It is a mistake to try to look too far ahead. The chain of destiny can only be grasped one link at a time.

Sir Winston Churchill
British politician
(1874 - 1965)

5 Sample-Path-Based Policy Iteration

In Chapter 3, we showed that potentials and performance gradients can be estimated with a sample path of a Markov chain, and the estimated potentials and gradients can be used in gradient-based performance optimization of Markov systems. In this chapter, we show that we can use sample-path-based potential estimates in policy iteration to find optimal policies. We focus on the average-reward optimality criterion and ergodic Markov chains. The main idea is as follows. At each iteration k with policy d_k, instead of solving the Poisson equation for potential g^{d_k}, we use its sample-path-based estimate $\bar{g}^{d_k}$ as an approximation in the policy improvement step to determine an improved policy. This leads to sample-path-based policy iteration algorithms.

This approach has several advantages. For example, it does not require solving a large number of linear equations and/or knowing the exact form/value of the transition probability matrix (see Section 5.1). These advantages make the approach practically useful, because for many real engineering systems such as communication networks or manufacturing systems the state spaces are too large and the transition probability matrices may not be entirely known due to unknown parameters and/or to the complexity of the system's structure. However, because the estimates may contain errors, a sample-path-based policy iteration algorithm may not converge, or if it does, it may not converge to an optimal policy. In this chapter, we propose some sample-path-based policy iteration algorithms and provide some conditions that ensure the convergence (either in probability, or with probability 1) of these algorithms to optimal policies.

Similar to the PA-based optimization in Section 6.3.1, there are two ways to implement sample-path-based policy iteration. We may first run the system long enough under one policy at every iteration to get accurate estimates of the potentials and then use them to update the policy, or we may run the system for a short period to get noisy estimates of the potentials, especially at the beginning of the policy iteration, and then gradually improve the estimates as we approach an optimal policy. These topics are discussed in Sections 5.2 and 5.3, respectively.

This chapter complements Chapter 3. Sample-path-based perturbation analysis applies to optimization problems with continuous parameters, while sample-path-based policy iteration applies to optimization problems in discrete policy spaces. This chapter is mainly based on [54], [88], and [97].

5.1 Motivation

We first use a well-designed example to show the advantages of the sample-path-based policy iteration approach.

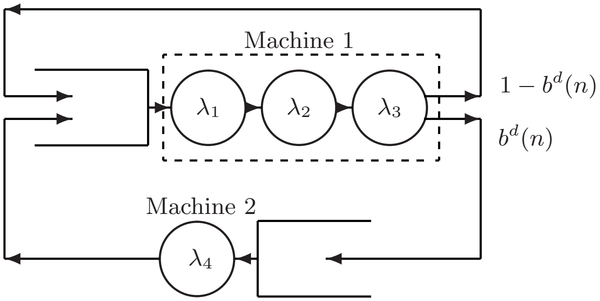

Fig. 5.1. A Two-Machine Manufacturing System

An Illustrative Example

Example 5.1. A manufacturing system consists of two machines and N pieces of works, which are circulating between the two machines, as shown in Figure 5.1. Each work piece has to undertake three consecutive operations at machine 1; thus, machine 1 is illustrated by three circles in the figure, each for one operation. The service times at these three operations are exponentially distributed with rates λ_1, λ_2, and λ_3, respectively. Machine 2 has only one operation with an exponentially distributed service time with rate λ_4. A work piece, after the completion of its service at machine 1, goes to machine 2 with probability $b^d(n)$ and feeds back to machine 1 with probability $1 - b^d(n)$. The superscript "d" represents a policy with $d \in \mathcal{D}$. For any $d \in \mathcal{D}$

and $n = 1, 2, \ldots, N$, $b^d(n) \in [0, 1]$. The system can be modelled as a Markov process with its state denoted as (n, i), $0 \leq n \leq N$, where n is the number of pieces at machine 1 and $i = 1, 2, 3$ denotes the operation that the piece at machine 1 is undertaking. When $n = 0$, we simply denote the state as 0. To apply the results for discrete-time Markov chains, we study the Markov chain embedded at the transition epochs. We assume that the cost function f does not depend on the actions.

The transition probabilities of the embedded Markov chain are

$$\begin{aligned}
p\left[(n,1),(n+1,1)\right] &= \frac{\lambda_4}{\lambda_1+\lambda_4}, \\
p\left[(n,1),(n,2)\right] &= \frac{\lambda_1}{\lambda_1+\lambda_4}, \\
p\left[(n,2),(n+1,2)\right] &= \frac{\lambda_4}{\lambda_2+\lambda_4}, \\
p\left[(n,2),(n,3)\right] &= \frac{\lambda_2}{\lambda_2+\lambda_4}, \\
p\left[(n,3),(n+1,3)\right] &= \frac{\lambda_4}{\lambda_3+\lambda_4}, \\
p^d\left[(n,3),(n-1,1)\right] &= \frac{\lambda_3}{\lambda_3+\lambda_4} b^d(n), \\
p^d\left[(n,3),(n,1)\right] &= \frac{\lambda_3}{\lambda_3+\lambda_4}\left[1-b^d(n)\right],
\end{aligned}$$

for $0 < n < N$; and

$$\begin{aligned}
p\left[0,(1,1)\right] &= 1, \\
p\left[(N,1),(N,2)\right] &= p\left[(N,2),(N,3)\right] = 1, \\
p^d\left[(N,3),(N,1)\right] &= 1-b^d(N), \\
p^d\left[(N,3),(N-1,1)\right] &= b^d(N).
\end{aligned}$$

The other transition probabilities are zeros.

We can see that (4.5) and (4.6) in step 3 of the policy iteration algorithm in Chapter 4 can be simplified. The transitions from states $(n, 1)$ and $(n, 2)$ do not depend on actions. The comparison of actions in the policy improvement step for state $(n, 3)$, $0 < n < N$, becomes (recall that the cost function does not depend on actions):

$$\begin{aligned}
&\frac{1}{\lambda_3+\lambda_4}\left\{\lambda_4 g^d(n+1,3) + \lambda_3 b^d(n) g^d(n-1,1) + \lambda_3\left[1-b^d(n)\right] g^d(n,1)\right\} \\
\geq &\frac{1}{\lambda_3+\lambda_4}\left\{\lambda_4 g^d(n+1,3) + \lambda_3 b^{d'}(n) g^d(n-1,1) + \lambda_3\left[1-b^{d'}(n)\right] g^d(n,1)\right\},
\end{aligned}$$

for all $d' \in \mathcal{D}$. This is equivalent to

$$\left[b^d(n) - b^{d'}(n)\right] g^d(n-1,1) - \left[b^d(n) - b^{d'}(n)\right] g^d(n,1) \geq 0. \qquad (5.1)$$

The system parameters, λ_1, λ_2, λ_3, and λ_4, do not appear in (5.1). □

In the above example, the service rates govern the evolution of the system, which runs automatically. The control action can affect only some of the system transitions. The transition probabilities corresponding to the uncontrolled transitions (e.g., the transition from $(n, 3)$ to $(n + 1, 3)$) are the same under all policies; they cancel each other in the comparison equation and hence do not appear in the final form. If we can estimate the (g^d)'s on a sample path, then we can implement policy iteration without knowing these transition probabilities, or the corresponding service rates.

Many practical systems have the same features as the above example. Indeed, in many systems, control can be exercised only in a very limited region (e.g., admission control can be applied only at the access points of a high-speed communications network); the remaining parts of such systems simply evolve through their own natures. In Example 5.1, the dashed box in Figure 5.1 can also be viewed as a machine whose service time has an Erlangian distribution. In such cases, the transitions between the different stages are not controllable. This type of service distribution and the more general forms, such as Coxian distributions and phase-type distributions, are very common in practical systems.

The Advantages of the Sample-Path-Based Approach

In summary, Example 5.1 and the above discussion illustrate that the sample-path-based approach has the following advantages.

1. Given a sample path, policy iteration can be implemented *without knowing the whole transition matrix*; only those items related to control actions, $b^d(n)$, have to be known (we do not even need to know the related transition probabilities, e.g., we only need to know $b^d(n)$, not $\frac{\lambda_3}{\lambda_3+\lambda_4}b^d(n)$). In particular, we do not need to estimate all the system parameters λ_i, $i = 1, 2, 3, 4$. Matrix inversion is not required.
2. The approach *saves memory space* required for implementing MDP. In general, only the S potentials, not the $S \times S$ transition matrix, have to be stored. This can be further reduced when there are some states that cannot be reached by "controllable" states, in which actions can be applied. As shown in (5.1), in Example 5.1 only $g^d(n, 1)$, $n = 0, 1, \ldots, N$, have to be estimated and stored; $g^d(n, 2)$ and $g^d(n, 3)$, $n = 0, 1, \ldots, N$, do not even need to be estimated.
3. In the standard computational approach, all the potentials are obtained together through a matrix inversion; thus, obtaining the potential of one state involves the same effort as obtaining the potentials of all the states. In the sample-path-based approach *potentials can be estimated one by one.* This feature makes the policy iteration procedure much more flexible.

a) The computational efforts and memory space of each iteration may be further reduced at the cost of the convergence rate. The idea is, if a state space is too large, at each iteration we may estimate the potentials of only a subset of the state space and update the actions that control the system moving to the states in this subset. For instance, in Example 5.1, we may set $0 = n_0 < n_1 < n_2 < \cdots < n_{k-1} < n_k = N$. Then, in the ith iteration, $i = 1, 2, \ldots, k$, we may estimate $g^d(n, 1)$ only for $n_{i-1} \leq n \leq n_i$, $i = 1, 2, \ldots, k$. Then, by (5.1), we may update the actions in $b^d(n)$ for $n = n_{i-1} + 1, \ldots, n_i$. Of course, it may need more iterations to reach the optimal policy; however, at each iteration, the computation and the memory requirement may be reduced to fit the capacity of the available computing equipment. This feature may be important for on-line optimization using specially designed hardware which may have limited capacity (e.g., in sensor networks). In high speed communications networks, the effect of a slow convergence rate in terms of iterations may be compensated by the fast speed in the system evolution.

b) For many practical systems, we may have some *a priori* knowledge about which states are more important than others. Then, we can estimate only the potentials of the states that are needed for updating the actions on these important states. This may reduce the computation and memory at the cost of the best performance achieved.

c) For large systems for which matrix inversion is not feasible (even if the matrix is completely known), we may simulate the system by using its particular structure (e.g., the queueing structure) and apply the above two methods to reach the optimal solution with more iterations or to obtain a near optimal solution.

d) Distributed optimization may be possible. For example, suppose that we have a communications network consisting of K nodes, which can be modelled as a closed queueing network of K single-server stations, with each server representing one node. Then, the routing decision can be made at each individual node with only the relevant potentials being estimated. This approach depends on state aggregation to further reduce the number of potentials to be estimated (see Chapters 8 and 9 for more discussion). This is an important research direction and more work needs to be done.

The convergence property of sample-path-based policy iteration depends on the errors of the potential estimates, which depend on the length of the sample paths used in the estimation. The remaining sections in this chapter are devoted to the study of the convergence issue.

5.2 Convergence Properties

We can use any algorithm in Section 3.1.2 to estimate potentials. The study in this section is based on (3.19), which expresses the potentials as the average of independent samples, each observed in one regenerative period defined in (3.18). Now let us write it in a slightly different form. First, we choose a reference state $i^* \in \mathcal{S}$. For convenience, we assume that $X_0 = i^*$. Define

$$\begin{aligned} &l_0(i^*) = 0 \\ &l_k(i^*) = \min\{l : \ X_l = i^*, \ l > l_{k-1}(i^*)\}, \qquad k \geq 1. \end{aligned}$$

The instants $l_0(i^*), l_1(i^*), \ldots, l_k(i^*), \ldots$, are regenerative points of the Markov chain $\boldsymbol{X} = \{X_l, l = 0, 1, \ldots\}$, and the sample path between $l_k(i^*)$ and $l_{k+1}(i^*)$ is the kth regenerative period. Next, we define $l_k(j) = \min\{l : l > l_k(i^*), X_l = j\}$, $k = 0, 1, \ldots$, and $\chi_k(j) = 1$ if $l_k(j) < l_{k+1}(i^*)$, and $\chi_k(j) = 0$ otherwise. $\chi_k(j)$ indicates whether the system visits state j in the kth regenerative period. The definition is notationally different from but essentially the same as (3.18): The Markov chain may not visit a given state j in a regenerative period.

