Simulation-Based Methods for Markov Decision Processes

Peter Marbach
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

Abstract

Markov decision processes have been a popular paradigm for sequential decision making under uncertainty. Dynamic programming provides a framework for studying such problems, as well as for devising algorithms to compute an optimal control policy. Dynamic programming methods rely on a suitably defined value function that has to be computed for every state in the state space. However, many interesting problems involve very large state spaces ("curse of dimensionality"), which prohibits the application of dynamic programming . In addition, dynamic programming assumes the availability of an exact model, in the form of transition probabilities ("curse of modeling"). In many practical situations, such a model is not available and one must resort to simulation or experimentation with an actual system. For all of these reasons, dynamic programming in its pure form may be inapplicable.

In this thesis we study an approach for overcoming these difficulties where we use

(a) compact (parametric) representations of the control policy, thus avoiding the curse of dimensionality, and

(b) simulation to estimate quantities of interest, thus avoiding model-based computations (curse of modeling).

It is not limited to Markov decision processes, but applies to general Markov reward processes for which the transition probabilities and the one-stage rewards depend on a parameter vector θ.

We propose gradient-type algorithms for updating θ based on the simulation of a single sample path, so as to improve a given performance measure. As possible performance measures we consider the weighted reward-to-go and the average reward. The corresponding algorithms

(a) can be implemented online and update the parameter vector either at visits to a certain state, or at every time step.

(b) have the property that the gradient (with respect to θ) of the performance measure converges to 0 with probability 1. This is the strongest possible result for gradient-related stochastic approximation algorithms.

We illustrate the methodology by considering the call admission control problem where a telecommunications provider sells bandwith of a single communication link to customers so as to optimize the revenue. We use the proposed algorithms to optimize the parameters a heuristic threshold policy for this problem.

Contents

Chapter 1

Introduction

1.1 Problem Statement

In this thesis we consider the problem of controlling finite-state systems. As the system proceeds (in stages) from state to state, a decision maker chooses at each stage a control action which influences the value of a one-stage reward, as well as the likelihood of a transition to any particular next state. The goal of the decision maker is to choose control actions so as to maximize the long term reward. A characteristic of this type of problems is that the choice of a control action influences both the immediate reward and the dynamics of the system, and therefore the potential future rewards. Thus, an optimal choice of a control action aims at maximizing the sum of the one-stage reward and the expected future rewards.

There is a wide class of problems in a variety of areas such as economics, manufacturing systems, telecommunications, operations research, etc., which can be formulated in this way. For example, consider the situation where a commercial provider of a telecommunications network serves customers so as to maximize the long term revenue. Customers are divided into different classes, called service types. Each service type is characterized by its bandwidth demand, its average call holding time, and by an immediate reward (price) the provider obtains whenever a customer of that service type is accepted to the network. The reward associated with a service type can be interpreted as the price the customer pays to the network provider for using the corresponding bandwidth for the duration of a call. The control actions available to the network provider (to maximize the long term revenue) are to accept or reject new connection requests (Call Admission Control), and to route the accepted requests appropriately through the network (Routing). Typically, the call admission control decisions of the provider will depend on the current state (configuration) of the network, as well as on the reward (price) associated with the call. If the network is (almost) empty, the provider will (probably) decide to accept all new connection requests and obtain the corresponding reward, while still having enough bandwidth available for serving potential future requests. However, if the network is almost full, the provider may decide to reject a request of a customer with a small reward, in order to reserve capacity for customers of more "valuable" service types, i.e., customers that pay a higher price for using the network. Also, the routing decisions will generally depend on the current state of the network. That is, the network provider will choose routes which avoid busy links. In order to maximize the long term revenue, at each state of the network an optimal call admission control and routing decision has to be chosen.

Markov decision processes provide a mathematical model for such problems which has the following ingredients.

1. A state space S which contains all possible states of the system.

2. A set U of possible control actions that can be applied at each state.

3. A law $P_{ij}(u)$ which specifies the probability of a transition from the state $i \in S$ to a next state $j \in S$ under control action $u \in U$.

4. A law $g_i(u)$ which specifies the one-stage reward the decision maker obtains at the state $i \in S$ under control action $u \in U$.

In this thesis, we consider Markov decision processes for which the state space S, and the set of possible control actions U, are finite. A control policy is given by a mapping

$$\mu : S \mapsto U,$$

which lists for each possible state of the system a corresponding control action. The goal of the decision maker is to find a control policy that maximizes the expected long term reward.

Dynamic programming provides a framework for studying Markov decision processes, as well as algorithms for computing optimal control policies. Methods of dynamic programming rely on a suitably defined value function that has to be computed for every possible state of the system. However, many interesting problems (e.g., telecommunication networks, manufacturing systems, business organizations, etc.) involve a large state space ("curse of dimensionality"), which makes it impossible to compute, or even store, this value function. In addition, dynamic programming assumes that the exact mathematical model of the Markov decision process is available. However, in many situations, the decision maker does not have not access to such a model ("curse of modeling"). As a result, algorithms of dynamic programming are often not a viable method for a given Markov decision process.

For the example of the call admission control and routing problem given above, each possible configuration (level of utilization of each route) corresponds to a different state. As the number of configurations increases exponentially in the number of links and customer classes, it is often impossible to even list all states, or to compute and store an optimal call admission control and routing policy. In practice, heuristic policies, such as "Open-Shortest-Path-First" [MMST98], are employed for call admission control and routing.

In this thesis, we investigate an alternative approach to using heuristics. In particular, we are interested in a methodology that allows us to compute a "good" control policy in a "systematic" and "efficient" way, without requiring the computational burden of dynamic programming.

1.2 Approach

We consider an approach to overcome the difficulties associated with dynamic programming that is based on the following two ideas:

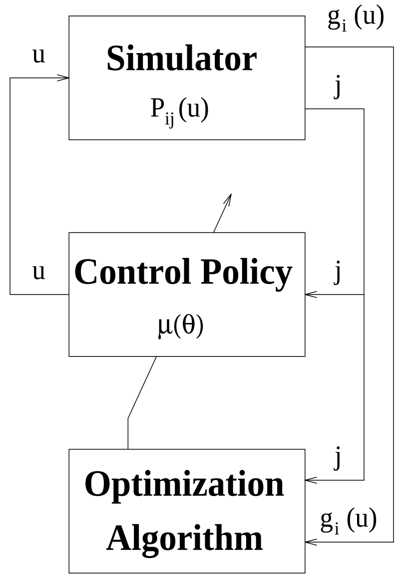

Figure 1.1: Block-Diagram of Simulation-Based Optimization of Parameterized Control Policies.

1. The use of compact (parametric) representations of the control policy, thus avoiding the curse of dimensionality.

2. The use of simulation to estimate quantities of interest, thus avoiding model-based computations and the curse of modeling.

Instead of specifying for each state in the state space a corresponding control action, we consider control policies which are described in terms of a parameter vector $\theta = (\theta_1, ..., \theta_K) \in \Re^K$, whose dimension K is tractably small. The goal of the decision maker is then to find a parameter vector θ that maximizes a given performance measure $\psi(\theta)$. As methods for optimizing the parameter vector θ, we consider gradient-type algorithms where the decision maker employs simulation to estimate the gradient of the performance measure $\psi(\theta)$ with respect to θ, and improves the policy by updating the parameter vector along an approximate gradient direction. We require that the corresponding algorithms can be implemented online, and that the parameter vector θ gets updated at every time step. This approach is illustrated by the block diagram given in Fig 1.1. Let us comment on its three components.

To produce an estimate of the gradient of the performance measure, the decision maker has to be able to access a simulator which accurately represents the Markov decision process. The simulator takes as an input a control action $u \in U$ and produces as output a next state $j \in S$ according to the transition probabilities $P_{ij}(u)$, where i is the current state of the system. In the same manner, the one-state reward $g_i(u)$ is obtained. Such a simulator may be given in the form of a computer program. In that case, the optimization of the parameter vector is made offline based on data obtained from a computer simulation. Also, the decision maker may use the actual system as a "simulator". In this case, the parameter vector gets updated online based on real-time observations of the actual system.

The simulator is controlled by a policy which is defined in terms of a parameter vector θ. The policy takes as an input a state of the system, and produces as an output a control action $u \in U$. The parametric representation $\mu(\theta)$ of the control policy is chosen by the decision maker. Here we stress that, in practice, this choice is important, as it influences both the performance (rate of convergence) of the optimization method and the quality of the resulting control policy. Typically, it is difficult to satisfy both of these objectives simultaneously, resulting in a trade-off between choosing a compact representation which is

- "simple", i.e., involves a parameter vector with few components, resulting in fast convergence of the optimization algorithm.
- "rich", i.e., involves a parameter vector with many components, and is able to approximate an optimal control policy closely.

The choice of a suitable compact representation may require a combination of intuition and analysis of the problem, as well as trial and error. In this thesis, we do not address the issue of how to choose a parametric representation of the control policy, but assume that a parameterized family of policies $\mu(\theta)$ is given and satisfies certain regularity conditions. In particular, in order to employ gradient-based algorithms, the performance measure $\psi(\theta)$ has to depend smoothly on θ, i.e., $\psi(\theta)$ has to be differentiable with respect to θ. To satisfy this assumption, we consider randomized control policies which specify for each state in the state space, the probability of choosing a specific control action. We associate with each $\theta \in \Re^K$ a mapping $\mu(\theta) : S \mapsto [0,1]^L$, where L is the number of possible control actions in the set U. The component $\mu_u(i,\theta)$, $u = 1, ..., L$, specifies the probability of choosing control action u at state i. Naturally, we require that, for every state $i \in S$, the sum $\sum_{u=1}^{L} \mu_u(i,\theta)$ is equal to 1. We will show in Chapter 2, that there exists a rich class of functions that can be used to represent a randomized control policy $\mu(\theta)$.

Based on data obtained from the simulator, the optimization algorithm produces an estimate $F(\theta)$ of the gradient $\nabla\psi(\theta)$ of the performance measure $\psi(\theta)$ with respect to θ, and iteratively updates the parameter vector of the policy $\mu(\theta)$ along an approximate gradient direction. The main thrust of this thesis is the analysis of simulation-based algorithms to tune θ. This analysis is based on the following three steps.

