$, we conclude that

$$E[S_{n+1}] \leq \sum_{k=1}^{n} K_1 \rho^{\tau_0+k} \left[K_2(\tau_0+k)^3 \right],$$

for some constants K_1, and K_2. Since the series in this bound is summable, we have that $\sup_n E[|S_n|] = K < \infty$, for some constant K. By Doob's Theorem (see [5]), the sequence $\lim_{n\to\infty} S_n$ converges almost surely to some random variable S with $E[S] \leq K < \infty$. ■

A.2 Proof of Proposition 2

As in [28], we proceed by constructing a family of Lyapunov functions that will be used to analyze the algorithm

$$r_{m+1} = r_m + \gamma_m h(r_m, i_m^*) + \varepsilon_m + \varepsilon'_m.$$

Let $\mathcal{D}_c = \{(\theta, \tilde{\lambda}) \in \Re^{K+1} || \tilde{\lambda} \leq c\}$. As in [28], we consider only functions ϕ that are twice differentiable and for which ϕ, $\nabla\phi$, and $\nabla^2\phi$ are bounded on $\mathcal{D}_c$ for every c. Let Φ be the set of all such Lyapunov functions. For any $\phi \in \Phi$, we define

$$\tilde{\varepsilon}_m(\phi) = \phi(r_{m+1}) - \phi(r_m) - \gamma_m \nabla\phi(r_m) \cdot h(r_m, i_m^*),$$

where $a \cdot b$ denotes the inner product of vectors a, b.

Lemma 3 *If $\phi \in \Phi$, then the series $\sum_m \tilde{\varepsilon}_m(\phi)$ converges with probability 1.*

Proof: We proceed as in [28] noting only that

$$\nabla\phi(r_m) \cdot (\varepsilon_m + \varepsilon'_m) - M\|r_{m+1} - r_m\|^2 \leq \tilde{\varepsilon}_m(\phi) \leq \nabla\phi(r_m) \cdot (\varepsilon_m + \varepsilon'_m) + M\|r_{m+1} - r_m\|^2.$$

From here it is enough to show that all of the terms of the upper and lower bounds are summable. Summability of $\nabla\phi(r_m) \cdot (\varepsilon_m + \varepsilon'_m)$ follows from the boundedness of $\nabla\phi(r_m)$ and the fact that both ε_m and ε'_m are summable. (The summability of $\|\varepsilon_m\|$, and therefore the summability of ε_m, follows as in [28]. Summability of ε'_m follows from Proposition 1.) It remains to prove the square summability of $\|r_{m+1} - r_m\|$. Noting that $\|r_{m+1} - r_m\| = \|\gamma_m h(r_m, i_m^*) + \varepsilon_m + \varepsilon'_m\|$, we have

$$\|r_{m+1} - r_m\|^2 \leq 2\gamma_m^2 \|h(r_m, i_m^*)\|^2 + 2\|\varepsilon_m\|^2 + 2\|\varepsilon'_m\|^2.$$

Again, it suffices to show that all of the terms of the bound are summable. To dispose of the first term, recall that the sequence $h(r_m)$ is bounded and γ_m^2 is summable. Moreover, from the fact that $\|\varepsilon_m\|$ is summable, we infer that $\|\varepsilon_m\|^2$ is summable. Similarly, from Proposition 1, we infer that $\|\varepsilon'_m\|^2$ is summable. Thus, we conclude that $\|r_{m+1} - r_m\|$ is square summable, and the result follows. ■

We now proceed by analyzing the stability of the algorithm in different regions, as in [28]. Slight modifications of the argument are required to account for the fact that the state i_m^* changes over time. The stability of the algorithm will follow from the intuitive fact that, even though i_m^* may vary in time, the expected direction of adjustments to θ_m is still consistent with the direction of the gradient. In other words, only the constants $E_{\theta_m, i_m^*}[T_m]$, and $G(\theta_m, i_m^*)$ change in the expression

$$E[H(r_m)] = E_{\theta_m, i_m^*}[T_m] \nabla\lambda(\theta_m) + G_{\theta, i_m^*}(\theta_m)(\lambda(\theta_m) - \tilde{\lambda}_m).$$

To formalize this, we restate and argue Lemmas 5-11 from [28].