Consider N regenerative periods. If $\chi_k(j) = 1$, we define

$$V_k(i^*, j) := \sum_{l=l_k(j)}^{l_{k+1}(i^*)-1} [f(X_l) - \bar{\eta}_N],$$

where $\bar{\eta}_N$ is the estimated performance based on N regenerative periods:

$$\bar{\eta}_N := \frac{\sum_{k=0}^{N-1} \left[\sum_{l=l_k(i^*)}^{l_{k+1}(i^*)-1} f(X_l)\right]}{\sum_{k=0}^{N-1} [l_{k+1}(i^*) - l_k(i^*)]} = \frac{1}{l_N(i^*)} \sum_{l=0}^{l_N(i^*)-1} f(X_l). \qquad (5.2)$$

$V_k(i^*, j)$ is undefined if $\chi_k(j) = 0$. Let

$$N(j) := \sum_{k=0}^{N-1} \chi_k(j). \qquad (5.3)$$

Because of the ergodicity, we have $\lim_{N\to\infty} N(j) = \infty$.

Now, we set $g(i^*) = 0$. Then, the estimated potential of state j, $j \neq i^*$, using N regenerative periods, is

$$\bar{g}_N(j) = \frac{1}{N(j)} \left\{ \sum_{k=0}^{N-1} \chi_k(j) V_k(i^*, j) \right\} \approx \gamma(i^*, j) = g(j), \qquad (5.4)$$

if $N(j) > 0$. $\bar{g}_N(j)$ is undefined if $N(j) = 0$.

5.2.1 Convergence of Potential Estimates

By the law of large numbers [26, 28], we have

$$\lim_{N\to\infty} \bar{\eta}_N = \eta, \qquad \text{w.p.1.} \tag{5.5}$$

Lemma 5.1. *As the number of regenerative periods $N \to \infty$, the sample-path-based potential estimate $\bar{g}_N(j)$ in (5.4) converges to its true value $g(j)$ with probability 1.*

Proof. First, let

$$\begin{aligned}\widetilde{V}_k(i^*, j) :=& \sum_{l=l_k(j)}^{l_{k+1}(i^*)-1} [f(X_l) - \eta] \\ =& V_k(i^*, j) - \sum_{l=l_k(j)}^{l_{k+1}(i^*)-1} (\eta - \bar{\eta}_N) \\ =& V_k(i^*, j) - \{l_{k+1}(i^*) - l_k(j)\}(\eta - \bar{\eta}_N).\end{aligned} \tag{5.6}$$

Then, we have

$$\begin{aligned}\bar{g}_N(j) = & \frac{1}{N(j)}\left\{\sum_{k=0}^{N-1} \chi_k(j)\widetilde{V}_k(i^*, j)\right\} \\ & + (\eta - \bar{\eta}_N)\left\{\frac{1}{N(j)}\left\{\sum_{k=0}^{N-1} \chi_k(j)\left[l_{k+1}(i^*) - l_k(j)\right]\right\}\right\}.\end{aligned}$$

By the law of large numbers [26, 28], we have

$$\lim_{N\to\infty} \frac{1}{N(j)}\left\{\sum_{k=0}^{N-1} \chi_k(j)\widetilde{V}_k(i^*, j)\right\} = g(j), \qquad \text{w.p.1,} \tag{5.7}$$

and

$$\lim_{N\to\infty} \frac{1}{N(j)}\left\{\sum_{k=0}^{N-1} \chi_k(j)\left[l_{k+1}(i^*) - l_k(j)\right]\right\} = E\left[l_{k+1}(i^*) - l_k(j)\right], \qquad \text{w.p.1,} \tag{5.8}$$

which is the average first-passage time from state j to state i^*. Because $E\left[l_{k+1}(i^*) - l_k(j)\right] < \infty$ and from (5.5), (5.7), and (5.8), we have

$$\lim_{N\to\infty} \bar{g}_N(j) = g(j), \qquad j \in \mathcal{S}, \quad \text{w.p.1.} \tag{5.9}$$

This completes the proof. □

We note that convergence with probability 1 implies convergence in probability (see Appendix A.1). Thus, as $N \to \infty$, $\bar{g}_N(j)$ in (5.4) also converges to $g(j)$ in probability, i.e., for any $\delta > 0$ and $1 > \epsilon > 0$, there is an integer $N_{\delta,\epsilon}$ such that when $N > N_{\delta,\epsilon}$ we have

$$\mathcal{P}(|\bar{g}_N(j) - g(j)| > \delta) < \epsilon. \tag{5.10}$$

5.2.2 Sample Paths with a Fixed Number of Regenerative Periods

In this subsection, we study the case in which the number of regenerative periods N used in estimating the potentials in each iteration is fixed. We will see that because of the estimation error in $\bar{g}_N^d$, instead of using the maximum value of $f^d + P^d g^{d_k}$ in the policy-improvement step (4.5) in policy iteration Algorithm 4.1, it is more appropriate to use a small region for the expected potentials (cf. $\phi(g)$ and $\psi(g)$ defined in (5.11) and (5.13)).

First, to simplify the notation, for any S-dimensional vector g, we define

$$\phi(g) := \arg\left\{\max_{d\in\mathcal{D}}(f^d + P^d g)\right\} \subseteq \mathcal{D}. \tag{5.11}$$

Precisely, $\phi(g) := \times_{i=1}^S \phi_i(g)$, with

$$\phi_i(g) = \left\{\alpha \in \mathcal{A}(i) :\ f(i,\alpha) + \sum_{j=1}^S p^\alpha(j|i)g(j) \right.$$
$$\left. = \max_{\alpha'\in\mathcal{A}(i)}\left[f(i,\alpha') + \sum_{j=1}^S p^{\alpha'}(j|i)g(j)\right]\right\}.$$

With this notation, the optimality equation for ergodic chains (4.7) becomes:

$$\widehat{d} \in \phi(g^{\widehat{d}}).$$

The set of optimal policies is

$$\mathcal{D}_0 := \left\{d \in \mathcal{D}\ :\ d \in \phi(g^d)\right\}.$$

In addition, for any S-dimensional vector g and a small positive number $\nu > 0$, we set[1]

$$U_\nu(g) := \left[\max_{d\in\mathcal{D}}(f^d + P^d g) - \nu e,\ \max_{d\in\mathcal{D}}(f^d + P^d g)\right]. \tag{5.12}$$

Similar to (5.11), we define

$$\psi(g) := \left\{d : f^d + P^d g \in U_\nu(g)\right\} \tag{5.13}$$

as the set of improved policies. Precisely, we have $\psi(g) = \times_{i=1}^S \psi_i(g)$, with

[1] For any two S-dimensional vectors a and b with $a < b$, we use $[a,b]$ to denote an S-dimensional array of intervals $[a,b] := ([a(1),b(1)],[a(2),b(2)],\ldots,[a(S),b(S)])$. An S-dimensional vector $c \in [a,b]$ means that $c(i) \in [a(i),b(i)]$ for all $i = 1,2,\ldots,S$.

$$\psi_i(g) = \left\{ \alpha \in \mathcal{A}(i): \; f(i,\alpha) + \sum_{j=1}^{S} p^{\alpha}(j|i)g(j) \in \right.$$

$$\left. \left[\max_{\alpha' \in \mathcal{A}(i)} \left\{ f(i,\alpha') + \sum_{j=1}^{S} p^{\alpha'}(j|i)g(j) \right\} - \nu, \max_{\alpha' \in \mathcal{A}(i)} \left\{ f(i,\alpha') + \sum_{j=1}^{S} p^{\alpha'}(j|i)g(j) \right\} \right] \right\},$$

where the large square bracket denotes an interval. We certainly have

$$\phi(g) \subseteq \psi(g)$$

for any $\nu > 0$.

The Algorithm

The *sample-path-based policy iteration algorithm with a fixed* N works as follows.

Algorithm 5.1. A Sample-Path-Based Policy Iteration Algorithm With a Fixed N:

1. Choose an integer $N > 0$, a real number $\nu > 0$, and an initial policy d_0; Set $k = 0$.
2. Observe the system under policy d_k to obtain a sample path with N regenerative periods. Estimate the potentials using (5.4). Denote the estimates as $\bar{g}_N^{d_k}$. (Set $\bar{g}_N^{d_k}(j) = \bar{g}_N^{d_{k-1}}(j)$ if $N_k(j) = 0$, where $N_k(j)$ is the $N(j)$ in (5.3) in the kth iteration, with $\bar{g}^{d_{-1}} = 0$).
3. Choose any policy
$$d_{k+1} \in \psi(\bar{g}_N^{d_k}), \tag{5.14}$$
component-wisely. If at a state i, action $d_k(i)$ is in the set (5.14), then set $d_{k+1}(i) = d_k(i)$.
4. If $d_{k+1} = d_k$, then stop; otherwise, set $k := k+1$ and go to step 2.

There may be multiple policies in the set on the right-hand side of (5.14). If $d_k(i) \in \psi_i(\bar{g}_N^{d_k})$, then we choose $d_{k+1}(i) = d_k(i)$; otherwise, we may choose randomly in $\psi_i(\bar{g}_N^{d_k})$. We will see that if we choose $d_{k+1} \in \phi(\bar{g}_N^{d_k})$ in (5.14), then it will have some problems in setting the stopping criterion in step 4.

The Effect of the Estimation Errors

Because of the errors in estimating potentials, two issues need to be addressed for sample-path-based policy iteration algorithms. The first one is that, at each iteration, the "true" performance may not necessarily improve and the stopping criterion may not be met; thus, we have to study if the algorithm ever stops. The second issue is that, if it does stop, whether it stops at a "true" optimal policy.

The answers to these two questions depend on the following property.

For a set of finite real numbers $\mathcal{C} := \{c_1, c_2, \ldots, c_M\}$, define the distance of c_i and c_j as $\rho_{c_i c_j} \equiv \rho_{ij} := |c_i - c_j|$, $i, j = 1, 2, \ldots, M$ and set $\delta := \min\{\rho_{ij}, c_i \neq c_j,\ i, j = 1, \ldots, M\}$. If we know that two numbers, $x \in \mathcal{C}$ and $y \in \mathcal{C}$, satisfy $\rho_{xy} = |x - y| < \delta$, then they must be the same, i.e., $x = y$.

In an MDP with a finite number of policies, the average reward takes only a finite number of different values. Define

$$\sigma = \frac{1}{2} \min_{d, d' \in \mathcal{D}} \left\{ \left| \eta^d - \eta^{d'} \right| :\ \eta^d \neq \eta^{d'} \right\} \tag{5.15}$$

to be the minimum "distance" between any two policies. We have $\sigma > 0$ (if the average rewards of all policies are not the same). Therefore, if the absolute value of the difference in the average rewards of two policies in $\mathcal{D}$ is less than σ, then either the average rewards of these two policies are the same, or they are simply the same policy. Thus, if the estimation error is small enough, this error can be adjusted and it will not affect the outcome of the policy iteration. This fact is formally stated in Lemma 5.2 below.

At each iteration, let g^d be the true potential vector under the current policy d (we omitted the subscript k in d_k), $\bar{g}^d$ be its estimate, and η^d be the corresponding (true) average reward. Denote the error in the potential estimate as a vector $r := \bar{g}^d - g^d$. Let $h \in \psi(\bar{g}^d)$ be an (improved) policy that reaches the neighborhood of the maximum in (5.12) by using the estimate $\bar{g}^d$ as g, and let π^h and η^h be the (true) steady-state probability and the average reward of h, respectively. The policy h depends on the estimate $\bar{g}^d$.

Lemma 5.2. *We choose $\nu = \sigma/2$ in the sample-path-based policy iteration Algorithm 5.1. Suppose that the Markov chain under every policy is ergodic with a finite state space, and the number of policies is finite. Then, the following holds.*

(a) If the algorithm does not stop at an iteration and $|r| < (\sigma/2)e$,[2] then $\eta^h \geq \eta^d$; i.e., at this iteration, the performance does not decrease.
(b) If the algorithm stops at an iteration and $|r| < (\sigma/2)e$, then it stops at a (true) optimal policy.

[2] For an S-dimensional vector r, we define $|r| = (|r(1)|, |r(2)|, \ldots, |r(S)|)^T$.