1. Derive an expression for the gradient $\nabla\psi(\theta)$ of the performance measure with respect to θ.
2. Using this expression, derive an estimate $F(\theta)$ of the gradient $\nabla\psi(\theta)$ which can be obtained through simulation.
3. Analysis the algorithm which iteratively updates θ along the approximate gradient direction $F(\theta)$, i.e.,
$$\theta_{k+1} = \theta_k + \gamma_k F(\theta_k),$$
where γ_k is a small step size parameter.

We can rewrite the iteration above as

$$\begin{aligned}\theta_{k+1} &= \theta_k + \gamma_k F(\theta_k) \\ &= \theta_k + \gamma_k \nabla\psi(\theta_k) + \varepsilon_k,\end{aligned}$$

where

$$\varepsilon_k = \gamma_k \left(F(\theta_k) - \nabla\psi(\theta_k) \right).$$

In that form, the optimization algorithm can be interpreted as a gradient algorithm with a stochastic noise term ε_k. We wish to show that (with probability 1) the algorithm converges in the following sense: the sequence $\psi(\theta_k)$ converges and $\lim_{k\to\infty} \nabla\psi(\theta_k) = 0$. This is the strongest possible result for gradient-related stochastic approximation algorithms. Note that this result only assures that the performance measures $\psi(\theta_k)$, but not the parameter vectors θ_k and the control policies $\mu(\theta_k)$, converge. In addition, it does not say anything about to what the performance measure $\psi(\theta_k)$ converges. In a worst case, even if the parameterization of the control policy is able to represent an optimal control policy, the algorithm may produce control policies with a very poor performance.

Let us close this section by summarizing the fundamental limitations of the aforementioned approach.

1. By considering compact (parameterized) representations of the control policy, we limit the search for a control policy to a set which is typically smaller than the set of all possible control policies.

2. By considering gradient-based algorithms as a search method for a control policy, we will typically find a solution which is locally, rather than globally, optimal.

Therefore, while methods of dynamic programming compute a control policy which is (globally) optimal in the set of all control policies, the proposed approach of using parameterized control policies and gradient-based optimization methods for solving Markov decision processes typically produces a control policy which is locally optimal within a given family of parameterized control policies.

1.3 Outline and Overview

In Chapter 2, we give a summary of the aspects of Markov reward processes, Markov decision processes, and dynamic programming that we will use. We use this chapter mainly to introduce definitions and notations, and to lay out the motivation for the problem formulation.

In Chapter 3, we state the mathematical framework that we consider. In particular, we introduce Markov reward processes that are described in terms of a parameter vector $\theta \in \Re^K$, and define the weighted reward-to-go $\chi(\theta)$, and the average reward $\lambda(\theta)$, as performance measures for optimizing θ. The weighted reward-to-go concerns problems that involve a finite (possibly random) termination time. Furthermore, the weighting is taken with respect to the random choice of an initial state.

In Chapter 4, we consider the weighted reward-to-go $\chi(\theta)$ as a performance measure associated with a parameterized Markov reward process. We derive an expression for the gradient of the weighted reward-to-go with respect to θ, and propose a gradient algorithm for tuning θ so as to improve $\chi(\theta)$. Using standard results of nonlinear programming, we show convergence for this algorithm. Unfortunately, for problems involving a large state space it is not feasible to compute $\nabla\chi(\theta)$ exactly. As an alternative approach, we consider the situation where the decision maker employs simulation to obtain an unbiased estimate of the derivative

$\nabla\chi(\theta)$, and updates the parameter vector θ along an approximate gradient direction at visits to a certain recurrent state i^*. Convergence, with probability 1, of this algorithm is established by using standard techniques of stochastic approximation. Furthermore, we derive an algorithm which updates the parameter vector θ at each time step. Extending techniques of infinitesimal perturbation analysis, we prove convergence for this modified algorithm.

In Chapter 5, we consider the average reward $\lambda(\theta)$ as a performance measure associated with a parameterized Markov reward process. The program we follow is similar to the one of Chapter 4. Again, we derive an expression of the gradient of the average reward with respect to θ, and propose a gradient algorithm for updating θ. Furthermore, we propose a method which employs simulation to obtain an estimate of $\nabla\lambda(\theta)$, and updates θ along an approximate gradient direction, either at visits to a certain state i^*, or at every time step. The analysis in this chapter is complicated by the fact that a biased estimate is used to update the parameter vector θ. Therefore, convergence cannot be established using standard approaches (e.g., martingale arguments), and a more elaborate proof is necessary.

In Chapter 6 we apply the theory developed for parameterized Markov reward processes to Markov decision processes. In particular, we consider Markov decision processes that involve a randomized control policy which is described in terms of a parameter vector. We show that this is a special instance of a parameterized Markov reward process. Therefore, all results obtained in Chapters 4 and 5 apply in this context. Furthermore, we show how some of the assumptions that we used in the previous chapters can be relaxed.

In Chapter 7, we illustrate the theory developed in the previous chapters by carrying out a case study. We consider a special case of the call admission control problem, which arises when a telecommunication provider sells bandwidth of a communication link to customers so as to maximize the long term revenue. We show how the proposed methodology can be applied, and provide numerical results illustrating the corresponding algorithms.

1.4 Contributions

Chapter 4

- We derive an expression for the gradient $\nabla\chi(\theta)$ of the weighted reward-to-go with respect to θ, which can be employed to produce an unbiased estimator.

- Using this estimator, we propose a simulation-based algorithm for optimizing the weighted reward-to-go $\chi(\theta)$ which can be implemented online, and updates the parameter vector either at visits to a certain state i^*, or at each time step. We prove convergence in the following sense: with probability 1 the sequence $\chi(\theta_k)$ converges, and $\lim_{k\to\infty} \|\nabla\chi(\theta_k)\| = 0$, where θ_k is the parameter vector at the kth iteration step.

- We propose two approaches that reduce the variance in the estimate of the gradient $\nabla\chi(\theta)$, but introduce a bias. For both approaches, we characterize the resulting bias, and the behavior of the corresponding algorithm, i.e., we show that for each approach the bias is bounded by a constant D, and $\liminf_{k\to\infty} \|\nabla\chi(\theta_k)\| \leq D$.

Chapter 5

- We propose a simulation-based algorithm for optimizing the average reward $\lambda(\theta)$ in parameterized Markov reward processes. The algorithm can be implemented online, and allows us to update the parameter vector θ at visits to a recurrent state i^*, or at each time step. We prove convergence in the following sense: with probability 1 the sequence $\lambda(\theta_k)$ converges, and $\lim_{k\to\infty} \|\nabla\lambda(\theta_k)\| = 0$, where θ_k is the parameter vector at the kth iteration step.

- We propose two approaches that reduce the variance in the estimate of the gradient $\nabla\lambda(\theta)$, but introduce a bias. For both cases, we characterize the resulting biases, and the behavior of the corresponding algorithm, i.e., we show that for each approach the bias is bounded by a constant D, and $\liminf_{k\to\infty} \|\nabla\lambda(\theta_k)\| \leq B(D)$, where $B_1(D)$ is a bound that grows linearly in D.

Appendix C.1

- The technique used to prove convergence for the algorithm that optimizes the average reward $\lambda(\theta)$ is of independent interest and is significantly different from published convergence proofs for related methods. The algorithm belongs to a class of stochastic approximation algorithms considered in [Bor97]. The author of [Bor97] employed a time-scale separation assumption. Such a time-scale separation may lead to slow convergence and our approach overcomes the need for such an assumption.

1.5 Related Work

In this section, we provide a literature review and compare our approach with existing work. We will discuss some of the cited papers in more detail in the following chapters.

Markov decision processes and the associated dynamic programming methodology provide a general framework for posing and analyzing problems of sequential decision making under uncertainty. For a textbook exposition of this theory see [Ber95a, Put94]. However, as mentioned earlier, the "curse of dimensionality" and the "curse of modeling" prevent dynamic programming from being a viable method for solving many problems of interest. The efforts to overcome these difficulties involve two main ideas:

1. The use of parametric representations to overcome the curse of dimensionality.

2. The use of simulation to estimate quantities of interest, thus avoiding model-based computations.

Parametric representations, and the associated algorithms, can be broadly classified into three main categories:

(a) One can use a parametric representation of the value function of dynamic programming. For example, instead of associating a value V_i with every state i, one uses a parametric form $\tilde{V}_i(r)$, where r is a vector of tunable parameters (weights), and $\tilde{V}$ is a so-called approximation architecture. For example, $\tilde{V}_i(r)$ could be the output of a multilayer perceptron with weights r, when the input is i. Other representations are possible, e.g., involving polynomials, linear combinations of feature vectors, state aggregation, etc. When the main ideas from dynamic programming are combined with such parametric representations, one obtains methods known as "reinforcement learning" or "neuro-dynamic programming"; see [SB98, BT96] for textbook expositions, as well as the references therein. Some of the main methods of this type are Sutton's temporal difference (TD) methods [Sut88], Watkins' Q-learning algorithm [Wat89], and approximate policy iteration [BT96]. The distinguishing characteristic of such methods is that policy optimization is carried out in an indirect fashion: one tries to obtain a good approximation of the optimal value function of dynamic programming, and then uses it to construct policies that are close to optimal. The understanding of such methods is still somewhat incomplete: convergence results or performance guarantees are available only for a few special cases such as state-space aggregation [TR96], optimal stopping problems [TR97], and an idealized form of Policy Iteration [BT96]. However, there have been some notable practical successes (see [SB98, BT96] for an overview), including the world-class backgammon player by Tesauro [Tes92].

(b) In an alternative approach, the tuning of a parameterized value function is bypassed. Instead, one considers a class of policies described in terms of a parameter vector θ. Simulation is employed to estimate the gradient of the performance metric with respect to θ, and the policy is improved by updating θ in a gradient direction. Methods of this type have been extensively explored in the Infinitesimal Perturbation Analysis literature [CR94, HC91]. Many of these methods are focused on special problem types and do not readily extend to general Markov decision processes.