Lemma 4 *Let L be such that $\|G(\theta, i^*)\| \leq L$ for all θ, i^*, and let*

$$\phi(r) = \tilde{\lambda} - \lambda(\theta).$$

We have $\phi \in \Phi$. Furthermore if $0 \leq \tilde{\lambda} - \lambda(\theta) \leq \eta/L^2$ then

$$\nabla\phi(r) \cdot h(r, i^*) \leq 0, \quad \forall r, a.$$

Proof: By definition of $\phi(r)$, and the expression for $h(r, a)$, we have

$$\nabla\phi(r) \cdot h(r, a) = -\eta(\tilde{\lambda} - \lambda(\theta))E_{\theta,i^*}[T] - \|\nabla\lambda(\theta)\|^2 E_{\theta,i^*}[T] + (\tilde{\lambda} - \lambda(\theta))\nabla\lambda(\theta) \cdot G(\theta, i^*).$$

Since $E_{\theta,i^*}[T] \geq 1$, and by the Cauchy-Schwartz inequality we have that

$$(\tilde{\lambda} - \lambda(\theta))\nabla\lambda(\theta) \cdot G(\theta, i^*) \leq \|\nabla\lambda(\theta)\|^2 + (\tilde{\lambda} - \lambda(\theta))^2\|G(\theta, i^*)\|^2,$$

which lets us conclude that

$$\nabla\phi(r) \cdot h(r, i^*) \leq -\eta(\tilde{\lambda} - \lambda(\theta)) + L^2(\tilde{\lambda} - \lambda(\theta))^2.$$

Finally, if the condition $0 \leq \tilde{\lambda} - \lambda(\theta) \leq \eta/L^2$ holds then the result holds. ■

The following lemma follows directly from the corresponding result in [28], so we omit the proof.

Lemma 5 *As in Lemma 4, let L be such that $\|G(\theta, i^*)\| \leq L$. Let also*

$$\phi(r) = \phi(\theta, \tilde{\lambda}) = \lambda(\theta) - (L^2/\eta)(\lambda(\theta) - \tilde{\lambda})^2.$$

We have $\phi \in \Phi$. Furthermore, if $|\lambda(\theta) - \tilde{\lambda}| \leq \eta/4L^2$, then

$$\nabla\phi(r) \cdot h(r, i^*) \geq 0.$$

Lemma 6 *Consider the same function ϕ as in Lemma 5, and the same constant L. Let α be some positive scalar smaller*

than $\eta/4L^2$. Suppose that for some integers n and n', with $n' > n$, we have

$$|\lambda(\theta_n) - \tilde{\lambda}_n| \leq \alpha, \qquad |\lambda(\theta_{n'} - \tilde{\lambda}_{n'}| \leq \alpha,$$

and

$$|\lambda(\theta_m) - \tilde{\lambda}_m| \leq \frac{\eta}{4L^2}, \quad m = n+1, \ldots, n'-1.$$

Then,

$$\tilde{\lambda}_{n'} \geq \tilde{\lambda}_n - 2\alpha\left((L^2\alpha/\eta) + 1\right) + \sum_{m=n}^{n'-1} \varepsilon_m(\phi).$$

Again, the proof is omitted since it rests in the arguments presented in [28].

Lemma 7 *We have* $\liminf_{m\to\infty} |\lambda(\theta_m) - \tilde{\lambda}_m| = 0$.

The proof is again omitted since the arguments in [28] hold by considering δ_m to be the last component of $\varepsilon_m + \varepsilon'_m$, which we have shown is summable. We also consider L to be an upper bound on all $\|G(\theta, i^*)\|$.