Proof. (a) From the average-reward difference formula $\eta^h - \eta^d = \pi^h \left[(P^h - P^d)g^d + (f^h - f^d)\right]$, we have

$$\eta^h - \eta^d = \pi^h \left[(P^h - P^d)\bar{g}^d + (f^h - f^d) + (P^d - P^h)r\right]. \tag{5.16}$$

Because the iteration procedure does not stop at this iteration, according to (5.14), we have $h \in \psi(\bar{g}^d)$ and, therefore,

$$(P^h - P^d)\bar{g}^d + (f^h - f^d) + \nu e \geq 0.$$

Thus, $\pi^h \left[(P^h - P^d)\bar{g}^d + (f^h - f^d)\right] \geq -\nu$, then, from (5.16),

$$\eta^h - \eta^d \geq \pi^h (P^d - P^h)r - \nu.$$

However,

$$\begin{aligned} \left|\pi^h(P^d - P^h)r - \nu\right| &\leq \left|\pi^h P^d r\right| + \left|\pi^h P^h r\right| + \left|\nu\right| \\ < \frac{\sigma}{2} + \frac{\sigma}{2} + \frac{\sigma}{2} &< \min_{d,d' \in \mathcal{D}} \left\{ \left|\eta^d - \eta^{d'}\right| : \ \eta^d \neq \eta^{d'} \right\}. \end{aligned}$$

Therefore, $\eta^h - \eta^d > -\min_{d,d' \in \mathcal{D}} \left\{ \left|\eta^d - \eta^{d'}\right| : \ \eta^d \neq \eta^{d'} \right\}$. This is only possible if $\eta^h \geq \eta^d$.

(b) Suppose that the algorithm stops at an iteration, and the policy at this iteration is denoted as $\widehat{d}$. Let $\bar{g}^{\widehat{d}}$ be the estimate of its potential. Then, from (5.14), for any policy $d \in \mathcal{D}$, we have

$$(P^d - P^{\widehat{d}})\bar{g}^{\widehat{d}} + (f^d - f^{\widehat{d}}) \leq \nu e.$$

Then

$$\begin{aligned} \eta^d - \eta^{\widehat{d}} &= \pi^d \left[(P^d - P^{\widehat{d}})\bar{g}^{\widehat{d}} + (f^d - f^{\widehat{d}}) + (P^{\widehat{d}} - P^d)r\right] \\ &\leq \pi^d (P^{\widehat{d}} - P^d)r + \nu. \end{aligned}$$

Thus, $\eta^d - \eta^{\widehat{d}} \leq (3\sigma)/2$, and hence $\eta^d \leq \eta^{\widehat{d}}$, for all policies $d \in \mathcal{D}$. That is, $\eta^{\widehat{d}}$ is the true optimal average reward. □

The next lemma shows that if the estimation error $|r| = |\bar{g} - g|$ is small enough, the policy iteration using the potential estimate $\bar{g}$ can be viewed as if the true potential g is used. First, we define

$$\begin{aligned} \kappa = \frac{1}{2} \min_{\text{all } d,h,h' \in \mathcal{D}} \Big\{ & \left|(f^h + P^h g^d)(i) - (f^{h'} + P^{h'} g^d)(i)\right| : \qquad \text{all } i \in \mathcal{S}, \\ & \text{with } \left[(f^h + P^h g^d)(i) - (f^{h'} + P^{h'} g^d)(i)\right] \neq 0 \Big\}. \end{aligned}$$

Because there is only a finite number of policies and the state space is finite, we have $\kappa > 0$.

Lemma 5.3. *We choose $\nu = \kappa/2$ in the sample-path-based policy iteration Algorithm 5.1. Suppose that the Markov chain under every policy is ergodic with a finite state space, and the number of policies is finite. If $|r| = |\bar{g}^d - g^d| < (\kappa/2)e$, where g^d and $\bar{g}^d$ are the potential of policy $d \in \mathcal{D}$ and its estimate, then*

$$\psi(\bar{g}^d) \subseteq \phi(g^d).$$

Proof. Let $h \in \phi(g^d)$ and $h' \in \psi(\bar{g}^d)$. By the definition of $\phi(g)$ in (5.11), we have $f^h + P^h g^d \geq f^{h'} + P^{h'} g^d$. By the definition of $\psi(g)$ in (5.13), we have $f^{h'} + P^{h'} \bar{g}^d \geq f^h + P^h \bar{g}^d - \nu e$. From this equation, we have

$$f^{h'} + P^{h'} g^d + (P^{h'} - P^h)(\bar{g}^d - g^d) \geq f^h + P^h g^d - \nu e.$$

Therefore,

$$(f^h + P^h g^d) - (f^{h'} + P^{h'} g^d) \leq (P^{h'} - P^h)(\bar{g}^d - g^d) + \nu e.$$

This, together with $f^h + P^h g^d \geq f^{h'} + P^{h'} g^d$, leads to

$$\left|(f^h + P^h g^d) - (f^{h'} + P^{h'} g^d)\right| \leq \left|(P^{h'} - P^h)(\bar{g}^d - g^d)\right| + \nu e. \tag{5.17}$$

From (5.17), if $|r| = |\bar{g}^d - g^d| < (\kappa/2)e$ and $\nu = \kappa/2$, then $|(f^h + P^h g^d) - (f^{h'} + P^{h'} g^d)| < (2\kappa)e$. By the definition of κ, we must have $f^h + P^h g^d = f^{h'} + P^{h'} g^d$. In other words, $h' \in \phi(g^d)$. Thus, $\psi(\bar{g}^d) \subseteq \phi(g^d)$. □

Note that $\psi(\bar{g}^d)$ may be smaller than $\phi(g^d)$. The implication of this lemma is as follows. Suppose that, at every iteration, the estimation error is $|r| < (\kappa/2)e$. If the sample-path-based algorithm does not stop at an iteration, then the improved policy picked up by using the estimated potentials with (5.14) is one of the policies that may be chosen by the standard policy iteration with the exact potentials. If the sample-path-based iteration stops at a policy $\widehat{d}$, then we have $\widehat{d} \in \psi(\bar{g}^{\widehat{d}})$ and by Lemma 5.3, we have $\widehat{d} \in \phi(g^{\widehat{d}})$; i.e, it will stop if the true potentials are used.

However, because of the random error in the estimates, we do not know if $\widehat{d} \in \psi(\bar{g}^{\widehat{d}})$, although $\widehat{d} \in \phi(g^{\widehat{d}})$; i.e., we do not know if the algorithm will stop even if it reaches an optimal policy. We may determine its probability. Suppose that $d_k = \widehat{d}$ is an optimal policy. Then, according to (5.13), the probability that the algorithm stops at this iteration is

$$p_0 := \mathcal{P}\left\{ f^{\widehat{d}} + P^{\widehat{d}} \bar{g}^{\widehat{d}} \geq \max_{d \in \mathcal{D}} \left[f^d + P^d \bar{g}^{\widehat{d}} \right] - \nu e \right\},$$

where

$$\begin{aligned} &\max_{d \in \mathcal{D}} \left[f^d + P^d \bar{g}^{\widehat{d}} \right] \\ &= \max_{d \in \mathcal{D}} \left\{ f^d + P^d g^{\widehat{d}} + P^d \left[\bar{g}^{\widehat{d}} - g^{\widehat{d}} \right] \right\} \end{aligned}$$

$$\begin{aligned}&\leq \max_{d\in\mathcal{D}}\left\{f^d+P^d g^{\widehat{d}}\right\}+\max_{d\in\mathcal{D}}\left\{P^d\left[\bar{g}^{\widehat{d}}-g^{\widehat{d}}\right]\right\}\\&=\left[f^{\widehat{d}}+P^{\widehat{d}}g^{\widehat{d}}\right]+\max_{d\in\mathcal{D}}\left\{P^d\left[\bar{g}^{\widehat{d}}-g^{\widehat{d}}\right]\right\}.\end{aligned}$$

Thus,

$$p_0 \geq \mathcal{P}\left\{P^{\widehat{d}}\left[\bar{g}^{\widehat{d}}-g^{\widehat{d}}\right] \geq \max_{d\in\mathcal{D}} P^d\left[\bar{g}^{\widehat{d}}-g^{\widehat{d}}\right]-\nu e\right\}. \tag{5.18}$$

We wish to find out under what condition this probability is positive. Let $\widehat{r} = \bar{g}^{\widehat{d}} - g^{\widehat{d}}$. Suppose that $|\widehat{r}| < (\nu/2)e$. Then, we have $\max_{d\in\mathcal{D}}\left\{P^d\left[\bar{g}^{\widehat{d}}-g^{\widehat{d}}\right]\right\} < (\nu/2)e$ and $\max_{d\in\mathcal{D}}\left\{P^d\left[\bar{g}^{\widehat{d}}-g^{\widehat{d}}\right]-\nu e\right\} < -(\nu/2)e$. On the other hand, we have $\left|P^{\widehat{d}}\left[\bar{g}^{\widehat{d}}-g^{\widehat{d}}\right]\right| < (\nu/2)e$. Thus, we have

$$P^{\widehat{d}}\left[\bar{g}^{\widehat{d}}-g^{\widehat{d}}\right] > -(\nu/2)e > \max_{d\in\mathcal{D}}\left\{P^d\left[\bar{g}^{\widehat{d}}-g^{\widehat{d}}\right]-\nu e\right\}.$$

Therefore, from (5.18) we have

$$p_0 \geq \mathcal{P}\left[|\widehat{r}| < (\nu/2)e\right]. \tag{5.19}$$

Convergence Property

As shown in (5.9), as the length of a sample path in each iteration $N \to \infty$, the estimate in each iteration converges with probability 1 to the exact value of g. Thus, by Lemmas 5.2 and 5.3, we can show that the sample-path-based policy iteration stops with probability 1 if N is large enough, and it stops at the optimal policy in probability as N goes to infinity.

Theorem 5.1. Convergence Property with Fixed Lengths

We choose $\nu = \min\{\sigma/2, \kappa/2\}$ in the sample-path-based policy iteration Algorithm 5.1. Suppose that the Markov chain under every policy is ergodic with a finite state space, and the number of policies is finite. Then, the following holds.

(*a*) When the length of the sample path N is large enough, the sample-path-based policy iteration (Algorithm 5.1) stops with probability 1.

(*b*) Let η^* be the true optimal average reward and η^*_N be the average reward of the "optimal" policy given by the sample-path-based policy iteration (Algorithm 5.1) with N regenerative periods in each iteration. Then,
$$\lim_{N\to\infty}\mathcal{P}(\eta^*_N=\eta^*)=1.$$

Proof. (a) can be proved by Lemma 5.3 and (5.19). Because there are only a finite number of states and a finite number of policies, from (5.10), for any $1 > \epsilon > 0$, there is an $N_{\frac{\nu}{2},\epsilon}$ such that if $N > N_{\frac{\nu}{2},\epsilon}$ then

$$\mathcal{P}\left\{\left|\bar{g}_N^d(j) - g^d(j)\right| > \frac{\nu}{2}\right\} < \epsilon$$

holds for all $j \in \mathcal{S}$ and all $d \in \mathcal{D}$. Thus, for this $\nu > 0$, if N is large enough (meaning $N > N_{\frac{\nu}{2},\epsilon}$), we have $|\bar{g}_N^d - g^d| < (\nu/2)e \leq (\kappa/2)e$ with probability $p > 1-\epsilon > 0$ for all $d \in \mathcal{D}$. Therefore, from Lemma 5.3, we have $\psi(\bar{g}_N^d) \subseteq \phi(g^d)$ with probability $p > 0$ for all $d \in \mathcal{D}$.

Therefore, if N is large enough, then at each iteration with probability $p > 0$, the sample-path-based policy iteration Algorithm 5.1 produces a correct and improved policy d_{k+1} in its step 3, which may be chosen by the standard policy iteration algorithm using the true potentials, and the average reward improves if the algorithm does not stop at the iteration.