(c) A third approach, which is a combination of the above two, involves a so-called "Actor-Critic" architecture, that includes parameterizations of the policy (actor) and of the value function (critic) [BSA83]. At present, little is known about the theoretical properties of such methods (see, however, [Rao96]).

This thesis concentrates on methods based on policy parameterization and (approximate) gradient improvement, in the spirit of item (b) above. This approach is not restricted to Markov decision processes, but applies to general Markov reward processes that depend on a parameter vector θ. As performance measures for tuning the parameter vector, we consider the weighted reward-to-go and the average reward. While, to our knowledge, the weighted reward criterion is an original contribution of this thesis, using the average reward as a metric for optimizing a parameter vector has been studied in the literature. For optimizing the average reward, we consider an approach that relies on a gradient formula that has been presented in different forms in [CC97, CW98, FH94, FH97, JSJ95, Gly87]. We exploit a variant of this formula and develop a method that updates the parameter vector θ, at every renewal time, in an approximate gradient direction. Furthermore, we show how to construct a method that updates the parameter vector at each time step. In this respect, our work is closer to the methods described in [CR94] (that reference assumes, however, the availability of an estimator, with certain guaranteed properties that are absent in our context) and [JSJ95] (which, however, does not contain convergence results). The method that we propose only keeps in memory and updates $2K+1$ numbers, where K is the dimension of θ. Other than θ

itself, this includes a vector similar to the "eligibility trace" in Sutton's temporal difference methods, and (similar to [JSJ95]) an estimate $\tilde{\lambda}$ of the average reward under the current value of the parameter vector. If that estimate were accurate, our method would be a standard stochastic gradient algorithm [BT97]. However, as the policy keeps changing, $\tilde{\lambda}$ is generally a biased estimate of the true average reward, and the mathematical structure of our method is more complex than that of stochastic gradient algorithms. For reasons that will become clearer later, convergence cannot be established using standard approaches (e.g., martingale arguments or the ODE approach), and therefore a more elaborate proof is necessary.

Generally, the ideas presented in this thesis draw from a variety of previous work in areas such as infinitesimal perturbation analysis, reinforcement learning/neuro-dynamic programming, and stochastic approximation. The main contribution of the thesis is to combine and extend this body of previous work in order to obtain a methodology which allows us to compute a "good" control policy in a "systematic" and "efficient" way, without requiring the computational burden of dynamic programming.

We view the resulting methodology not as a general purpose solution, but rather as an additional tool for addressing large scale Markov decision processes. As mentioned earlier, a successful application of the proposed algorithms will heavily rely on the choice of a suitable representation $\mu(\theta)$ of the randomized control policy. We expect that our methodology will be useful in situations where a decision maker wishes to optimized a well-designed heuristic policy which involves tunable parameters. We provide in Chapter 7 such an application, where we tune the parameters of a call admission control policy for a single communication link. At the end of Chapter 7, we will discuss some related work for this problem which helps to put our method and its performance in perspective.

Chapter 2

Markov Decision Processes and Dynamic Programming

In this chapter, we give a brief summary of Markov reward processes, Markov decision processes and dynamic programming. We mainly use this chapter to introduce definitions and notations. The derivation of the results that we state here can be found in [Ber95a, Gal95, Put94].

2.1 Markov Reward Processes

In this section, we introduce Markov reward processes by considering the situation where each state in a Markov chain is associated with a reward.

We consider a discrete-time, finite-state Markov chain which is defined on the state space $S = \{1, \ldots, N\}$. We denote the one-step transition probabilities by P_{ij}, $i, j \in S$, and the $n-$step transition probabilities by P_{ij}^n, i.e.,

$$P_{ij} = P(i_1 = j \mid i_0 = i), \quad \text{and } P_{ij}^n = P(i_n = j \mid i_0 = i), \quad n = 1, 2, \ldots ,$$

where i_n stands for the state of the chain at time n. We make the following assumption.

Assumption 1 *There exists a state $i^* \in S$ such that, for every state $i \in S$, we have*

$$\sum_{n=1}^{N} P_{ii^*}^n > 0,$$

where N is the number of states in the state space S.

Assumption 1 is equivalent to assuming that the underlying Markov chain has a single recurrent class and the state i^* is a recurrent state.

Under Assumption 1, the balance equations

$$\sum_{i=1}^{N} P_{ij}\pi_i = \pi_j, \qquad j = 1, \ldots, N-1, \tag{2.1}$$

$$\sum_{i=1}^{N} \pi_i = 1, \tag{2.2}$$

have a unique solution $\pi = (\pi_1, \ldots, \pi_N)$. We refer to π as the steady state probability vector of the Markov chain with transition probabilities P_{ij}, and to π_i as the steady state probability of being in state i.

We are interested in the situation where we receive a one-state reward g_i whenever we visit the state $i \in S$. As the Markov chain proceeds from state to state, there is a reward sequence $(g_{i_0}, g_{i_1}, g_{i_2}, \ldots)$ associated with the trajectory (state sequence) $(i_0, i_1, i_2, \ldots)$ of the Markov chain. For this reward process, we define the average reward λ by

$$\lambda = \lim_{t \to \infty} \frac{1}{t} E\left[\sum_{k=0}^{t} g_{i_k}\right],$$

where the expectation $E[\cdot]$ is taken with respect to the probability measure induced by the Markov chain with transition probabilities P_{ij}. Under Assumption 1, the average reward λ is well defined and does not depend on the initial state i_0, and we have the following relation

$$\lambda = \sum_{i \in S} \pi_i g_i. \tag{2.3}$$

We define the differential reward v_i of state i by

$$v_i = E\left[\sum_{k=0}^{T-1} (g_{i_k} - \lambda) \mid i_0 = i\right],$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. With this definition we have $v_{i^*} = 0$. When the Markov chain is ergodic, then the differential reward has the following interpretation. The differential reward v_i of state i captures the relative advantage of starting the reward process in state i, rather than in the recurrent state i^*, i.e.,

$$v_i = \lim_{t \to \infty} \left(E\left[\sum_{k=0}^{t} g_{i_k} \mid i_0 = i\right] - E\left[\sum_{k=0}^{t} g_{i_k} \mid i_0 = i^*\right]\right).$$

Under Assumption 1, the the differential reward vector v is the unique solution to the following system of linear equations,

$$\begin{aligned} v_i &= g_i - \lambda + \sum_{j \in S} P_{ij} v_j, \qquad \text{for all } i \in S \backslash \{i^*\}, \\ v_{i^*} &= 0, \end{aligned}$$

where the average reward λ is given by Eq.(2.3).

A special case where Assumption 1 holds, is characterized by the following condition.

Assumption 2 *There exists a state $i^* \in S$ such that $g_{i^*} = 0$ and $P_{i^* i^*} = 1$, and, for all states $i \in S$, we have*

$$P_{ii^*}^N > 0,$$

where N is the number of states in the state space S.

The state i^* has the interpretation of a reward-free terminal state which is eventually reached from every state $i \in S$.

Note that under Assumption 2, the steady state probability vector π is given by

$$\pi_i = \begin{cases} 1, & \text{if } i = i^*, \\ 0, & \text{otherwise,} \end{cases}$$

and the average reward λ is equal to 0.

When Assumption 2 holds, we define the reward-to-go J_i of state i by

$$J_i = E\left[\sum_{k=0}^{\infty} g_{i_k} \mid i_0 = i\right] = E\left[\sum_{k=0}^{T-1} g_{i_k} \mid i_0 = i\right],$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. The reward-to-go J_i of state i is the total expected reward obtained along a trajectory which starts at state i. With this definition we have $J_{i^*} = 0$.

Under Assumption 2 holds, the reward-to-go vector J is the unique solution to the following system of linear equations,

$$\begin{aligned} J_i &= g_i + \sum_{j \in S} P_{ij} J_j, \qquad \text{for all } i \in S \backslash \{i^*\}, \\ J_{i^*} &= 0. \end{aligned}$$

2.2 Markov Decision Processes

In this section, we generalize Markov reward processes to Markov decision processes.

We consider a discrete-time Markov decision process which is defined on a finite state space $S = \{1, ..., N\}$. At each state $i \in S$, a decision maker can choose a control action u from a finite set $U = \{1, ..., L\}$. The control action u determines both the dynamics of the system at state i, as well as the reward the decision maker obtains at state i. More precisely, given that the decision maker applies control action $u \in U$ at state $i \in S$, a transition from state i to any particular state $j \in S$ occurs with probability $P_{ij}(u)$, and the decision maker receives the one-stage reward $g_i(u)$.

We define a pure stationary policy to be a mapping

$$\mu : S \mapsto U.$$

A stationary policy specifies, for each state $i \in S$, the control action $u \in U$ that the decision maker applies in state i. Let M be the set of all possible stationary policies, which is finite as both the state space S and the set of control actions U are finite. Note that a stationary policy $\mu \in M$ defines a Markov reward process on the state space S. In the following, we use the notation $P_{ij}(\mu)$, and $P_{ij}^n(\mu)$, to denote the one-step transition probabilities, and $n-$step transition probabilities, respectively, for the Markov chain associated with a given stationary policy $\mu \in M$, i.e.,

$$P_{ij}(\mu) = P(i_1 = j \mid i_0 = i, \mu), \quad \text{and} \quad P_{ij}^n(\mu) = P(i_n = j \mid i_0 = i, \mu).$$

Furthermore, we use the notation $E_\mu[\cdot]$ to indicate that the expectation is taken with respect to the probability measure on the state space S induced by the Markov chain with transition probabilities $P_{ij}(\mu)$.

Here we are interested in the situation where the decision maker tries to choose a stationary policy $\mu \in M$ to satisfy a given optimality criterion regarding the expected reward obtained under policy μ. In the next two subsections, we introduce the two optimality criteria that we consider in this thesis.

2.2.1 Optimizing the Reward-to-Go

In this subsection, we consider Markov decision processes for which the following assumption holds.

Assumption 3 (MDP Termination) *There exists a state $i^* \in S$ such that, for every control action $u \in U$, we have*

$$g_{i^*}(u) = 0, \ \textit{and} \ \ P_{i^* i^*}(u) = 1,$$

and, for all stationary policies $\mu \in M$ and all states $i \in S$, we have

$$P_{ij}^N(\mu) > 0,$$

where N is the number of states in the state space S.