Similarly, proofs of the following lemmas will also be omitted.

Lemma 8 *We have* $\liminf_{m\to\infty} \left(\lambda(\theta_m) - \tilde{\lambda}_m\right) \geq 0$.

Lemma 9 *We have* $\lim_{m\to\infty} \left(\lambda(\theta_m) - \tilde{\lambda}_m\right) = 0$.

Lemma 10 *The sequences $\tilde{\lambda}_m$ and $\lambda(\theta_m)$ converge.*

B Continuous Time Implementation of the i^*-adaptation Method

In this appendix we consider the application of Algorithm 1 to continuous-time Markov chains. We restrict attention to controlled Markov chains that have a maximal transition rate ν^*. In principle, it is possible to apply Algorithm 1 to an equivalent representation of the process through uniformization (cf. [3], Chapter 7). Uniformization essentially samples the state of the continuous-time Markov chain according to a Poisson process with rate ν^*, so the discrete-time representation of the process exhibits a random (Poisson-distributed) number of self-transitions between each "real" state transition in the continuous-time process. A potential problem in applying Algorithm 1 is that, depending on the context of the application, it may not be possible to observe these self-transitions directly. Since the distribution of the number of self-transitions is

known, it is possible account for the effect of self-transitions by generating a Poisson random variable (with the appropriate mean) for each state transition in the continuous time process, and thus the update rule of Eqns. (5) and (6) can be faithfully reproduced in continuous time.

Unfortunately, if the self-transitions cannot be observed directly, the appropriate instant in (continuous) time to break a cycle for i^*-adaptation can be difficult to judge. The natural thing to do in this case is to set a deterministic window of continuous time for deciding to break the process. In the context of the uniformized chain, this corresponds to a Poisson-distributed random number of stages (a random threshold) for determining when to stop the process, reset i^* and continue. In the following we show that Algorithm 1 retains its convergence properties if we allow the threshold value τ_m to be random in this way.

Formally, let $\bar{\tau}_m$ be a time threshold for the duration of the cycle (in continuous time), and let τ_m be the random number of steps that can be observed by the uniformized chain during that same time interval. In other words, $\tau_m \sim Poisson(\bar{\tau}_m \nu^*)$. The continuous-time threshold will be increased by the average interarrival time of the uniformized chain each time a cycle is abandoned, i.e. $\bar{\tau}_{m+1} = \bar{\tau}_m + \mathbf{1}_{\{i_{t_{m+1}} \neq i_m^*\}}/\nu^*$. We now redefine

$$t_{m+1} = \min\{t_m + \tau_m, \min\{n | n > t_m, i_n = i_m^*\}\}$$

and $\varepsilon'_m = \mathbf{1}_{\{i_{t_{m+1}} \neq i_m^*\}} \gamma_m H(r_m)$.

Proposition 3 *If* $\gamma_m = \frac{a}{b+m}$ *for* $a, b > 0$ *then*

$$\sum_m \|\varepsilon'_m\| \leq \infty.$$

To show this, we recall that that if $\gamma_m = \frac{a}{b+m}$ for $a, b > 0$, and by defining $\{m_n\}_{n=0}^{\infty}$ to be the subsequence of abandoned cycles, then

$$E[\|\gamma_{m_n} H(r_{m_n})\| \mid t_{m_{n+1}} - t_{m_{n+1}-1} > \tau_{m_n}] \leq K \tau_{m_n} \frac{\rho^{\tau_{m_n}+1}}{(1-\bar{\rho}^{\tau_{m_n}+1})^{n+b}} \left[\frac{(\tau_{m_n}+1)^2}{1-\rho} + \frac{2(\tau_{m_n}+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3} \right],$$

where $K = -log(1-\bar{\rho}) 2aCC_2$, with C and C_2 constants from earlier arguments. Using this result, we can show the following lemma.