Suppose that there are K different values of the average rewards η corresponding to all the policies in $\mathcal{D}$. If we have K consecutive iterations, and, in each of them, the sample-path-based algorithm produces a correct policy (i.e.,with a better performance), then the sample-path-based policy iteration process must reach the set of optimal policies $\mathcal{D}_0$. Now, we group every $K+1$ iterations together in the policy iteration sequence: The first group consists of the first $K+1$ iterations, the second group consists of the second $K+1$ iterations, and so on. As discussed above, the probability that the sample-path-based algorithm produces a correct policy in every iteration in the first K iterations in the same group is larger than $p^K > 0$. Thus, the probability that the policy at the $(K+1)$th iteration is an optimal policy, denoted as $\widehat{d}$, is larger than $p^K > 0$. Once the algorithm reaches an optimal policy, we may apply (5.19). That is, under the condition that the algorithm reaches an optimal policy, the probability that the algorithm stops at the $(K+1)$th iteration is $p_0 > 0$. Therefore, the policy iteration algorithm does not stop at any group is less than $q = 1 - p^K p_0 < 1$. Thus, the probability that policy iteration does not stop at the first L groups is less than q^L, which goes to zero as $L \rightarrow \infty$. That is, the probability that policy iteration never stops is zero if N is large enough.

For (b), note that η_N^* is a random variable depending on the sample path. We need to prove that for any $\epsilon > 0$, there is an integer $N_\epsilon > 0$ such that if $N > N_\epsilon$ then

$$\mathcal{P}(\eta_N^* \neq \eta^*) < \epsilon. \tag{5.20}$$

Recall that, in Lemma 5.2 and (5.15), we have

$$\sigma = \frac{1}{2} \min_{d,d' \in \mathcal{D}} \left\{\left|\eta^d - \eta^{d'}\right| : \ \eta^d \neq \eta^{d'}\right\}.$$

Because there is only a finite number of policies, from (5.10), there is an N_ϵ such that if $N > N_\epsilon$ then the probability that the potential-estimation error

$|r| > (\sigma/2)e$ for all policies is less than ϵ. Then, (5.20) follows directly from Lemma 5.2(b). □

Some comments are in order.

1. Because $\phi(g) \subseteq \psi(g)$, we may choose $d_{k+1} \in \phi(g^{d_k})$ (i.e., set $\nu = 0$ in (5.12)) in step 3 of Algorithm 5.1 to replace (5.14). If we do so, the average reward does not decrease at each iteration if the estimation error is small enough. However, we will meet a problem for choosing a stopping criterion: The condition $d_k \in \phi(\bar{g}^{d_k})$ may not hold even if $d_k = \widehat{d}$ is an optimal policy. This can be explained as follows. Suppose that there are two optimal policies $\widehat{d}, d' \in \mathcal{D}_0$. Then, we have $f^{\widehat{d}} + P^{\widehat{d}} g^{\widehat{d}} = f^{d'} + P^{d'} g^{\widehat{d}}$. Because of the error in $\bar{g}^{\widehat{d}}$, it is entirely possible that $f^{d'} + P^{d'} \bar{g}^{\widehat{d}} \succeq f^{\widehat{d}} + P^{\widehat{d}} \bar{g}^{\widehat{d}}$, and thus $\widehat{d} \notin \phi(\bar{g}^{\widehat{d}})$. That means that, if we choose $d_{k+1} \in \phi(\bar{g}^{d_k})$, the algorithm may not stop even if it reaches an optimal policy.
2. Because the probability of the estimation error $r = \bar{g}^d - g^d$ may be widely distributed, it is clear that for any fixed N, the probability that the error of a potential estimate is larger than any $\delta > 0$ is positive. Thus, no matter how large N is, the probability that the fixed-length sample-path-based policy iteration does not stop at the true optimal policy is positive. This means that any algorithm with a fixed N cannot converge to the true optimal with probability 1.
3. If we use a sequence of increasing numbers of regenerative periods, N_1, $N_2, \ldots, N_{k+1} > N_k$, in the iteration, we may face the problem that, at some iterations, the algorithm stops at a false optimal policy because the improved policy is the same as the original one ($i.e., d_k \in \psi(\bar{g}^{d_k}_{N_k})$). This probability may be large at the beginning of the iteration procedure when N_k is small. Therefore, if we use a sequence of increasing integers N_k, we should let the iteration continue even if we obtain the same policy in some iterations, i.e., even if $d_{k+1} = d_k$. In the next subsection, we will prove that under some conditions for the sequence of N_k, the policy iteration, if we let it continue even if $d_{k+1} = d_k$, converges to the true optimal policy either in probability or with probability 1.

5.2.3 Sample Paths with Increasing Lengths

The Algorithm

As discussed at the end of the last subsection, in order to converge with probability 1 to an optimal policy, the policy iteration algorithm with an increasing number of regenerative periods in each iteration should never stop. Because we do not need to set a stopping criterion, we may use $d_{k+1} \in \phi(\bar{g}^{d_k}_{N_k})$ in the policy improvement step.

The algorithm is stated as follows.

Algorithm 5.2 A Sample-Path-Based Policy Iteration Algorithm with Increasing Lengths:

1. Choose a sequence of integers, $N_0, N_1, \dots$, with $N_{k+1} \geq N_k$, $k = 0, 1, 2, \dots$, $\lim_{k\to\infty} N_k = \infty$. Set $k = 0$. Choose an initial policy d_0.
2. Observe the system with d_k for N_k regenerative periods. Estimate the potentials using (5.4). Denote the estimates as $\bar{g}_{N_k}^{d_k}$.
3. Choose
$$d_{k+1} \in \phi(\bar{g}_{N_k}^{d_k}) = \arg\left\{\max_{d\in\mathcal{D}}\left[f^d + P^d \bar{g}_{N_k}^{d_k}\right]\right\},$$
component-wisely. (If there is more than one policy in $\phi(\bar{g}_{N_k}^{d_k})$, we may randomly choose one of them.)
4. Set $k := k + 1$; go to step 2.

No stopping criterion is used in the algorithm because it never stops. Thus, in step 3, there is no requirement to set $d_{k+1}(i) = d_k(i)$ whenever possible (as Algorithm 5.1 does). One implication of this change is that after the algorithm reaches an optimal policy, it may oscillate among different optimal policies even if the accurate values of the potentials are used.

The General Conditions for Convergence

The algorithm produces a sequence of policies denoted as $d_0, d_1, \dots, d_k, \dots$, and we now study its convergence property. We first study the probability of a wrong decision because of the errors in the potential estimates. We define

$$q(N, d) = \mathcal{P}\left[\phi(\bar{g}_N^d) \subseteq \phi(g^d)\right].$$

This is the probability that the estimated potential will definitely lead to the right choice of the improved policy. Indeed, if at the kth iteration $\phi(\bar{g}_{N_k}^{d_k}) \subseteq \phi(g^{d_k})$, then the improved policy based on the estimated potential is one policy that could be chosen if the true potential were used. We denote it as $d_{k+1} \in \phi(\bar{g}_{N_k}^{d_k}) \subseteq \phi(g^{d_k})$. Thus, we have

$$q(N_k, d_k) \leq \mathcal{P}\left[d_{k+1} \in \phi(g^{d_k}) | d_k\right]. \tag{5.21}$$

We need the following lemma.

Lemma 5.4. Convergence of Products of Infinite Many Numbers: If $\sum_{k=0}^{\infty}(1 - y_k) < \infty$ and $0 \leq y_k \leq 1$ for all k, then $\lim_{n\to\infty}\prod_{k\geq n} y_k = 1$.

Proof. Set $x_k = 1 - y_k$, $k = 0, 1, \ldots$. Then $0 \leq x_k \leq 1$ and $\sum_{k=0}^{\infty} x_k < \infty$. Because $x_k \geq 0$, then $\sum_{k=0}^{n} x_k$ is nondecreasing, and it must converge to a finite number as $n \to \infty$. Thus,

$$\lim_{n\to\infty} \sum_{k\geq n} x_k = 0. \tag{5.22}$$

Next, for any $0 \leq x < 1$, we have the MacLaurin series

$$\ln(1-x) = -x(1 + \frac{x}{2} + \frac{x^2}{3} + \cdots).$$

If $0 \leq x < \frac{1}{2}$, then $1 \leq 1 + \frac{x}{2} + \frac{x^2}{3} + \cdots \leq 1 + x + x^2 + \cdots < 1 + \frac{1}{2} + \frac{1}{2^2} + \cdots = 2$, and

$$-2x < \ln(1-x) \leq -x. \tag{5.23}$$

From (5.22), we can assume that $x_n < \frac{1}{2}$ if n is large enough. Therefore, it follows from (5.23) that if n is large enough, we have

$$-2\sum_{k\geq n} x_k \leq \sum_{k\geq n} \{\ln(1-x_k)\} \leq -\sum_{k\geq n} x_k.$$

From (5.22), we get $\lim_{n\to\infty} \sum_{k\geq n} \{\ln(1-x_k)\} = 0$. The lemma then follows from $\prod_{k\geq n} y_k = \prod_{k\geq n}(1-x_k) = \exp\left\{\sum_{k\geq n} [\ln(1-x_k)]\right\}$. □

We are now ready to give sufficient conditions for the sample-path-based policy iteration algorithm to reach the set of optimal policies and remain there indefinitely with probability 1 (the proof here follows [88] with some modifications).

Theorem 5.2. Convergence Property with Increasing Lengths

Consider the sample-path-based policy iteration Algorithm 5.2 starting from an initial policy d_0. If the sample paths in different iterations are independently generated and

$$\sum_{k=0}^{\infty}(1-q_k) < \infty, \tag{5.24}$$

where $q_k := \min_{d\in\mathcal{D}} q(N_k, d)$, then there exists an almost surely finite random integer L such that

$$\mathcal{P}(d_k \in \mathcal{D}_0, \text{for all } k \geq L) = 1.$$

Proof. Denote the underlying probability space as Ω. Any point $\omega \in \Omega$ represents all the sample paths (with policies $d_0, d_1, \ldots$, and lengths $N_0, N_1, \ldots$)

generated in one run of the policy iteration with the initial policy d_0. Every variable or quantity observed in a policy iteration run depends on ω; e.g., we may denote the policy used in its kth iteration as $d_k = d_k(\omega)$. Define

$$L(\omega) = \min\{l : \ d_k \in \mathcal{D}_0 \text{ for all } k \geq l\},$$

provided that the set of integers $\{l : d_k \in \mathcal{D}_0, \text{for all } k \geq l\}$, which depends on ω, is non-null ($\neq \emptyset$). To simplify the notation, we denote

$$\begin{aligned} &\{(l : \ d_k \in \mathcal{D}_0 \text{ for all } k \geq l) \neq \emptyset\} \\ := &\{\omega \in \Omega : (l : \ d_k \in \mathcal{D}_0 \text{ for all } k \geq l) \neq \emptyset\} \subseteq \Omega, \end{aligned}$$

and similar expressions will be used. It suffices to prove

$$\mathcal{P}\{(l : \ d_k \in \mathcal{D}_0 \text{ for all } k \geq l) \neq \emptyset\} = 1,$$

or

$$\mathcal{P}\{\exists \ l : d_k \in \mathcal{D}_0 \text{ for all } k \geq l\} = 1.$$

For any integer $n \geq 0$, define $A_n := \{d_k \in \mathcal{D}_0, \text{ for all } k \geq n\} \subseteq \Omega$. Then, we have $A_n \subseteq A_{n+1}$, $n \geq 0$, and $\{\exists \ l : d_k \in \mathcal{D}_0 \text{ for all } k \geq l\} = \cup_{n \geq 0} A_n$. Hence,

$$\mathcal{P}\{\exists \ l : d_k \in \mathcal{D}_0 \text{ for all } k \geq l\} = \mathcal{P}(\cup_{n\geq 0} A_n) = \lim_{n\to\infty} \mathcal{P}(A_n).$$

Let $K < \infty$ be the number of all policies in $\mathcal{D}$. As proved in Section 4.1.1, in policy iteration with accurate potentials, policies do not repeat in the iteration procedure before it reaches an optimal policy, and once it reaches $\mathcal{D}_0$, it stays there forever. Thus, if $d_{k+1} \in \phi(g^{d_k})$ for consecutive K iterations, then the policy iteration must reach $\mathcal{D}_0$. Therefore, if $d_{k+1} \in \phi(g^{d_k})$ for all $k \geq n - K$, $n \geq K$, then we have $d_k \in \mathcal{D}_0$ for all $k \geq n$. Thus,

$$\left\{d_{k+1} \in \phi(g^{d_k}) \text{ for all } k \geq n - K\right\} \subseteq A_n.$$

Therefore,

$$\mathcal{P}(A_n) \geq \mathcal{P}\left\{d_{k+1} \in \phi(g^{d_k}) \text{ for all } k \geq n - K\right\}.$$