When Assumption 3 holds, the reward-to-go $J_i(\mu)$ of state i under policy μ is given by

$$J_i(\mu) = E_\mu\left[\sum_{k=0}^{\infty} g_{i_k}(\mu(i_k)) \mid i_0 = i\right] = E_\mu\left[\sum_{k=0}^{T-1} g_{i_k}(\mu(i_k)) \mid i_0 = i\right],$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. From our discussion of Markov reward processes in Section 2.1, it follows that the reward-to-go vector $J(\mu)$ can be obtained by solving the following system of linear equations,

$$\begin{aligned} J_i &= g_i(\mu(i)) + \sum_{j \in S} P_{ij}(\mu) J_j, \qquad \text{for all } i \in S \backslash \{i^*\}, \\ J_{i^*} &= 0. \end{aligned}$$

In this case, the goal of the decision maker is to find a control policy $\mu^* \in M$ which maximizes the reward-to-go $J_i(\mu)$ simultaneously for all states $i \in S$, i.e.,

$$J_i(\mu^*) \geq J_i(\mu), \qquad \text{for all } \mu \in M, \text{ and all } i \in S. \tag{2.4}$$

We refer to a policy which satisfies condition (2.4) as an optimal policy. There may be several policies that are optimal. However, there exists a unique vector J^* such that, for all optimal policies μ^*, we have $J(\mu^*) = J^*$. We refer to the vector J^* as the optimal reward-to-go vector.

2.2.2 Optimizing the Average Reward

In this subsection, we consider Markov decision processes for which the following assumption holds.

Assumption 4 (MDP Recurrence) *There exists a state $i^* \in S$ such that, for every policy $\mu \in M$ and for every state $i \in S$, we have*

$$\sum_{n=1}^{N} P^n_{ii^*}(\mu) > 0,$$

where N is the number of states in the state space S.

If Assumption 4 holds, the differential reward $v_i(\mu)$ of state i under policy μ is given by

$$v_i(\mu) = E_\mu \left[\sum_{k=0}^{T-1} \left(g_{i_k}(\mu(i_k)) - \lambda(\mu)\right) \mid i_0 = i \right], \tag{2.5}$$

where

$$\lambda(\mu) = \lim_{t \to \infty} \frac{1}{t} E_\mu \left[\sum_{k=0}^{t} g_{i_k}(\mu(i_k)) \right], \tag{2.6}$$

and $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. From our discussion of Markov reward processes in Section 2.1, it follows that when Assumption 4 holds, then the average reward $\lambda(\mu)$ is well defined for every policy $\mu \in M$ and does not depend on the initial state i_0. We have the following relation

$$\lambda(\mu) = \sum_{i \in S} \pi_i(\mu) g_i(\mu(i)), \tag{2.7}$$

where the $\pi_i(\mu)$ is the unique solution to the balance equations given by Eqs.(2.1)-(2.2) for the transition probabilities $P_{ij}(\mu)$. The differential reward vector $v(\mu)$ can be computed by solving the following system of linear equations,

$$\begin{aligned} v_i &= g_i(\mu(i)) - \lambda(\mu) + \sum_{j \in S} P_{ij}(\mu) v_j, \qquad \text{for all } i \in S \backslash \{i^*\}, \\ v_{i^*} &= 0, \end{aligned}$$

where $\lambda(\mu)$ is given by Eq.(2.7).

When Assumption 4 holds, we are interested in the situation where a decision maker tries to find a control policy $\mu^* \in M$ which maximizes the average reward, i.e.,

$$\lambda(\mu^*) \geq \lambda(\mu), \qquad \text{for all } \mu \in M. \tag{2.8}$$

We refer to a policy μ^* which satisfies condition (2.8) as an optimal policy, and to $\lambda^* = \lambda(\mu^*)$ as the optimal average reward. Again, there may be several optimal policies. However, there exists a unique vector v^* such that, for all optimal policies μ^*, we have $v(\mu^*) = v^*$. We refer to v^* as the optimal differential reward vector.

2.3 Dynamic Programming

In the previous section, we introduced Markov decision processes, and considered the situation where a decision maker tries to find a stationary control policy that satisfies a certain optimality criterion. Dynamic programming provides a framework for studying Markov decision processes, as well as for devising algorithms for computing optimal control policies. Here, we provide a brief overview of dynamic programming, and point out why methods of dynamic programming cannot be applied to many problems.

Optimizing the Reward-to-go

Let us first consider a Markov decision process for which Assumption 3 (MDP Termination), i.e., there exists a reward-free terminal state $i^* \in S$ that is eventually reached from every state $i \in S$. A central result from dynamic programming states that the optimal reward-to-go vector J^* is the unique solution J^* to Bellman's optimality equations,

$$\begin{aligned} J_i &= \max_{u \in U} \left\{ g_i(u) + \sum_{j \in S} P_{ij} J_j \right\}, \qquad \text{for all } i \in S \backslash \{i^*\}, \\ J_{i^*} &= 0. \end{aligned}$$

Furthermore, for the optimal reward-to-go vector J^*, a policy $\mu^* \in M$ which has the property that

$$\mu^*(i) = \arg\max_{u \in U} \left\{ g_i(u) + \sum_{j \in S} P_{ij}(u) J_j^* \right\}, \qquad \text{for all } i \in S, \tag{2.9}$$

is an optimal policy. In other words, the theory of dynamic programming tells us that if we know the optimal reward-to-go vector J^*, then we are able to derive an optimal policy by using the rule given by Eq.(2.9). Based on this result, methods of dynamic programming compute first the optimal reward-to-go vector J^*, and then derive an optimal control policy by using Eq.(2.9). There several algorithms for computing an optimal control policy in this way. Here we discuss only policy iteration.

Policy iteration iteratively generates reward-to-go vectors, and policies, by starting with an initial vector $J^{(0)} \in \Re^N$, where N is the number of states in S, and by setting

$$\mu^{(k)}(i) = \arg\max_{u \in U} \left\{ g_i(u) + \sum_{j \in S} P_{ij}(u) J_j^{(k)} \right\}, \qquad \text{for all } i \in S,$$

and $J^{(k+1)} = J(\mu^{(k)})$, where $J(\mu^{(k)})$ is the reward-to-go vector of policy $\mu^{(k)}$. This iteration is carried out, until we have $J^{(k+1)} = J^{(k)}$.

Under Assumption 3, for every initial vector $J^{(0)}$, this algorithm is guaranteed to terminate with the optimal reward-to-go vector J^*, and therefore with an optimal control policy μ^*, in a finite number of steps.

Optimizing the Average Reward

Similarly, given a Markov decision process for which Assumption 4 holds, the theory of dynamic programming asserts that the optimal differential reward vector v^* is the unique solution to the optimality equations

$$v_i = \max_{u \in U} \left\{ g_i(u) - \lambda^* + \sum_{j \in S} P_{ij} v_j \right\}, \qquad \text{for all } i \in S \backslash \{i^*\},$$

and $v_{i^*} = 0$, where λ^* is the optimal average reward. For the optimal differential reward vector v^*, a policy μ^* which has the property that

$$\mu^*(i) = \arg\max_{u \in U} \left\{ g_i(u) + \sum_{j \in S} P_{ij}(u) v_j^* \right\}, \qquad \text{for all } i \in S, \tag{2.10}$$

is an optimal control policy.

A method for computing the optimal differential reward vector is Howard's policy iteration algorithm. Again, the algorithm iteratively generates differential reward vectors, and policies, by starting with an initial vector $v^{(0)} \in \Re^N$, where N is the number of states in S, and by setting

$$\mu^{(k)}(i) = \arg\max_{u \in U} \left\{ g_i(u) + \sum_{j \in S} P_{ij}(u) v_j^{(k)} \right\}, \qquad \text{for all } i \in S,$$

and $v^{(k+1)} = v(\mu^{(k)})$, where $v(\mu^{(k)})$ is the differential reward vector of policy $\mu^{(k)}$. This iteration is carried out, until we have $v^{(k+1)} = v^{(k)}$.

Under Assumption 4, for every initial vector $v^{(0)}$, this algorithm is guaranteed to terminate with the optimal differential reward vector v^*, and therefore with an optimal control policy μ^*, in a finite number of steps.

2.3.1 The Curse of Dimensionality and the Curse of Modeling

As discussed in Chapter above, methods of dynamic programming allow us to obtain an optimal control policy for a Markov decision process in the following way.

- First, the optimal reward-to-go, and the optimal differential reward, respectively, are computed for every state $i \in S$.
- Second, the optimal reward-to-go vector, and the optimal differential reward vector, respectively, are used, together with the mathematical model of the system, to derive an optimal control policy.

However, this procedure cannot be applied to problems which evolve over a large state space or for which an exact mathematical model is not available. For these situations, rather than using an optimal control policy obtained by methods of dynamic programming, one typically resorts to using a heuristic control policy. The drawback of this approach is that it is often difficult and time-consuming to devise a good heuristic policy. In this thesis, we consider an alternative approach which allows us to compute control policies in a systematic way, without requiring the computational burden of dynamic programming. In addition, we show that the resulting policies are optimal in a sense to be made more precise later. Before we introduce the precise mathematical framework, we illustrate the theory presented in this chapter with a specific application.

2.4 Call Admission Control

There is a broad range of problems that can be modeled as Markov decision processes. Basically, Markov decision processes can be employed whenever we encounter a situation where (control) decisions are made in stages; with the additional constraint that the one-stage reward, and the transition to a next state, depend only on the current state and the (control) decision made at the current stage. In this section, we provide a specific application. In particular, we consider the call admission control problem for a single link which arises when a telecommunication provider wants to sell the bandwidth of the link to customers so as to maximize the expected long term revenue. We will revisit this problem in Chapter 7 and provide a overview of proposed approaches to compute call admission control policies.