Lemma 11

$$E[\|\varepsilon'_{m_n}\|] \leq \frac{K\rho}{(1-\bar{\rho})^b} \, e^{-\bar{\tau}_{m_n} \nu^* (1-\psi)} K_2 (\psi \nu^* \bar{\tau}_{m_n})^3,$$

where $\psi = \frac{\rho}{1-\bar{\rho}}$, *and for some* $K_2 > 0$.

Proof: By conditioning on the number of events τ_n that can be observed during an interval of length $\bar{\tau}_n$, we can obtain

$$\begin{aligned} E[\|\varepsilon'_{m_n}\|] &= E[E[\|\gamma_{m_n} H(r_{m_n})\| | t_{m_n} - t_{m_n-1} > \tau_{m_n}]] \\ &= \sum_{t=0}^{\infty} P(\tau_{m_n} = t) E[\|\gamma_{m_n} H(r_{m_n})\| | t_{m_n} - t_m > t]. \end{aligned}$$

By the result in Lemma 2, we obtain

$$\begin{aligned} E[\|\varepsilon'_{m_n}\|] &\leq \sum_t e^{-\nu^* \bar{\tau}_{m_n}} \frac{(\nu^* \bar{\tau}_{m_n})^t}{t!} K \frac{\rho^{t+1}}{(1-\bar{\rho}^{t+1})^{t+b}} t \left[\frac{(t+1)^2}{1-\rho} + \frac{2(t+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3} \right] \\ &\leq \frac{K\rho e^{-\nu^* \bar{\tau}_{m_n}}}{(1-\bar{\rho})^b} \sum_t \left(\frac{\nu^* \bar{\tau}_{m_n} \rho}{1-\bar{\rho}} \right)^t \frac{1}{t!} t \left[\frac{(t+1)^2}{1-\rho} + \frac{2(t+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3} \right]. \end{aligned}$$

In the last inequality, the fact that $1 - \bar{\rho}^b > 1 - \bar{\rho}$ for all $b > 1$ was used. Finally, multiplying and dividing by $e^{-\psi \nu^* \bar{\tau}_{m_n}}$ we obtain

$$E[\|\varepsilon'_{m_n}\|] \leq \frac{K\rho e^{-\bar{\tau}_{m_n} \nu^* (1-\psi)}}{(1-\bar{\rho})^b} \sum_t e^{-\psi \nu^* \bar{\tau}_{m_n}} (\psi \nu^* \bar{\tau}_{m_n})^t \frac{1}{t!} t \left[\frac{(t+1)^2}{1-\rho} + \frac{2(t+1)\rho}{(1-\rho)^2} + \frac{\rho(\rho+1)}{(1-\rho)^3} \right].$$

By using the first three moments of a Poisson distribution with parameter $\psi \nu^* \bar{\tau}_{m_n}$, we obtain a polynomial of degree three, for which we can find a constant K_2 that satisfies the desired result. ■

Proof of Proposition 3

Note that $S_n = \sum_{k=0}^{n} \|\varepsilon'_{m_n}\|$ is a supermartingale. Moreover,

$$E[S_n] \leq \frac{K\rho}{(1-\bar{\rho})^b} \sum_{k=1}^{\infty} e^{-\bar{\tau}_{m_k} \nu^* (1-\psi)} K_2 (\psi \nu^* \bar{\tau}_{m_k})^3$$

and $\bar{\tau}_{m_n} = \bar{\tau}_0 + n/\nu^*$. Thus, by choosing $\bar{\rho}$ such that $\rho + \bar{\rho} < 1$, the right hand side of the expression above is convergent almost surely. (It is always possible to choose $\bar{\rho}$ so that $\rho + \bar{\rho} < 1$ since $\rho < 1$ by definition and $\bar{\rho}$ is a lower bound for $1 - P_{ii^*}(\theta)$ and can be chosen arbitrarily small, say $\bar{\rho}' < \min\{\frac{1-\rho}{2}, \bar{\rho}\}$.) Therefore, $\sum_n \|\varepsilon'_{m_n}\| < \infty$. ■