Next, given any sequence of policies $d_0, d_1, \ldots$, the potential estimates at different iterations are independently generated. Note, however, that d_{k+1} depends on d_k, $k = 0, 1, \ldots$. For any d_{n-K}, we have

$$\begin{aligned} &\mathcal{P}\left\{d_{k+1} \in \phi(g^{d_k}) \text{ for all } k \geq n - K | d_{n-K}\right\} \\ = &\Bigg\{ \sum_{d_{n-K+1} \in \phi(g^{d_{n-K}})} \{\mathcal{P}\left[d_{k+1} \in \phi(g^{d_k}) \text{ for all } k \geq n - K + 1 | d_{n-K+1}\right] \\ &\mathcal{P}\big[d_{n-K+1} | d_{n-K+1} \in \phi(g^{d_{n-K}})\big]\big\} \Bigg\} \mathcal{P}\big[d_{n-K+1} \in \phi(g^{d_{n-K}}) | d_{n-K}\big], \end{aligned} \tag{5.25}$$

where $\mathcal{P}\left[d_{n-K+1}|d_{n-K+1}\in\phi(g^{d_{n-K}})\right]$ is the conditional probability of d_{n-K+1} given that $d_{n-K+1}\in\phi(g^{d_{n-K}})$. In addition, we have

$$\begin{aligned}
&\mathcal{P}\left\{d_{k+1}\in\phi(g^{d_k}) \text{ for all } k\geq n-K+1|d_{n-K+1}\right\}\\
=&\left\{\sum_{d_{n-K+2}\in\phi(g^{d_{n-K+1}})}\left\{\mathcal{P}\left[d_{k+1}\in\phi(g^{d_k}) \text{ for all } k\geq n-K+2|d_{n-K+2}\right]\right.\right.\\
&\left.\left.\times\,\mathcal{P}\left[d_{n-K+2}|d_{n-K+2}\in\phi(g^{d_{n-K+1}})\right]\right\}\right\}\\
&\times\,\mathcal{P}\left\{d_{n-K+2}\in\phi(g^{d_{n-K+1}})|d_{n-K+1}\right\}
\end{aligned}$$

Continuing this process, we obtain

$$\begin{aligned}
&\mathcal{P}\left\{d_{k+1}\in\phi(g^{d_k}) \text{ for all } k\geq n-K|d_{n-K}\right\}\\
=&\quad\left\{\prod_{k=n-K}^{\infty}\mathcal{P}\left\{d_{k+1}\in\phi(g^{d_k})|d_k\right\}\right\}\\
&\times\left\{\sum_{d_{k+1}\in\phi(g^{d_k}),\ k\geq n-K}\ \prod_{k=n-K}^{\infty}\mathcal{P}\left\{d_{k+1}|d_{k+1}\in\phi(g^{d_k})\right\}\right\}. \qquad (5.26)
\end{aligned}$$

From (5.21), we have $\mathcal{P}\left\{d_{k+1}\in\phi(g^{d_k})|d_k\right\}\geq q_k$. Also, we have

$$\sum_{d_{k+1}\in\phi(g^{d_k}),\ k=n-K,\ldots}\left\{\prod_{k=n-K}^{\infty}\mathcal{P}\left\{d_{k+1}|d_{k+1}\in\phi(g^{d_k})\right\}\right\}=1.$$

Finally, from (5.25) and (5.26), we get, for any d_{n-K}, that

$$\mathcal{P}\left\{d_{k+1}\in\phi(g^{d_k}) \text{ for all } k\geq n-K|d_{n-K}\right\}\geq\prod_{k\geq n-K}q_k.$$

Thus, with any initial policy d_0, we have

$$\mathcal{P}\left\{d_{k+1}\in\phi(g^{d_k}) \text{ for all } k\geq n-K\right\}\geq\prod_{k\geq n-K}q_k.$$

By (5.24) and Lemma 5.4, we have $\lim_{n\to\infty}\prod_{k\geq n-K}q_k=1$. Thus, $\lim_{n\to\infty}\mathcal{P}(A_n)=1$ and the theorem holds. □

Theorem 5.2 means that the sample-path-based policy iteration algorithm converges with probability 1 to the set of optimal policies if condition (5.24) holds. We will see that, to meet this condition, the length of the sample path N_k must increase fast enough. However, we have a weaker result under a weaker condition.

Theorem 5.3. *If the sample paths in different iterations are independently generated, and* $\lim_{k\to\infty} q_k = 1$*, where* $q_k := \min_{d\in\mathcal{D}} q(N_k, d)$*, then*

$$\lim_{n\to\infty} \mathcal{P}(d_n \in \mathcal{D}_0) = 1.$$

Proof. From the proof of Theorem 5.2, if $d_{k+1} \in \phi(d_k)$ for $k = n-K, n-K+1, \ldots, n-1$, then $d_n \in \mathcal{D}_0$. Thus, we have

$$\mathcal{P}(d_n \in \mathcal{D}_0) \geq \prod_{k=n-K}^{n-1} q_k.$$

The theorem follows directly from $\lim_{k\to\infty} q_k = 1$. □

More Specific Conditions

The conditions in Theorems 5.2 and 5.3 are not very easy to verify directly, so we need some further work. First, we observe that, as discussed in the last subsection, because the policy space is finite, small errors in potential estimates can be corrected. This leads to the following lemma. To simplify the notation, for any S-dimensional vector v, we define $||v|| = \max_{i\in\mathcal{S}} |v(i)|$.

Lemma 5.5. *There exists a* $\delta > 0$ *such that if*

$$\sum_{k=0}^{\infty} \max_{d\in\mathcal{D}} \mathcal{P}(||\bar{g}^d_{N_k} - g^d|| > \delta) < \infty,$$

then condition (5.24) holds.

Proof. By Lemma 5.3 and $\phi(\bar{g}^d_N) \subseteq \psi(\bar{g}^d_N)$, for any policy d, there is a $\delta^d > 0$, such that if $||\bar{g}^d_N - g^d|| \leq \delta^d$, then $\phi(\bar{g}^d_N) \subseteq \phi(g^d)$. Set $\delta := \min_{d\in\mathcal{D}} \delta^d > 0$.

$$\begin{aligned} q(N,d) &= \mathcal{P}\left[\phi(\bar{g}^d_N) \subseteq \phi(g^d)\right] \\ &\geq \mathcal{P}(||\bar{g}^d_N - g^d|| \leq \delta^d) \geq \mathcal{P}(||\bar{g}^d_N - g^d|| \leq \delta). \end{aligned}$$

Thus, $1 - q(N,d) \leq 1 - \mathcal{P}(||\bar{g}^d_N - g^d|| \leq \delta) = \mathcal{P}(||\bar{g}^d_N - g^d|| > \delta)$, and

$$\begin{aligned} 1 - \min_{d\in\mathcal{D}} q(N,d) &= \max_{d\in\mathcal{D}} \{1 - q(N,d)\} \\ \leq \max_{d\in\mathcal{D}} \mathcal{P}(||\bar{g}^d_N - g^d|| > \delta). \end{aligned}$$

From this, we have

$$1 - q_k \leq \max_{d\in\mathcal{D}} \mathcal{P}(||\bar{g}^d_{N_k} - g^d|| > \delta).$$

Condition (5.24) now follows directly. □

Note that we not only proved the lemma, but also found the δ required in the lemma. The next lemma follows immediately.

Lemma 5.6. *Suppose that $\bar{g}_{N_k}$ is an unbiased estimate of g. If*

$$E\left[\bar{g}^d_{N_k}(j) - g^d(j)\right]^2 \le \frac{c^d}{N_k}, \qquad \textit{for all } d \in \mathcal{D} \textit{ and } j \in \mathcal{S},$$

$c^d > 0$, and

$$\sum_{k=1}^{\infty} \frac{1}{N_k} < \infty,$$

then condition (5.24) holds.

Proof. By Chebychev's inequality, for any $\delta > 0$, we have

$$\begin{aligned}
&\mathcal{P}(||\bar{g}^d_{N_k} - g^d|| > \delta) = \mathcal{P}\left[\cup_{j\in\mathcal{S}}\left\{|\bar{g}^d_{N_k}(j) - g^d(j)| > \delta\right\}\right] \\
&\le \sum_{j\in\mathcal{S}} \mathcal{P}(|\bar{g}^d_{N_k}(j) - g^d(j)| > \delta) \le \sum_{j\in\mathcal{S}} \frac{E\left[\bar{g}^d_{N_k}(j) - g^d(j)\right]^2}{\delta^2} \\
&\le \frac{c^d S}{N_k \delta^2}.
\end{aligned}$$

Since $\mathcal{D}$ is finite, we may set $c = \max_{d\in\mathcal{D}} c^d < \infty$. Therefore, for any $\delta > 0$ we have

$$\max_{d\in\mathcal{D}} \mathcal{P}(||\bar{g}^d_{N_k} - g^d|| > \delta) < \frac{cS}{N_k \delta^2}.$$

Now, let us choose δ as the one that satisfies Lemma 5.5. Then, condition (5.24) holds. □

Note that the conditions in this lemma can be changed to $E\left[\bar{g}^d_{N_k}(j) - g^d(j)\right]^2 \le c^d \kappa(N_k)$ for all $j \in \mathcal{S}$ and $d \in \mathcal{D}$, $c^d > 0$, where $\kappa(N)$ is a non-negative function of N, and $\sum_{k=1}^{\infty} \kappa(N_k) < \infty$.

Convergence of the Algorithm with Estimate (5.4)

We now study the policy iteration algorithms that are based on a particular estimate (5.4). We first note that for any finite N, $\bar{g}_N$ in (5.4) is biased because $E[\bar{\eta}_N] \neq \eta$. To get some insight, we first simplify the problem by using the unbiased potential estimate

$$\widetilde{g}_N(j) = \frac{1}{N}\left\{\sum_{k=1}^{N} \widetilde{V}_k(i^*, j)\right\}, \tag{5.27}$$

with $\widetilde{V}_k(i^*, j)$ defined in (5.6). To simplify the discussion, we assume that $\chi_k(j) = 1$ for every regenerative period. We want to apply Lemma 5.6. The first condition can be easily verified as follows. Because all $\widetilde{V}_k$, $k = 0, 1, \ldots, N$, are independent and $E(\widetilde{V}_k) = g^d$, we have, for any policy d,

$$E\left[\widetilde{g}_N^d(j) - g^d(j)\right]^2 = E\left[\frac{1}{N}\sum_{k=1}^{N}\widetilde{V}_k^d(i^*,j) - g^d(j)\right]^2$$

$$= \frac{1}{N^2}E\left\{\sum_{k=1}^{N}\left[\widetilde{V}_k^d(i^*,j) - g^d(j)\right]\right\}^2 = \frac{1}{N}\left\{E\left[\widetilde{V}_k^d(i^*,j)\right]^2 - \left[g^d(j)\right]^2\right\}. \quad (5.28)$$

Next, because

$$\left|\widetilde{V}_k^d(i^*,j)\right| \le \max_{i\in\mathcal{S}}\left|f(i,d(i)) - \eta^d\right|\left[l_k^d(i^*) - l_{k-1}^d(i^*)\right]$$

and

$$E\left[l_k^d(i^*) - l_{k-1}^d(i^*)\right]^2 < \infty$$

for finite ergodic chains, we have

$$E\left[\widetilde{V}_k^d(i^*,j)\right]^2 < \infty.$$

Thus, the first condition in Lemma 5.6 holds. Therefore, from Theorem 5.2 and Lemma 5.6, the sample-path-based policy iteration Algorithm 5.2 with potential estimate (5.27) converges with probability 1 to the optimal policy if $\sum_{k=1}^{\infty}\frac{1}{N_k} < \infty$.