We consider a single communication link with a total bandwidth of B units, which supports a finite set $\{1, 2, .., M\}$ of different service types. Each service type is characterized by its bandwidth demand $b(m)$, its call arrival rate $\alpha(m)$, and its average holding time $1/\beta(m)$, where we assume that the calls arrive according to independent Poisson Processes, and that the holding times are exponentially (and independently) distributed. When a new customer requests a connection, the link provider can decide to reject, or, if enough bandwidth is available, to accept the customer (*Call Admission Control*). Once accepted, a customer of class m seizes $b(m)$ units of bandwidth for the duration of the call. Whenever a call of service type m gets accepted, the provider receives an immediate reward of $c(m)$ units. The reward $c(m)$ can be interpreted as the price customers of service type m are paying for using $b(m)$ units of bandwidth of the link for the duration of the call. The goal of the link provider is to exercise call admission control so as to maximize the long term revenue.

We will formulate the call admission control problem as a Markov decision process. In particular, we will define the underlying state space S, the control space U, the transition probabilities $P_{ij}(u)$, and the one-stage reward $g_i(u)$, for this problem. We start out by defining the configuration of the communication link by an $M-$tuple $s = (s(1), ..., s(M))$, where $s(m)$ denotes the number of customers of class m currently using the link. The set of all possible configurations S_0 of the link is then given by

$$S_0 = \left\{ s \in \Re^M \;\middle|\; \sum_{m=1}^{M} s(m)b(m) \leq B, \quad s(m) \in \{0, 1, 2, ...\} \right\}.$$

Transitions from a configuration s to a next configuration s' are triggered by events such as the arrival of a new call, or the departure of a call. Here, we formally denote an event as a M-tuple $\omega = (\omega(1), ..., \omega(M))$ where $\omega(m)$ equals 1 if a new customer of class m requests a connection, $\omega(m)$ equals -1 if a customer of class m departs from the system and $\omega(m)$ equals 0 otherwise. We allow (for reasons which will become apparent later) the trivial event $\omega = (0, ..., 0)$, which we denote as ω^*. The set Ω of all possible events is then given by

$$\Omega = \left\{ \omega \;\middle|\; \omega \in \{-1, 0, 1\}^M, \;\; \sum_{m=1}^{M} |\omega(m)| \leq 1 \right\}$$

We find it convenient to work with an augmented state $i = (s, \omega)$ of the link which consists of the current configuration of the link s and the event ω which triggers a transition. The state space S is then given by

$$S = \{i = (s, \omega) \mid s \in S_0, \omega \in \Omega, \text{ and } \omega_m \geq 0 \text{ if } s(m) = 0\},$$

where the last condition states that the next event is possibly a departure of a call of service type m, only if there is currently a call of that type using the link.

At a state $i = (s, \omega)$ for which the event ω corresponds to an arrival of a new call, the control actions available to the link provider are to accept or reject that new call. We define the set of possible control actions as $U = \{u_a, u_r\}$, where u_a stands for accepting and u_r stands for rejecting.

To define the transition probabilities $P_{ij}(u)$ under control actions u, we will use a technique called uniformization [Gal95] to formulate the admission control problem as a discrete-time problem. In particular, let r be defined by

$$r = \max_{s \in S} \left\{ \sum_{m=1}^{M} (\alpha(m) + s(m)\beta(m)) \right\},$$

and, for $s \in S_0$, let $r(s)$ be defined by

$$r(s) = \sum_{m=1}^{M} (\alpha(m) + s(m)\beta(m)).$$

Note that, for a configuration $s \in S_0$, the probability that the next event ω is an arrival of a call of service type m is equal to $\alpha(m)/r$, the probability that the next event ω is a departure of a call of service type m is equal to $s(m)\beta(m)/r$, and the probability that then next event ω is the trivial event $(0, ..., 0)$ is equal to $1 - r(s)/r$.

The probability $P_{ij}(u)$ of a transition from state $i = (s, \omega) \in S$ to the state $j = (s', \omega') \in S$ under control action u is then given as follows

1. if ω corresponds to an arrival of a new call, $u = u_a$ and $s + \omega \in S$, then

$$P_{ij}(u) = \begin{cases} \alpha(m)/r, & \text{if } s' = s + \omega, \text{ and } \omega'_m = 1, \\ s'(m)\beta(m)/r, & \text{if } s' = s + \omega, \text{ and } \omega'_m = -1, \\ 1 - r(s')/r, & \text{if } s' = s + \omega, \text{ and } \omega'_m = \omega^*, \end{cases}$$

2. if ω corresponds to an arrival of a new call and $u = u_r$, or ω corresponds to an arrival of a new call, $u = u_a$ and $s + \omega \notin S$, then

$$P_{ij}(u) = \begin{cases} \alpha(m)/r, & \text{if } s' = s, \text{ and } \omega'_m = 1, \\ s'(m)\beta(m)/r, & \text{if } s' = s, \text{ and } \omega'_m = -1, \\ 1 - r(s')/r, & \text{if } s' = s, \text{ and } \omega' = \omega^*. \end{cases}$$

3. if ω corresponds to a departure of a call, then

$$P_{ij}(u) = \begin{cases} \alpha(m)/r, & \text{if } s' = s + \omega, \text{ and } \omega'_m = 1, \\ s'(m)\beta(m)/r, & \text{if } s' = s + \omega, \text{ and } \omega'_m = -1, \\ 1 - r(s')/r, & \text{if } s' = s + \omega, \text{ and } \omega'_m = \omega^*. \end{cases}$$

4. if $\omega = (0, ..., 0)$, then

$$P_{ij}(u) = \begin{cases} \alpha(m)/r, & \text{if } s' = s, \text{ and } \omega'_m = 1, \\ s'(m)\beta(m)/r, & \text{if } s' = s, \text{ and } \omega'_m = -1, \\ 1 - r(s')/r, & \text{if } s' = s, \text{ and } \omega'_m = \omega^*. \end{cases}$$

For all other cases, the transition probabilities $P_{ij}(u)$ are equal to 0.

The one-stage reward at state $i = (s, \omega) \in S$ is equal to $c(m)$ whenever a call of service type m gets accepted, or more formally,

$$g_i(u) = \begin{cases} c(m), & \text{if } u = u_a,\ \omega_m = 1,\ s + \omega \in S_0, \\ 0, & \text{otherwise.} \end{cases}$$

Note that for every initial configuration $s \in S$, the link will eventually empty out, i.e., the configuration $s^* = (0, ..., 0) \in S$ will be reached from every initial state $s \in S$ with a positive probability. To see this, note that for every configuration $s \neq s^*$, there is a positive probability that the next event will be a departure of a call. Furthermore, observe that $r(s^*)$ is positive. This implies that the call admission control problem satisfies Assumption 4 with state $i^* = (s^*, \omega^*)$ serving as the recurrent state, and that there exists an optimal call admission control policy $\mu^* : S \mapsto U$ which maximizes the average revenue the link provider obtains.

Note that the number of states in the state space S increases in a combinatorial way in the number of customer classes. Therefore, if there is a large number of different customer classes competing for the bandwidth of the link, the provider may not be able to employ methods of dynamic programming to compute an optimal call admission control policy and maximize the expected long term revenue. As an alternative approach, the link provider may resort to some heuristic, such as a threshold policy ([MT97]), to exercise call admission control.

Chapter 3

Formulation

In this chapter, we present the mathematical framework that we use in our analysis. The structure is the following. Section 3.1 defines randomized policies that are described in terms of a parameter vector. In Section 3.2, we introduce Markov reward processes which depend on a parameter. In Section 3.3, we give give an overview of gradient-based optimization methods which is the optimization technique we consider in this thesis.

3.1 Parameterized Control Policies

In this section, we introduce randomized control policies which are described in terms of a parameter vector $\theta \in \Re^K$.

We consider a discrete-time Markov decision process which is defined on a finite state space $S = \{1, ..., N\}$, and a finite set of control actions $U = \{1, ..., L\}$. A randomized control policy is given by a mapping

$$\mu : S \mapsto [0,1]^L,$$

with components $\mu_u(i)$, $u \in U$, such that for all states $i \in S$ we have

$$\sum_{u \in U} \mu_u(i) = 1.$$

Under the randomized policy μ, the decision maker applies at state i the control action u with probability $\mu_u(i)$, independent of everything else.

For problems involving very large state spaces, it is impossible to even describe an arbitrary randomized policy μ, since this requires a listing of the probabilities of choosing a control action $u \in U$ corresponding to every state. This leads us to consider policies described in terms of a parameter vector $\theta \in \Re^K$, whose dimension K is tractable small. With every $\theta \in \Re^K$, we associate a randomized policy $\mu(\theta)$, which at any given state i chooses action u with probability $\mu_u(i, \theta)$. Naturally, we require that, for every state $i \in S$ and every control action $u \in U$, the component $\mu_u(i, \theta)$ is nonnegative and that $\sum_{u \in U} \mu_u(i, \theta) = 1$. Let us consider the following two examples.

Example 1 *In Section 2.4 we considered the situation where a communication link provider defines a call admission control policy by specifying whether a new call gets accepted or rejected at a given state i. We denoted the control action which corresponds to accepting a new call by u_a, and the control action action which corresponds to rejecting a new call by u_r*

In an alternative approach, using randomized control policies, the link provider defines a control policy by specifying the probability with which a new call gets accepted. Here, we consider a particular choice of such a randomized control policy which is given in terms of a parameter vector $\theta \in \Re^M$. If for $i = (s, \omega)$, the event ω indicates the arrival of a call of service type m, then the call gets accepted with probability

$$\mu_{u_a}(i, \theta) = \frac{1}{1 + \exp(s \cdot b - \theta_m)},$$

where the vector $s = (s(1), ..., s(M))$ specifies the number of users of each service type m using the link, and $b = (b(1), ..., b(M))$ is a vector that specifies the bandwidth demands of the different service types. Accordingly, the provider rejects a new call at state i with probability $\mu_{u_r}(i, \theta) = 1 - \mu_{u_a}(i, \theta)$.

The following example is due to [Mih98].

Example 2 *More generally, let $r_u(i, \theta)$, $\theta \in \Re^K$, be a parameterized function that maps states and actions to real numbers, and consider the case where the decision maker chooses at state i control action u with probability*

$$\mu_u(i, \theta) := \frac{\exp(r_u(i, \theta))}{\sum\limits_{u' \in U} \exp(r_{u'}(i, \theta))}.$$

Obviously, we have $\mu_u(i, \theta) \geq 0$ and $\sum_{u \in U} \mu_u(i, \theta) = 1$, and $\mu_u(i, \theta)$ defines a randomized policy.