Next, we consider the biased estimate (5.4). Lemma 5.6 cannot be applied, and we need to use Lemma 5.5. First, we study the bias of the potential estimate. Set $\varDelta_N(j) := \left|E\left[\bar{g}_N^d(j)\right] - g^d(j)\right|$ and $\varDelta_N := \max_{j\in\mathcal{S}}\varDelta_N(j)$. Because the regenerative periods are independent, we have (from (5.27)):

$$E\left[\bar{g}_N^d(j)\right] = \frac{1}{N}\sum_{k=1}^{N}E\left\{\sum_{l=l_k^d(j)}^{l_{k+1}^d(i^*)-1}\left[f(X_l,d(X_l)) - \bar{\eta}_N^d\right]\right\}$$

$$= \frac{1}{N}\sum_{k=1}^{N}\left\{E\left[\sum_{l=l_k^d(j)}^{l_{k+1}^d(i^*)-1}\left[f(X_l,d(X_l)) - \eta^d\right]\right]\right.$$

$$\left.+E\left[\sum_{l=l_k^d(j)}^{l_{k+1}^d(i^*)-1}\left[\eta^d - \bar{\eta}_N^d\right]\right]\right\}$$

$$= g^d(j) + \frac{1}{N}\sum_{k=1}^{N}E\left\{\sum_{l=l_k^d(j)}^{l_{k+1}^d(i^*)-1}\left[\eta^d - \bar{\eta}_N^d\right]\right\}.$$

Thus,

$$\varDelta_N(j) = \frac{1}{N}\sum_{k=1}^{N}E\left\{\sum_{l=l_k^d(j)}^{l_{k+1}^d(i^*)-1}\left[\eta^d - \bar{\eta}_N^d\right]\right\} = E\left\{\sum_{l=l_k^d(j)}^{l_{k+1}^d(i^*)-1}\left[\eta^d - \bar{\eta}_N^d\right]\right\},$$

for any fixed k. Because there are a finite number of states and actions, we have $|f(i,d(i))| < R < \infty$ for some $R > 0$, all $i \in \mathcal{S}$, and all $d \in \mathcal{D}$. Therefore, from (5.2), we have $\bar{\eta}_N^d < R$. Thus,

$$\left| \sum_{l=l_k^d(j)}^{l_{k+1}^d(i^*)-1} \left[\eta^d - \bar{\eta}_N^d\right] \right| < (\eta^d + R)\left[l_{k+1}^d(i^*) - l_k^d(i^*)\right],$$

with $E\left[l_{k+1}^d(i^*) - l_k^d(i^*)\right] < \infty$. Thus, by applying the Lebesgue dominated convergence theorem [28], we have

$$\lim_{N\to\infty} \varDelta_N(j) = E\left\{ \lim_{N\to\infty} \sum_{l=l_k^d(j)}^{l_{k+1}^d(i^*)-1} \left[\eta^d - \bar{\eta}_N^d\right] \right\} = 0,$$

and $\lim_{N\to\infty} \varDelta_N = 0$. Therefore, for the $\delta > 0$ specified in Lemma 5.5, there is an integer $N_0 > 0$ such that $0 < \varDelta_N < \delta/2$ for all $N > N_0$.

Now, assume that $N > N_0$. We have

$$\begin{aligned} &\left|\bar{g}_N^d(j) - g^d(j)\right| \\ =& \left|\bar{g}_N^d(j) - E\left[\bar{g}_N^d(j)\right] + E\left[\bar{g}_N^d(j)\right] - g^d(j)\right| \le \left|\bar{g}_N^d(j) - E\left[\bar{g}_N^d(j)\right]\right| + \varDelta_N. \end{aligned}$$

Therefore, if $\left|\bar{g}_N^d(j) - g^d(j)\right| > \delta$, $\delta > 0$, then $\left|\bar{g}_N^d(j) - E\left[\bar{g}_N^d(j)\right]\right| > \delta - \varDelta_N > \delta/2$. Thus,

$$\mathcal{P}\left\{\left|\bar{g}_N^d(j) - g^d(j)\right| > \delta\right\} \le \mathcal{P}\left\{\left|\bar{g}_N^d(j) - E\left[\bar{g}_N^d(j)\right]\right| > \delta/2\right\}.$$

Then, by Chebychev's inequality, we get

$$\mathcal{P}\left\{\left|\bar{g}_N^d(j) - E\left[\bar{g}_N^d(j)\right]\right| > \delta/2\right\} \le \frac{E\left\{\bar{g}_N^d(j) - E\left[\bar{g}_N^d(j)\right]\right\}^2}{(\delta/2)^2}.$$

Similar to (5.28), we have

$$E\left\{\bar{g}_N^d(j) - E\left[\bar{g}_N^d(j)\right]\right\}^2 = \frac{1}{N}\left\{E\left[V_k^d(i^*,j)\right]^2 - \left[E(\bar{g}_N^d(j))\right]^2\right\},$$

where $V_k^d(i^*,j) = \sum_{l=l_k^d(j)}^{l_k^d(i^*)-1}\left[f(X_l,d(X_l)) - \bar{\eta}_N^d\right]$. It is easy to verify that $E\left[V_k^d(i^*,j)\right]^2 < \infty$. From the above three equations, we can obtain

$$\mathcal{P}\left\{\left|\bar{g}_N^d(j) - g^d(j)\right| > \delta\right\} \le \frac{4c^d(j)}{N\delta^2},$$

for some $c^d(j) > 0$, and

$$\mathcal{P}\left\{||\bar{g}_N^d - g^d|| > \delta\right\} \le \frac{4c^d}{N\delta^2},$$

for some $c^d > 0$. Therefore,

$$\begin{aligned}
&\sum_{k=0}^{\infty} \max_{d \in \mathcal{D}} \mathcal{P}\left\{ ||\bar{g}^d_{N_k} - g^d|| > \delta \right\} \\
&= \sum_{k=0}^{N_0} \max_{d \in \mathcal{D}} \mathcal{P}\left\{ ||\bar{g}^d_{N_k} - g^d|| > \delta \right\} + \sum_{k=N_0+1}^{\infty} \max_{d \in \mathcal{D}} \mathcal{P}\left\{ ||\bar{g}^d_{N_k} - g^d|| > \delta \right\} \\
&\leq \sum_{k=0}^{N_0} \max_{d \in \mathcal{D}} \mathcal{P}\left\{ ||\bar{g}^d_{N_k} - g^d|| > \delta \right\} + \frac{4 \max_{d \in \mathcal{D}} c^d}{\delta^2} \sum_{k=N_0+1}^{\infty} \frac{1}{N_k},
\end{aligned}$$

in which the first term is finite. Thus, Lemma 5.5 holds if $\sum_{k=1}^{\infty} \frac{1}{N_k} < \infty$.

In the above analysis, we have assumed that every regenerative period visits state j. This may not be true for all $j \in \mathcal{S}$, especially for those states that are not visited often at steady state. To make sure that we can apply Lemmas 5.5 or 5.6, we may need to extend the length of the kth iteration N_k to a larger number N'_k such that, in the iteration, the number of regenerative periods that visit state j is larger than the N_k required by the algorithms, i.e., $N_k(j) = \sum_{l=1}^{N'_k} \chi_k(j) \geq N_k$, for all $j \in \mathcal{S}$. N'_k may be too large if some states are rarely visited. However, such states are usually not so "important", and, furthermore, the results in Lemmas 5.5 or 5.6 for true optimal policies may be a bit conservative. Further research in this direction is needed.

The results show that for the sample-path-based policy iteration to converge to the optimal policy with probability 1, the lengths of the sample paths in the iterations have to increase fast enough. In addition, in the algorithms with increasing lengths, it is difficult to determine the stopping criteria. At any iteration, it is always possible to have an estimate with a large error that leads to a wrong policy. We cannot be absolutely sure if the obtained policy is optimal even if the iteration stays at the same policy for a few (any finite number of) iterations. On the other hand, if the length is long enough, we may guarantee that the probability of the iteration stopping at a wrong policy is less that any given small positive number, by using the stopping criterion $d_{k+1} = d_k$.

The algorithm updates the policy (or the actions for all states) at the end of each iteration. Therefore, the required computation may be overwhelming at the end of every iteration. This may require a powerful machine for real-time applications, and the computation power may be wasted in the middle of every iteration. To overcome this disadvantage, we may determine the action for a state only when this state is visited during the iteration. More specifically, we may implement step 2 in Algorithms 5.1 and 5.2 at the end of each iteration and implement step 3 in these two algorithms for state i when this state is visited in the next iteration period. In this way, the computation is distributed to all the state transition instants. See [97] for more discussion.

Figure 5.2 illustrates the difference between the fixed-length and increasing-length algorithms. In the fixed-length algorithm, for any fixed length N_k, the

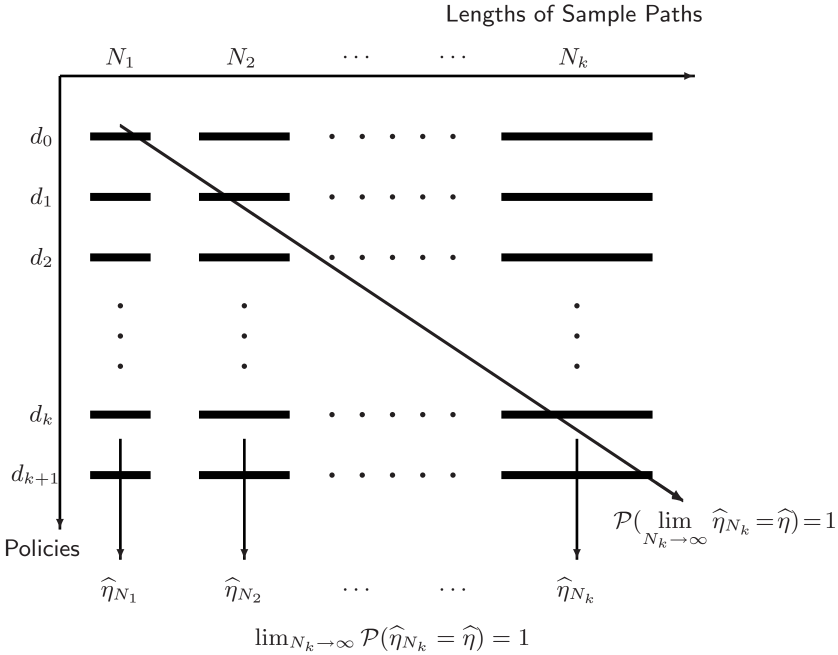

Fig. 5.2. Comparison of the Fixed- and Increasing-Length Policy Iteration Algorithms

algorithm stops at a near-optimal performance $\widehat{\eta}_{N_k}$, which converges to the optimal performance $\widehat{\eta}$ in probability as the length of the regenerative period N_k goes to infinity. In the increasing-length algorithm, the policy iteration goes in the diagonal direction in the figure and converges to the set of optimal policies with probability 1. However, it is difficulty to design stopping criteria for the increasing-length policy iteration algorithm.

Most results in this subsection appeared in [88].

5.3 "Fast" Algorithms*

In the algorithms presented in the above two sections, the potentials are estimated and policies are updated every iteration consisting of N regenerative periods, with N being a relatively large integer. In these algorithms, the potentials are estimated separately in each iteration. The estimates are relatively accurate with large N's. In this section, we explore the possibility of updating the potential estimates as well as the policies in every regenerative period, or after a few regenerative periods, in policy-iteration based performance optimization. The length of a regenerative period is not long enough for applying the algorithms in Sections 5.2.2 and 5.2.3, and therefore the information in the previous regenerative periods need to be used together with that in the current regenerative period to obtain an estimate, and stochastic approximation techniques may be employed.

5.3.1 The Algorithm That Stops in a Finite Number of Periods*

In the "fast" algorithm proposed in this subsection, the potential estimation is also based on (5.4). However, because the policies are updated whenever the system visits a reference state i^* in the algorithm, the different periods between the consecutive visits to i^* may be under different policies; therefore they are not identically distributed and hence are no longer "regenerative". We simply call them "periods". The kth period is denoted as Y_k, $k = 1, 2, \ldots$.

In the algorithm, to obtain an accurate estimate of the potentials, we start with running the system under an initial policy for N periods. Then, we update the policy in every period. The algorithm stops when the same policy is used for N consecutive periods.

Algorithm 5.3 Updating Policies in Every Period:

1. Choose an integer N; set $c := 0$ and $k := 0$; choose an initial policy d_0.
2. Observe the system under policy d_0 for N periods, and get an estimate $\bar{g}^{d_0}$ by applying (5.4) to these N periods.
3. Determine the next policy d_{k+1} by applying (5.14) with $\bar{g}^{d_k}$ as the estimated potentials.
4. If $d_{k+1} = d_k$, set $c := c + 1$; otherwise, set $c := 0$. If $c = N$, then exit; otherwise, go to the next step.
5. Change the policy to d_{k+1}, set $k := k + 1$, observe the system for one period with policy d_k, and update $\bar{g}^{d_k}$ by applying (5.4) to the latest N consecutive periods. Go to step 3.