One may interpret the function $r_u(i, \theta)$ as a "utility" function associated with state i and control action u, in the sense that the control action with the largest value $r_u(i, \theta)$ is most likely to be chosen at state i.

Note that, under policy $\mu(\theta)$, the transition probabilities, which we denote by $P_{ij}(\theta)$, are given by

$$P_{ij}(\theta) = \sum_{u \in U} \mu_u(i, \theta) P_{ij}(u), \tag{3.1}$$

and the expected one-stage reward at state i, which we denote by $g_i(\theta)$, is given by

$$g_i(\theta) = \sum_{u \in U} \mu_u(i, \theta) g_i(u). \tag{3.2}$$

Eqs.(3.1) and (3.2) suggest that we can associate with every parameter vector $\theta \in \Re^K$ a Markov reward process with transition probabilities $P_{ij}(\theta)$ and one-state rewards $g_i(\theta)$. This leads us to consider Markov Reward Processes that depend on a parameter vector.

3.2 Markov Reward Processes Depending on a Parameter

In this section, we introduce Markov reward processes that depend on a parameter $\theta \in \Re^K$. This section serves as a basis for the analysis in the following chapters.

We consider a discrete-time, finite-state Markov chain on the state space $S = \{1, \ldots, N\}$, whose transition probabilities depend on a parameter vector $\theta \in \Re^K$ and are denoted by $P_{ij}(\theta)$, $i, j \in S$. Furthermore, with every state $i \in S$, we associate a one-stage reward that also depends on θ and is denoted by $g_i(\theta)$.

In the following, we use the notation $P_\theta(\cdot)$, to denote the probability distribution induced by the Markov chain with transition probabilities $P_{ij}(\theta)$. Accordingly, we use $E_\theta[\cdot]$ and $\mathrm{Var}_\theta(\cdot)$ to indicate that the expectation and the variance, respectively, are taken with respect to the probability distribution of the Markov chain with transition probabilities $P_{ij}(\theta)$.

We assume that the following regularity condition holds for the transition probabilities $P_{ij}(\theta)$ and the one-stage reward $g_i(\theta)$.

Assumption 5 (MRP Parameterization) *For all states $i, j \in S$, the transition probability $P_{ij}(\theta)$, and the one-state reward $g_i(\theta)$, are bounded, twice differentiable, and have bounded first and second derivatives. Furthermore, for all states $i, j \in S$, we have*

$$\nabla P_{ij}(\theta) = P_{ij}(\theta) f_{ij}(\theta),$$

where the function $f_{ij}(\theta)$ is bounded, differentiable, and has bounded first derivatives.

This assumption, which assures that the transition probabilities $P_{ij}(\theta)$ and the one-stage reward $g_i(\theta)$ are "smooth" in θ, will become important when we study gradient methods for tuning θ. We will revisit Assumption 5 in Chapter 6, and provide examples for which this regularity condition hold.

For every parameter vector $\theta \in \Re^K$, let $P(\theta)$ be the transition matrix with entries $P_{ij}(\theta)$, and let $g(\theta) = (g_1(\theta), ..., g_N(\theta))$ be the reward vector associated with the parameter vector θ. Furthermore, let $\mathcal{P} = \{P(\theta) \mid \theta \in \Re^K\}$ be the set of all possible transition matrices. Here, we think of $\mathcal{P}$ as being a subset of $[0,1]^{N^2}$, where N is the number of states in the state space S, and let $\overline{\mathcal{P}}$ be its closure. Note that $\overline{\mathcal{P}}$ is a compact set, as it is closed and bounded. We have the following lemma for the elements of $\overline{\mathcal{P}}$.

Lemma 1 *Every element $P \in \overline{\mathcal{P}}$ is a stochastic matrix i.e., P is nonnegative and, for every state $i \in S$, we have*

$$\sum_{j \in S} P_{ij}(\theta) = 1.$$

Proof: Let P be a matrix in the set $\overline{\mathcal{P}}$. Then there exists a sequence (θ_k) of parameter vectors in $\Re^K$ such that

$$\lim_{k \mapsto \infty} P(\theta_k) = P. \tag{3.3}$$

By definition, for every $k = 1, 2, ...$, the matrix $P(\theta_k)$ is a stochastic matrix, i.e., is nonnegative and for every state $i \in S$, we have

$$\sum_{j \in S} P_{ij}(\theta_k) = 1. \tag{3.4}$$

Combining (3.3) and (3.4), it follows that for every $i \in S$, we have

$$1 = \lim_{k \mapsto \infty} \sum_{j \in S} P_{ij}(\theta_k) = \sum_{j \in S} P_{ij},$$

and that P is a stochastic matrix. □

Lemma 1 implies that every element $P \in \overline{\mathcal{P}}$ can also be interpreted as a transition matrix of a Markov chain on the state space S.

We are interested in the situation where a decision maker tries to choose a parameter vector $\theta \in \Re^K$, so as to optimize a given performance measure associated with the Markov reward process with the parameter vector θ. In particular, we consider two performance measures, the weighted reward-to-go and the average reward, which are based on the optimality criteria introduced in Section 2.2 for Markov decision processes.

3.2.1 Optimizing the Weighted Reward-to-Go

In this subsection, we consider Markov reward processes for which the following assumption holds.

Assumption 6 (MRP Termination) *There exists a state $i^* \in S$, such that, for every parameter vector $\theta \in \Re^K$, we have*

$$g_{i^*}(\theta) = 0 \qquad \textit{and} \qquad P_{i^* i^*}(\theta) = 1,$$

and, for every state $i \in S$ and every transition matrix $P \in \overline{\mathcal{P}}$, we have

$$P^N_{ii^*} > 0,$$

where N is the number of states in the state space S

Assumption 6 can be interpreted as the generalization of Assumption 3, which we made in Section 2.2 for Markov decision processes. In particular, the state i^* in Assumption 6 has the interpretation of a reward-free terminal state, which is eventually reached from every state $i \in S$.

Assumption 6 implies that there exists a constant $\rho > 0$, such that for every state $i \in S$ and every transition matrix $P \in \overline{\mathcal{P}}$, we have

$$P^N_{ii^*} \geq \rho.$$

This follows from the fact that the set $\overline{\mathcal{P}}$ is compact and that the transition probabilities $P_{ij}(\theta)$ are continuous in θ.

When Assumption 6 holds, the reward-to-go $J_i(\theta)$ of state i is given by

$$J_i(\theta) = E_\theta\left[\sum_{k=0}^{\infty} g_{i_k}(\theta) \mid i_0 = i\right] = E_\theta\left[\sum_{k=0}^{T-1} g_{i_k}(\theta) \mid i_0 = i\right], \tag{3.5}$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. We have that $J_{i^*}(\theta) = 0$.

The following lemma states that $J(\theta)$ depends smoothly on θ.

Lemma 2 *Let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold. Then, $J(\theta)$ is bounded, twice differentiable, and has bounded first and second derivatives.*

We provide the proof for Lemma 2 in Appendix A.

When Assumption 6 holds, then we are interested in the situation where the decision maker tries to find a parameter vector which maximizes the weighted reward-to-go $\chi(\theta)$ which is defined by

$$\chi(\theta) = \sum_{i \in S} \bar{\pi}_i J_i(\theta), \tag{3.6}$$

where $\bar{\pi}$ is a given probability distribution on the state space S. The weighted reward-to-go captures the situation where the decision maker wants to maximize the reward-to-go, knowing that the initial state i_0 is equal to a specific state $i \in S$ with probability $\bar{\pi}$.

The situation which we consider here is somewhat different from the one in Section 2.2.1. In Section 2.2.1., the goal of the decision maker was to find a control policy which maximizes the reward-to-go simultaneously for all states. The existence of such an optimal control policy was guaranteed by theory of dynamic programming. However, for parameterized Markov reward processes, it is generally not true that there exists a parameter vector $\theta^* \in \Re^K$ which maximizes the reward-to-go simultaneously for all states, or maximizes the weighted reward-to-go associated with a given probability distribution $\bar{\pi}$. Furthermore, an optimal parameter vector θ^* (if it exists) may depend on the particular choice of the probability distribution $\bar{\pi}$. However, we have the following result.

Lemma 3 *If there exists a parameter vector $\theta^* \in \Re^K$, such that for all vectors $\theta \in \Re^K$ and for all states $i \in S$ we have*

$$J_i(\theta^*) \geq J_i(\theta),$$

then θ^* maximizes the weighted average reward-to-go $\chi(\theta)$ for every probability distribution $\bar{\pi}$ over the state space S.

Proof: Let $\theta^* \in \Re^K$ be such that, that for all vectors $\theta \in \Re^K$ and for all states $i \in S$, we have

$$J_i(\theta^*) \geq J_i(\theta),$$

and let $\bar{\pi}$ be a probability distribution on the state space S. As $\bar{\pi}_i$ is non-negative for every state $i \in S$, it follows that

$$\chi(\theta^*) = \sum_{i \in S} \bar{\pi}_i J_i(\theta^*) \geq \sum_{i \in S} \bar{\pi}_i J_i(\theta) = \chi(\theta).$$

□

Lemma 3 states that if the parameter vector θ^* maximizes the reward-to-go simultaneously for all states, then θ^* is also a solution for the maximization problem associated with the weighted reward-to-go $\chi(\theta)$, independent of the choice of the probability distribution $\bar{\pi}$. However, it is generally difficult to check in practice whether there exists such an optimal parameter vector.

3.2.2 Optimizing the Average Reward

In this subsection, we consider Markov reward processes for which the following assumption holds.

Assumption 7 (MRP Recurrence) *There exists a state $i^* \in S$, such that, for every state $i \in S$ and every transition matrix $P \in \overline{\mathcal{P}}$, we have*

$$\sum_{n=1}^{N} P_{ii^*}^n > 0,$$

where N is the number of states in the state space S.

Assumption 7 can be interpreted as the generalization of Assumption 4, which we made in Section 2.2 for Markov decision processes.

Note that Assumption 7 implies that there exists a constant $\rho > 0$, such that for every state $i \in S$ and every transition matrix $P \in \overline{\mathcal{P}}$, we have

$$\sum_{n=1}^{N} P_{ii^*}^n \geq \rho.$$

Assumption 7 implies that for every $\theta \in \Re^K$, the average reward

$$\lambda(\theta) = \lim_{t \mapsto \infty} \frac{1}{t} E_\theta \left[\sum_{k=0}^{t} g_{i_k}(\theta) \mid i_0 = i \right]$$

is well defined, and does not depend on the initial state i_0.

Furthermore, under Assumption 7, the balance equations

$$\sum_{i=1}^{N} \pi_i P_{ij}(\theta) = \pi_j, \qquad j = 1, \ldots, N-1, \tag{3.7}$$

$$\sum_{i=1}^{N} \pi_i = 1, \tag{3.8}$$

have a unique solution $\pi(\theta) = (\pi_1(\theta), \ldots, \pi_N(\theta))$.

The mean recurrence time $E_\theta[T]$, and the differential reward $v_i(\theta)$ of state $i \in S$, are given by

$$E_\theta[T] = E_\theta[T \mid i_0 = i^*], \text{ and} \tag{3.9}$$

$$v_i(\theta) = E_\theta\left[\sum_{k=0}^{T-1} (g_{i_k}(\theta) - \lambda(\theta)) \mid i_0 = i\right], \tag{3.10}$$

respectively, where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. We have that $v_{i^*}(\theta) = 0$.

The following lemma states that $\pi(\theta)$, $\lambda(\theta)$, $E_\theta[T]$ and $v(\theta)$ depend "smoothly" on θ.

Lemma 4 *Let Assumptions 5 (MRP Parameterization) and 7 (MRP Recurrence) hold. Then, $\pi(\theta)$, $\lambda(\theta)$, $E_\theta[T]$ and $v(\theta)$ are, as functions of θ, bounded, twice differentiable, and have bounded first and second derivatives. Furthermore, for every integer $s > 0$, there exists a constant D_s, such that for all $\theta \in \Re^K$, we have*

$$E_\theta[T^s] = E_\theta[T^s \mid i_0 = i^*] \le D_s,$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited.

We provide the proof for Lemma 4 in Appendix A.

When Assumption 7 holds, then we are interested in the situation where the decision maker tries to find a parameter vector θ to optimize the average reward $\lambda(\theta)$.

3.3 Gradient-Based Optimization

In this section, we discuss gradient-based algorithms which we will use as a method for tuning the parameter vector θ and for optimizing the weighted reward-to-go and the average reward. We use this section to introduce the basic form of the algorithms, and to state the corresponding assumptions. We will provide a more precise formulation in the following chapters.

For tuning the parameter vector θ we consider methods where, starting with an initial vector θ_0, the decision maker iteratively updates the parameter vector along a (approximate) gradient direction. In particular, in the case where the gradient of the performance measure $\psi(\theta)$ with respect to θ can be computed exactly, we consider the gradient algorithm given by the update rule

$$\theta_{k+1} = \theta_k + \gamma_k \nabla\psi(\theta_k),$$

where γ_k is a small step size parameter. For this algorithm, under suitable assumptions on the step size parameter and on the performance metric, we have that $\psi(\theta_k)$ converges, and $\lim_{k\to\infty} \nabla\psi(\theta_k) = 0$ ([Ber95b]).

Unfortunately, for Markov reward processes involving a large state space, it is (generally) not possible to compute the gradient of the average reward $\lambda(\theta)$, or the weighted reward-to-go $\chi(\theta)$, exactly. As an alternative approach, we consider the case where the decision maker employs simulation to produce an estimate of the gradient $\nabla\psi(\theta)$, and updates the parameter vector along an approximate gradient iteration. The update can take place either at visits to a certain state i^*, or at every time step.

In the first case, the algorithm is the following. At the $(m+1)$th visit to i^*, θ_m is available from the previous iteration. Using an estimate $F_m(\theta)$ of the gradient based on observations between the mth and $(m+1)$th visit to i^*, we update θ according to

$$\theta_{m+1} = \theta_m + \gamma_m F_m(\theta_m), \tag{3.11}$$

We can rewrite this iteration rule as

$$\theta_{m+1} = \theta_m + \gamma_m \nabla\psi(\theta_m) + \varepsilon_m,$$

where

$$\varepsilon_m = \gamma_m(F_m(\theta_m) - \nabla\psi(\theta_m)).$$

In this form, the algorithm can be interpreted as a gradient algorithm with a stochastic noise term. We make following assumption on the step size parameters.

Assumption 8 (Step Size 1) *The step sizes γ_m are deterministic, nonnegative and satisfy*

$$\sum_{m=1}^{\infty} \gamma_m = \infty, \qquad \sum_{m=1}^{\infty} \gamma_m^2 < \infty.$$

Assumption 8 is a standard assumption for gradient methods with a stochastic noise term, and is satisfied if, for example, $\gamma_m = \frac{1}{m}$.

To derive an algorithm which updates the parameter vector at every time step, we express the estimate $F_m(\theta)$ in the form

$$F_m(\theta) = \sum_{k=t_m}^{t_{m+1}-1} R_k(\theta),$$

where t_m is the time of the mth visit to state i^*, and $R_k(\theta)$ is a term that becomes available at the k-th time-step. We define the term $R_k(\theta)$ more precisely in the following chapters. We then use the following rule to update θ,

$$\theta_{k+1} = \theta_k + \gamma_k R_k(\theta_k).$$

For this algorithm, we impose an additional assumption on the step size parameters.

Assumption 9 (Step Size 2) *The step sizes γ_k are non-increasing. Furthermore, there exists a positive integer p and a positive scalar A such that*

$$\sum_{k=n}^{n+t} (\gamma_n - \gamma_k) \leq A t^p \gamma_n^2,$$

for all positive integers n and t.

Again, Assumption 9 holds for $\gamma_k = \frac{1}{k}$.

In the following, we will analysis the behavior of these simulation-based algorithm for optimizing the weighted reward-to-go, and the average reward, respectively. In particular, we show that for the case where the decision maker uses an estimate of the gradient to update the parameter vector θ, the convergence result of the idealized gradient algorithm still holds (with probability 1). That is, for the given performance measure $\psi(\theta)$, the sequence $\psi(\theta_k)$ converges and $\lim_{k\to\infty} \nabla\psi(\theta_k) = 0$.

Chapter 4

Optimizing the Weighted Reward-to-Go

In this chapter, we consider Markov reward processes which depend on a parameter $\theta \in \Re^K$, and for which Assumption 6 (MRP Termination) holds, i.e., there exists a state $i^* \in S$ such that, for every parameter $\theta \in \Re^k$, i^* is a reward-free terminal state which is eventually reached from every state $i \in S$. Furthermore, we consider the case where a decision maker tunes the parameter θ so as to optimize the weighted reward-to-go

$$\chi(\theta) = \sum_{i \in S} \bar{\pi}_i J_i(\theta),$$

where $\bar{\pi}$ is a given probability distribution on the state space S, and where $J_i(\theta)$ is the reward-to-go of state i, i.e., we have

$$J_i(\theta) = E_\theta \left[\sum_{k=0}^{T-1} g_{i_k}(\theta) \mid i_0 = i \right], \tag{4.1}$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that the state i^* is visited.

The remainder of the section is structured as follows. In Section 4.1, we derive an expression for the derivative $\nabla\chi(\theta)$ of the weighted reward-to-go, and propose a gradient algorithm for tuning θ so as to improve $\chi(\theta)$. For problems involving a large state space it is not feasible to compute $\nabla\chi(\theta)$ exactly. As an alternative approach, we consider in Section 4.2 the situation, where the decision maker employs simulation to obtain an estimate of the derivative $\nabla\chi(\theta)$, and updates the parameter vector θ along an approximate gradient direction. In particular, given a trajectory $(i_0 = i, i_1, i_2, ...)$ of the Markov chain with transition probabilities $P_{ij}(\theta)$ which starts at the state i, the expression

$$\tilde{J}_i(\theta) = \sum_{k=0}^{T-1} g_{i_k}(\theta) \tag{4.2}$$

yields an unbiased estimate $J_i(\theta)$. Here, $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that the state i^* is visited. By accumulating such estimates of the reward-to-go, we then produce an unbiased estimate of the gradient of the weighted reward-to-go. In Section 4.3, we use this estimate to formulate a simulation based gradient method which updates the parameter vector at visits to the state i^*. If in Eq.(4.2) the time T until we visit the terminal state i^* is large, as it is the case in many problems of interest, then the variance of the estimate $\tilde{J}_i(\theta)$ can become quite large. As a consequence, the variance of the updates in the above proposed algorithm may be quite large. Using updates with a large variance is undesirable, as this leads to slow convergence. To overcome this difficulty, we consider in Section 4.4 two approaches to reduce the variance in the estimate of the reward-to-go, which we then use in Section 4.5 to produce a modified estimate of the gradient $\nabla\chi(\theta)$. Using this modified estimate of $\nabla\chi(\theta)$, we then formulate in Section 4.6 an alternative algorithm for tuning θ. Finally, in Section 4.7, we discuss an efficient way to compute estimates of the gradient $\nabla\chi(\theta)$, which allows us to derive an algorithm that updates the parameter vector at every time step.

4.1 The Gradient of $\chi(\theta)$

In this section we derive an expression for the the gradient of the weighted reward-to-go

$$\chi(\theta) = \sum_{i\in S} \bar{\pi}_i J_i(\theta)$$

with respect to θ, and propose a gradient algorithm for tuning θ so as to improve $\chi(\theta)$.

We start out by defining a new Markov reward process on the state space S, which will play an important role in our analysis. In particular, using the probability distribution $\bar{\pi}$ on the state space S, we define a new Markov Reward Process with transition probabilities

$$P_{\bar{\pi},ij} = \begin{cases} P_{ij}(\theta), & i \neq i^*, \\ \bar{\pi}_j, & i = i^*, \end{cases}$$

and one-stage rewards

$$g_{\bar{\pi},i}(\theta) = g_i(\theta).$$

Note that this new Markov reward process differs from the original one only in the probabilities for transitions from state i^* to states $j \in S$, for which we now have $P_{\bar{\pi},i^*j} = \bar{\pi}_j$.