In the Markov chain generated by the above algorithm, the initial policy d_0 is used in the first N periods, $Y_1, \ldots, Y_N$. $\bar{g}^{d_0}$ is estimated using these N periods and then d_1 is determined by $\bar{g}^{d_0}$. Policy d_1 is then used in the $(N+1)$th period, Y_{N+1}. In general, d_k and its corresponding transition matrix P^{d_k}, which is used in the $(N + k)$th period Y_{N+k}, are determined by $\bar{g}^{d_{k-1}}$, which is estimated using the kth period, Y_k, to the $(N + k - 1)$th period, Y_{N+k-1}, $k = 1, 2, \ldots$. This is illustrated in Figure 5.3.

Strictly speaking, in this algorithm, $\bar{g}^{d_k}$, $k \geq 1$, may not be the potential vector corresponding to policy d_k, which is only used in the last period. With this in mind, we will keep the same notation with superscript d_k to denote the estimated potential, since no confusion will be caused.

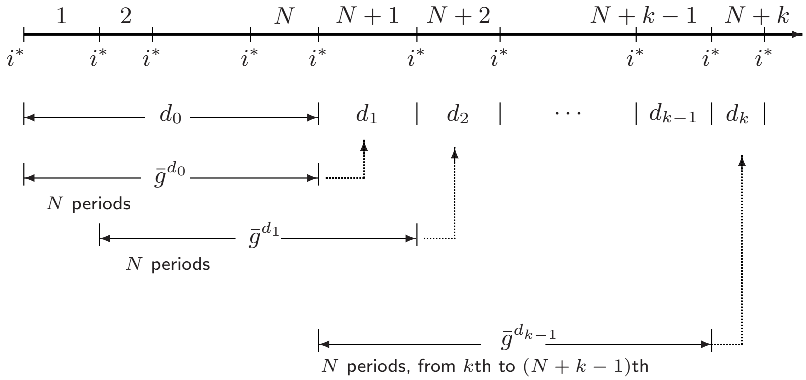

Fig. 5.3. The “Fast” Algorithm 5.3

The rationale behind the algorithm is as follows. If d_{k+1} is “close” to d_k, then the previous data under d_k can be used in obtaining $\bar{g}^{d_{k+1}}$. If d_{k+1} is not “close” to d_k, then the data collected in one period under d_{k+1} would not make a big impact on $\bar{g}^{d_{k+1}}$, which is estimated on N periods, i.e., we may have $\bar{g}^{d_{k+1}} \approx \bar{g}^{d_k}$. Therefore, most likely we would have $d_{k+2} = d_{k+1}$. Thus, the potential estimates would be more accurate for $\bar{g}^{d_{k+2}}$ in the next period, since two periods under the same policy $(d_{k+1} = d_{k+2})$ have been used. The policy gets updated when enough data under this policy $d_{k+1}(= d_{k+2})$ is collected. This also roughly explains that the algorithm might be “fast” because it wastes no periods to collect data that are more than needed to update the policies.

The Policy Reached When the Algorithm Stops

Lemma 5.7. *Suppose that Algorithm 5.3 stops at $\widehat{d}_N$; let η_N^* be the corresponding average reward. For any $\epsilon > 0$, there is an integer $N_\epsilon > 0$ such that if $N > N_\epsilon$, then $\mathcal{P}(\eta_N^* \neq \eta^*) < \epsilon$, where η^* is the true optimal average reward.*

Proof. When the algorithm stops at the last period, denoted as Y_{K+N}, the policies used in Y_K to Y_{K+N} are the same, i.e., $d_K = d_{K+1} = \cdots = d_{K+N} :=$ d. The potential estimated from the N periods Y_K to Y_{K+N-1}, $\bar{g}_N^d$, are based on the same policy d. By Algorithm 5.3, $\bar{g}_N^d$ leads to the same improved policy $d \in \psi(\bar{g}_N^d)$, which is used in Y_{K+N}. The theorem then follows directly from Theorem 5.1(b). □

The lemma claims that if the algorithm stops, then it stops at the true optimal policy with a large probability, if N is large enough. However, it does not indicate whether the algorithm will stop.

Does the Algorithm Stops?

Define the kth period as $Y_k := \left\{X_{l_k(i^*)+1}, \ldots, \ X_{l_{k+1}(i^*)}\right\}, k > 0$, and put N consecutive periods together as an augmented state $Z_k := (Y_k, Y_{k+1}, \ldots, Y_{N+k-1})$. Let $\mathscr{Z}$ be the space of all possible Z_k's. Then, we can write

$$d_k = \varphi(Z_k), \tag{5.29}$$

where φ is a mapping from $\mathscr{Z}$ to the policy space. The algorithm stops when the same policy is used for N consecutive basic periods.

From (5.29), the augmented chain $\boldsymbol{Z} = \{Z_1, Z_2, \ldots\}$ is a Markov chain defined on state space $\mathscr{Z}$. However, it may not be irreducible. In fact, if the algorithm converges to a policy (e.g., an optimal policy), then as time goes to infinity, $\boldsymbol{Z}$ tends to stay in the states generated by this policy (e.g., the optimal policy) and the other states may not be reached. Therefore, some conditions on $\boldsymbol{Z}$ may be required for the algorithm to stop. Let us study the issue formally. First, we have a lemma.

Lemma 5.8. *If $Z_{K+N} = Z_K$, then Algorithm 5.3 stops at the end of Z_{K+N}.*

Proof. $Z_{K+N} = Z_K$ means that $Y_{K+l} = Y_{K+N+l}$, $l = 0, \ldots, N-1$ (see Figure 5.4). Let the policies used in the $(K+N+l)$th period be d_{K+l}, $l = 0, \ldots, N-1$. Note that d_{K+1} depends on $Z_{K+1} = (Y_{K+1}, Y_{K+2}, \ldots, Y_{K+N-1}, Y_{K+N})$, which is the same as $Z_K = (Y_K, Y_{K+1}, \ldots, Y_{K+N-1})$ (because $Y_K = Y_{K+N}$), regardless of the order. Therefore, $d_K = \varphi(Z_K) = d_{K+1}$. In the same way, we can prove $d_{K+N} = d_{K+N-1} = \cdots = d_{K+N-2} = \cdots = d_K$. That is, the $N+1$ consecutive policies are the same. Thus, $c = N$ in the algorithm and hence it stops. □

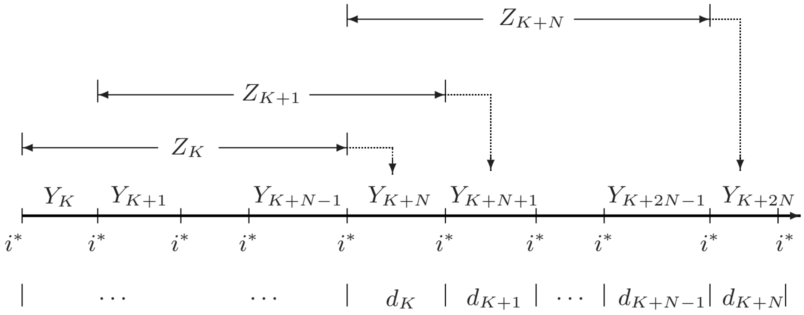

Fig. 5.4. The Periods in Lemma 5.8

Under many conditions, we may find $Z_{K+N} = Z_K$ on a sample path. These conditions require that the transition probability matrices used in different periods have some similarity. For instance, if the transition probability matrix used in the Kth period is completely different from that in the $(K+N)$th period, then a Y_K that is the same as Y_{K+N} may not exist.

To study the structure of a transition probability matrix P, we define a graph G consisting of S nodes. In the graph, two nodes i and j, $i, j \in \mathcal{S}$, are connected by an arrow from i to j if and only if $p(j|i) > 0$. A *loop* in G is a sequence of arrows starting from one node and ending at the same node. Let G^d be the graph corresponding to the transition probability matrix P^d, $d \in \mathcal{D}$.

The Common Loop Condition:
All the graphs G^d, $d \in \mathcal{D}$, have a *common loop.*

Any state lying on the common loop can be picked up as the reference state i^* in generating regenerative periods. Denote the common loop as $i^*, i_1, \ldots, i_m, i^*$. The common loop condition means that there is a period consisting of the sequence of states, $i^*, i_1, \ldots, i_m, i^*$, that can be generated with a positive probability by any policy in $\mathcal{D}$. We call this a *common period.*

Many policies satisfy this condition. For example, if for each $i \in \mathcal{S}$, we have a state $j \neq i$, such that $p^\alpha(j|i) > 0$ for all $\alpha \in \mathcal{A}(i)$, then the set of policies in $\mathcal{D}$ satisfies the common loop condition. Let us find one of the common loops under this condition. We start from any state i. Suppose that j_1 is the state such that $p^\alpha(j_1|i) > 0$ for all actions $\alpha \in \mathcal{A}(i)$, and j_2 is the state such that $p^\alpha(j_2|j_1) > 0$ for all actions $\alpha \in \mathcal{A}(j_1)$, and so on. In this way, we may obtain a sequence of states j_1, j_2, ... Since there are only a finite number of states, there must be two states denoted as j_{k_1} and j_{k_2}, with $k_1 \leq k_2$, such that $j_{k_1} = j_{k_2}$. We then have a common loop $j_{k_1} \to j_{k_1+1} \to \cdots \to j_{k_2} = j_{k_1}$.

Here is an example in which the two graphs do not have a common loop: G^{d_1}: $1 \to 2 \to 3 \to 4 \to 3$ and $4 \to 1$; G^{d_2}: $1 \to 2 \to 3 \to 1$ and $1 \to 4 \to 1$. In this example, the transition in state 3 is completely different for G^{d_1} and G^{d_2}: the system goes to 4 in G^{d_1} and to 1 in G^{d_2}. Therefore, the same path cannot be generated with P^{d_1} and P^{d_2} after the system reaches state 3.

Lemma 5.9. *Under the common loop condition, for any finite integer $N > 0$, Algorithm 5.3 stops with probability 1.*

Proof. Let us choose any state i^* in the common loop as the reference state. Because the number of policies is finite, the probability that any period is a common period is at least $p > 0$. Now, we divide the sample path into many intervals, each consisting of $2N$ periods. Consider a very special interval in which all the $2N$ periods are the same as the common period. The probability that an interval is such a special interval is larger than $p^{2N} > 0$. Therefore, the probability that in the first k intervals there is no such special interval is less than $(1-p^{2N})^k$, where $1 - p^{2N} < 1$. As $k \to \infty$, this probability goes to zero. That is, the probability that on a sample path the special interval never appears is zero. Because the special interval is a special case of $Z_{K+N} = Z_K$, the lemma follows directly from Lemma 5.8. □

Remark

It should be noted that although we only proved that under the common loop condition the algorithm stops with probability 1, it does not mean that the algorithm only stops when the special situation in the proof of Lemma 5.9 holds. In fact, in most cases, the algorithm stops before $\boldsymbol{Z}$ reaches such a special situation. To prove "stop with probability 1", we only need to find any special case that may stop the algorithm and prove such a case occurs with probability 1. It is true that, under this special case, the general property, e.g., Lemma 5.7, may not hold; however, it does hold in general because when the algorithm stops, the special case usually does not occur. See Problem 5.14 for more understanding.

5.3.2 With Stochastic Approximation*

In Algorithm 5.3, the potentials are estimated by using a fixed N number of periods (albeit possibly under different policies). This is similar to what is discussed in Section 5.2.2. In this subsection, we propose an algorithm (Algorithm 5.4) based on the stochastic approximation technique. In the algorithm, the potentials are estimated recursively at each period. This subsection parallels Section 6.3.1.

Algorithm 5.4 A Sample-Path-Based Algorithm with Stochastic Approximation:

1. Choose an initial policy d_0, and set $k = 0$ and $\bar{g}^{d_{-1}} = 0$. Choose an $\epsilon \in (0, 1/2)$ and a $C > 0$.
2. Observe the system under policy d_k for one period. For all $j \in \mathcal{S}$, calculate

$$V_k^{d_k}(i^*, j) = \begin{cases} \sum_{l=l_k^{d_k}(j)}^{l_{k+1}^{d_k}(i^*)-1} \left[f(X_l, d_k(X_l)) - \breve{\eta}_k^{d_k} \right], & \text{if } \chi_k(j) = 1, \\ \bar{g}^{d_{k-1}}(j), & \text{if } \chi_k(j) = 0, \end{cases}$$

where $\chi_k(j) = 1$ if the period contains j, and $\chi_k(j) = 0$ otherwise, and

$$\breve{\eta}_k^{d_k} = \frac{\sum_{l=l_k^{d_k}(i^*)}^{l_{k+1}^{d_k}(i^*)-1} f(X_l, d_k(X_l))}{l_{k+1}^{d_k}(i^*) - l_k^{d_k}(i^*)}.$$

Update the potential estimates as follows:

$$\bar{g}^{d_k}(j) = \bar{g}^{d_{k-1}}(j) + \frac{1}{k+1}\left[V_k^{d_k}(i^*, j) - \bar{g}^{d_{k-1}}(j)\right]. \tag{5.30}$$

3. For every $i \in \mathcal{S}$, set

$$\beta(i) \in \arg\left\{\max_{\alpha \in \mathcal{A}(i)}\left[\sum_{j=1}^{S} p^{\alpha}(j|i)\bar{g}^{d_k}(j) + f(i,\alpha)\right]\right\}.$$

   If there exists an i' such that

$$\begin{aligned}&\sum_{j=1}^{S} p^{\beta(i')}(j|i')\bar{g}^{d_k}(j) + f(i', \beta(i')) \\ \geq &\sum_{j=1}^{S} p^{d_k(i')}(j|i')\bar{g}^{d_k}(j) + f(i', d_k(i')) + \frac{C}{(k+1)^{1/2-\epsilon}},\end{aligned} \tag{5.31}$$

   then set $d_{k+1}(i) = \beta(i)$ for all i; otherwise, let $d_{k+1}(i) = d_k(i)$ for all i.
4. Set $k := k+1$ and go to step 2.

In step 2, $V_k^{d_k}(i^*, j)$ is the new information obtained in the kth period for potential $g^{d_k}(j)$. This information is used in (5.30) to update the estimate, in a way similar to stochastic approximation. If j does not appear in the period, no new information about $g^{d_k}(j)$ can be obtained in this period, and $\bar{g}^{d_{k-1}}(j)$ is used again.

Because the policies are updated often, the potential estimates may not be so accurate, especially at the beginning of the iteration procedure. This may cause the algorithm to be unstable. To avoid unnecessary oscillation between policies due to estimation errors, we add a threshold $\frac{C}{(k+1)^{1/2-\epsilon}}$ in (5.31) in step 3. The policy is not updated unless the difference in the comparison inequality exceeds a threshold. The value of the threshold gradually goes to zero as the policy approaches to the optimal one. The rate of the threshold approaching zero is controlled by ϵ. With this carefully designed updating scheme with a threshold, the algorithm converges to the optimal policy with probability 1 (a slightly different algorithm is proposed in [97], and its convergence is proved there).

It should be mentioned that there are many ways to propose such “fast” algorithms. The two proposed in Sections 5.3.1 and 5.3.2 just serve as examples. For such algorithms, the convergence speed is not known even if the convergence is proved. That is, we are not sure if they are really “faster” than the other sample-path-based algorithms, and in what sense they may be faster.

PROBLEMS

5.1. Repeat Example 5.1 by using the continuous-time Markov model.

5.2. A machine produces M different products, denoted as $1, 2, \ldots, M$. To process product i, the machine has to perform N_i different operations, denoted as $(i, 1), (i, 2), \ldots, (i, N_i)$. We use a discrete time model. At each time l, $l = 0, 1, \ldots$, the machine can only process one product and perform one operation. If at time instant l the machine is producing product i and is at operation (i, j), $j \neq N_i$, then at time instant $l + 1$ the machine will take operation (i, j') with probability $p_i(j'|j)$, $i = 1, 2, \ldots, M$, $j = 1, \ldots, N_i - 1$, and $j' = 1, \ldots, N_i$. If the machine is at operation (i, N_i), then it will pick up a new product i' and start to process it at operation $(i', 1)$ at the next time instant with probability $p^{\alpha}(i'|i)$, $i, i' = 1, 2, \ldots, M$, where $\alpha \in \mathcal{A}(i)$ represents an action. The operation $(i, 1)$ is called an *entrance operation* and (i, N_i) is called an *exit operation*. The system can be modelled as a Markov chain with state space $\mathcal{S} := \{(i, j) : i = 1, 2, \ldots, M, j = 1, \ldots, N_i\}$. Let f be the properly defined reward function. Derive the policy iteration condition (similar to (5.1) in Example 5.1) for this problem and show that with the sample-path-based approach we do not need to estimate the potentials for all the states.

5.3. In Problem 4.1, prove that if we use the sample-path-based approach, then we do not need to know the value of r.

5.4. As discussed in Section 5.1, to save memory and computation at each iteration, we may partition the state space $\mathcal{S} = \{1, 2, \ldots, S\}$ into N subsets and at each iteration we may only update the actions for the states in one of the subsets. In the extreme case, at each iteration, we may update the action for only one state. That is, at the first iteration, we update $d(1)$; at the second iteration, we update $d(2), \ldots$, and at the Sth iteration, we update $d(S)$. Then, at the $(S + 1)$th iteration, we update $d(1)$ again, and so on in a round robin manner. In such an iteration procedure, we cannot stop if there is no improvement in the performance at some iteration. We let the iteration algorithm stop after the performance does not improve in S consecutive iterations.

a. Formally state this policy iteration algorithm.
b. Prove that the algorithm stops after a finite number of iterations.
c. Prove that the algorithm stops at a gain-optimal policy.
d. Extend this algorithm to the general case where $\mathcal{S}$ is partitioned into N subsets.

5.5. To illustrate the idea behind Lemma 5.2, we consider the following simple problem. There are N different balls with identical appearance but different weights, denoted as $m_1, m_2, \ldots, m_N$, respectively, $m_i \neq m_j$, $i \neq j$. These weights are known to us. You have a scale in your hand that is inaccurate with a maximal absolute error of $r > 0$. Under what condition will you accurately identify these balls using this scale?

5.6. Suppose that when the sample-path-based policy iteration algorithm 5.2 stops, the estimation error of the potentials satisfies $|r| = |\bar{g} - g| < \delta/2$, where $\delta > 0$ is any positive number. Let $\bar{\eta}$ be the optimal average reward thus obtained. Prove

$$|\bar{\eta} - \eta^*| < \delta,$$

where η^* is the true optimal average reward.

5.7. If we use

$$\frac{\sum_{n=0}^{N-L+1} \left\{ I_i(X_n) \left[\sum_{l=0}^{L-1} f(X_{n+l}) - \eta \right] \right\}}{\sum_{n=0}^{N-L+1} I_i(X_n)}$$

to estimate the potentials, then the estimates are biased.

a. Convince yourself that the results in Section 5.2 still hold, and
b. Revise the proofs in Section 5.2 for the sample-path-based policy iteration with the above potential estimates.

5.8. With the sample-path-based policy iteration Algorithm 5.1, suppose that the Markov chain is ergodic with a finite state space under all policies, and the number of policies is finite. Let $|r| = |\bar{g}^d - g^d| < (\kappa/2)e$, where g^d and $\bar{g}^d$ are the potential of policy d and its estimate. Following the same argument as that in Lemma 5.3, prove that

$$\phi(\bar{g}^d) \subseteq \phi(g^d).$$

5.9. In Problem 5.8, we proved that $\phi(\bar{g}^d) \subseteq \phi(g^d)$.

a. On the surface, it looks like the same method as that in Lemma 5.3 can be used to prove $\phi(g^d) \subseteq \phi(\bar{g}^d)$. Give it a try.
b. If you cannot prove the result in a), explain why; if you feel that you did prove it, determine what is wrong in your proof.
c. Suppose that $h, h' \in \phi(g^d)$, and thus $f^h + P^h g^d = f^{h'} + P^{h'} g^d$. Because of the error in $\bar{g}^d$, we may have $f^h + P^h \bar{g}^d \neq f^{h'} + P^{h'} \bar{g}^d$. Therefore, one of them cannot be in $\phi(\bar{g}^d)$. Give an example to show that no matter how small the error $r = g^d - \bar{g}^d$ is, this fact is true.

5.10. Are the following statements true? Please explain the reasons for your answers:

a. Suppose that we use $d_{k+1} \in \phi(\bar{g}_N^{d_k})$ to replace (5.14) in step 3 of Algorithm 5.1 (i.e., set $\nu = 0$ in (5.12)). Then, the algorithm may not stop even if $\phi(\bar{g}_N^{d_k}) \subseteq \phi(g^{d_k})$ for $K' > K$ consecutive iterations $k = n, n+1, \ldots, n+K-1$, where K is the number of policies in $\mathcal{D}$.
b. Algorithm 5.2 may not always stay in $\mathcal{D}_0$ even after $\phi(\bar{g}_{N_k}^{d_k}) = \phi(g^{d_k})$ for K consecutive iterations, where K is any large integer.
c. Statement b) above is true even if we add the following sentence to step 3 of Algorithm 5.2: "If at a state i, action $d_k(i)$ attains the maximum, then set $d_{k+1}(i) = d_k(i)$."

5.11. Can you propose any stopping criteria for the sample-path-based algorithms to stop at an optimal policy in a finite number of iterations with probability 1?

5.12. In Lemma 5.4, $\sum_{k=0}^{\infty}(1 - y_k) < \infty$ implies $\lim_{k\to\infty} y_k = 1$, which, however, is not enough for $\lim_{n\to\infty} \prod_{k\geq n} y_k = 1$. For the latter to hold, y_k has to approach 1 fast enough.

a. For $y_k = 1 - \frac{1}{k}$, $k = 1, 2, \ldots$, we have $\lim_{k\to\infty} y_k = 1$. What is $\lim_{n\to\infty} \prod_{k\geq n} y_k$?
b. Verify the lemma for $y_k = 1 - \frac{1}{k^2}$, $k = 1, 2, \ldots$. What is $\lim_{n\to\infty} \prod_{k\geq n} y_k$?
c. For a sequence y_k, $0 \leq y_k \leq 1$, $k = 1, 2, \ldots$, if $\sum_{k=0}^{\infty}(1 - y_k) < \infty$ we have $\sum_{k=0}^{\infty}(1 - y_k^c) < \infty$ for any $c < 1$ and we can apply this lemma. How about $c > 1$?

5.13.* Write a simulation program for the "fast" Algorithm 5.3. Run it for a simple example with, say, $S = 3$, and each $\mathcal{A}(i)$, $i \in \mathcal{S}$, containing three to five actions. Record the sequence of d_k, $k = 0, 1, 2, \ldots$, and observe its behavior, e.g., how it changes from one policy to another one. Run it a few times with different N's.

5.14.* This problem is designed to help you to understand the remark on the proofs in Section 5.3.1. Consider an ergodic Markov chain $\boldsymbol{X} = \{X_0, X_1, \ldots, X_l, \ldots\}$ with state space $\mathcal{S}$ and reward function $f(i)$, $i \in \mathcal{S}$. Let $i^* \in \mathcal{S}$ be a special state. We repeat the following game: Every time we run the Markov chain, we let it stop when $X_l = X_{l+2} = i^*$; and when it stops, we receive a total reward of $f(X_{l+1})$.

a. We may prove that the Markov chain stops with probability 1 under the special condition $p(i^*|i^*) \neq 0$.
b. Suppose that the Markov chain stops with probability 1. Then, the expected total reward we receive is $\bar{r} = \sum_{k\in\mathcal{S}} p(k|i^*) f(k)$.

Obviously, $p(i^*|i^*) \neq 0$ is not a necessary condition, and this special condition does not change the expected total reward $\bar{r}$ in part b).

5.15.* If we implement Algorithm 5.3 for a few reference states $i^*, j^*, k^*, \ldots$ in parallel on the same sample path, then we can update the policy whenever the system reaches one of these states. In the extreme case, if we implement the algorithm using every state as the reference state separately on the same sample path, we may update the policy at every state transition on the sample path.

We need to study the convergence of such algorithms. Consider, for example, the case where we have two reference states i^* and j^*. Whenever we meet states i^* or j^*, we will update the policy. Therefore, if in a period starting from one i^* to the next i^*, the sample path visits state j^*, then the policy used in this period before visiting j^* is different from that used after the visit. Does this cause a major problem in the convergence of the algorithm? How about the algorithm in which we use all states as reference states?