In the following, we use the notation $P_{\bar{\pi},\theta}(\cdot)$, to denote the probability distribution induced by the Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$. Accordingly, we use $E_{\bar{\pi},\theta}[\cdot]$, and $\mathrm{Var}_{\bar{\pi},\theta}(\cdot)$, to indicate that the expectation, and the variance, respectively, are taken with respect to the probability distribution of the Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$.

Using the transition probabilities $P_{\bar{\pi},ij}(\theta)$, we make the following definitions. Let $P_{\bar{\pi}}(\theta)$ be the transition matrix with entries $P_{\bar{\pi},ij}(\theta)$, let $\mathcal{P}_{\bar{\pi}}$ be the set of all such transition matrices, and let $\overline{\mathcal{P}}_{\bar{\pi}}$ be its closure. We have the following result.

Lemma 5 *Let Assumption 5 (MRP Parameterization), and 6 (MRP Termination) hold. Then, every element $P_{\bar{\pi}} \in \overline{\mathcal{P}}_{\bar{\pi}}$ is a stochastic matrix. Furthermore, for every $P_{\bar{\pi}} \in \overline{\mathcal{P}}_{\bar{\pi}}$ and every state $i \in S$, we have*

$$\sum_{n=1}^{N} P^n_{\bar{\pi},ii^*} > 0,$$

where N is the number of states in the state space S.

This lemma follows directly from the definitions and Assumption 5 (MRP Parameterization) and 6 (MRP Termination). We omit a detailed proof here.

For the Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$, let $\pi_{\bar{\pi},i}(\theta)$ be the steady state probability distribution of being in state $i \in S$, let $\lambda_{\bar{\pi}}(\theta) = \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) g_{\bar{\pi},i}(\theta)$ be the average reward, and let $E_{\bar{\pi},\theta}[T]$ be the mean recurrence time, i.e., we have

$$E_{\bar{\pi},\theta}[T] = E_{\bar{\pi},\theta}\left[T \mid i_0 = i^*\right],$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. We then have the following relationship between $\chi(\theta)$, $E_{\bar{\pi},\theta}[T]$ and $\lambda_{\bar{\pi}}(\theta)$.

Lemma 6 *Let $\chi(\theta)$, $E_{\bar{\pi},\theta}[T]$ and $\lambda_{\bar{\pi}}(\theta)$ be defined as above. Then we have*

$$\chi(\theta) = E_{\bar{\pi},\theta}[T]\lambda_{\bar{\pi}}(\theta).$$

Proof: Using renewal theory [Gal95], it follows that we have

$$\lambda_{\bar{\pi}}(\theta) = \frac{R(\theta)}{E_{\bar{\pi},\theta}[T]},$$

where

$$R(\theta) = E_{\bar{\pi},\theta}\left[\sum_{k=0}^{T-1} g_{\bar{\pi},i_k}(\theta) \mid i_0 = i^*\right],$$

with $T = \min\{k > 0 \mid i_k = i^*\}$ being the first future time that state i^* is visited. Since, for every $\theta \in \Re^K$, the one-stage reward $g_{i^*}(\theta)$ is equal to 0, note that we have

$$R(\theta) = \sum_{i \in S} \bar{\pi}_i J_i(\theta) = \chi(\theta),$$

where $J_i(\theta)$ is the reward-to-go associated with the Markov reward process with transition probabilities $P_{ij}(\theta)$, and one-stage rewards $g_i(\theta)$. □

Using the lemma above, we can derive the following proposition which gives an expression for the gradient of the weighted reward-to-go $\chi(\theta)$ with respect to θ.

Proposition 1 *Let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold. Then,*

$$\nabla\chi(\theta) = E_{\bar{\pi},\theta}[T] \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta) J_j(\theta)\right).$$

Proof: Let θ and θ' be two parameter vectors in $\Re^K$. Using matrix notation, we have

$$J(\theta') = g(\theta') + P(\theta')J(\theta'),$$

and

$$J(\theta) = g(\theta) + (P(\theta) - P(\theta'))J(\theta) + P(\theta')J(\theta), \tag{4.3}$$

where $J(\theta)$ and $J(\theta')$ are the reward-to-go vectors for the Markov reward processes with transition matrices $P(\theta)$ and $P(\theta')$, respectively, and one-stage reward vectors $g(\theta)$ and $g(\theta')$, respectively. Note that Eq.(4.3) implies that $J_i(\theta)$ can be interpreted as the reward-to-go of state i in the Markov reward process with transition probabilities $P_{\bar{\pi},ij}(\theta')$, and one-stage rewards $g_i'(\theta) = g_i(\theta) + \sum_{j\in S}(P_{ij}(\theta) - P_{ij}(\theta'))J_j(\theta)$. Combining this observation with Lemma 6, we obtain

$$\begin{aligned}
\chi(\theta') - \chi(\theta) &= E_{\bar{\pi},\theta'}[T]\sum_{i\in S}\pi_{\bar{\pi},i}(\theta')g_i(\theta') - \\
&\quad E_{\bar{\pi},\theta'}[T]\sum_{i\in S}\pi_{\bar{\pi},i}(\theta')\left(g_i(\theta) + \sum_{j\in S}(P_{\bar{\pi}_i,ij}(\theta) - P_{\bar{\pi}_i,ij}(\theta')J_j(\theta))\right) \\
&= E_{\bar{\pi},\theta'}[T]\sum_{i\in S}\pi_{\bar{\pi},i}(\theta')\left((g_i(\theta') - g_i(\theta)) + \sum_{j\in S}(P_{\bar{\pi},ij}(\theta') - P_{\bar{\pi},ij}(\theta))J_j(\theta)\right).
\end{aligned}$$

By Lemma 4 and 5, $E_{\bar{\pi},\theta}[T]$ and $\pi_{\bar{\pi}}(\theta)$ are continuous in θ, and the result follows. □

Let us illustrate Proposition 1 with the following example which is due to [Tsi98].

Example 3 *Consider the Markov reward process on the state space* $S = \{1,2\}$, *with transition probabilities*

$$\begin{aligned}
P_{11}(\theta) &= \frac{\exp(\theta)}{2(1+\exp(\theta))}, \\
P_{12}(\theta) &= 1 - P_{11}(\theta), \\
P_{22}(\theta) &= 1,
\end{aligned}$$

and one-stage rewards

$$g_1(\theta) = 1 \quad \text{and} \quad g_2(\theta) = 0.$$

The structure of this Markov reward process is given in Figure 4.1. Note that, for this example, Assumption 5 (MRP Parameterization) and 6 (MRP Termination) hold, with state 2 serving as the reward-free terminal state i^*. *Let the probability distribution* $\bar{\pi}$ *on* S *be given by*

$$\bar{\pi}_1 = 1 \quad \text{and } \bar{\pi}_2 = 0.$$

We then have the following relations,

$$\begin{aligned}
\chi(\theta) = J_1(\theta) &= \frac{1}{P_{12}(\theta)} \\
E_{\bar{\pi},\theta}[T] &= 1 + 1/P_{12}(\theta) \\
\pi_{\bar{\pi},1}(\theta) &= \frac{1/P_{12}(\theta)}{1 + 1/P_{12}(\theta)}.
\end{aligned}$$

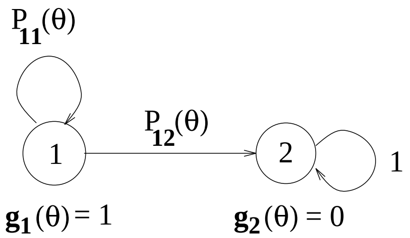

Figure 4.1: Structure of the Markov reward process in Example 3.

It follows that

$$\begin{aligned}
\nabla \chi(\theta) &= \frac{-\nabla P_{12}(\theta)}{(P_{12}(\theta))^2} \\
&= \big(1 + 1/P_{12}(\theta)\big) \left(\frac{1/P_{12}(\theta)}{1 + 1/P_{12}(\theta)} \right) \big(- \nabla P_{12}(\theta) \big) \left(\frac{1}{P_{12}(\theta)} \right) \\
&= E_{\bar{\pi},\theta}[T] \pi_{\bar{\pi},1}(\theta) \nabla P_{11}(\theta) J_1(\theta),
\end{aligned}$$

as predicted by Proposition 1.

4.1.1 An Idealized Gradient Algorithm

Proposition 1 provides an expression for the derivative of $\chi(\theta)$ with respect to the parameter vector θ. Assuming that for every parameter $\theta \in \Re^K$ and every state $i \in S$ we can compute the steady state probability $\pi_{\bar{\pi},i}(\theta)$, and the reward-to-go $J_i(\theta)$, it is then natural to consider the following gradient algorithm for tuning θ. Starting with an initial parameter vector $\theta_0 \in \Re^K$, we update the parameter vector iteratively according to the rule

$$\begin{aligned}
\theta_{k+1} &= \theta_k + \gamma_k \nabla \chi(\theta_k) \\
&= \theta_k + \gamma_k E_{\bar{\pi},\theta_k}[T] \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta) J_j(\theta) \right),
\end{aligned}$$

where γ_k is a step size parameter. We have the following convergence result.

Proposition 2 *Let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold, and let $\theta_0 \in R^k$ be an initial parameter vector. Consider the iteration defined by the rule*

$$\begin{aligned}
\theta_{k+1} &= \theta_k + \gamma_k \nabla \chi(\theta_k) \\
&= \theta_k + \gamma_k E_{\bar{\pi},\theta_k}[T] \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta) J_j(\theta) \right).
\end{aligned}$$

Then, there exists a interval $[a,b]$, *with* $0 < a < b$, *such that if* $\gamma_k \in [a,b]$ *for every* k, *then* $\chi(\theta_k)$ *converges and*

$$\lim_{k\to\infty} \|\nabla\chi(\theta_k)\| = 0.$$

This Proposition follows immediately from the following result from nonlinear programming [Ber95b], and from Lemma 2, which implies that $\chi(\theta)$ is bounded, and that there exists a constant L, such that

$$\|\nabla\chi(\theta) - \nabla\chi(\theta')\| \leq L\|\theta - \theta'\|, \qquad \text{for all } \theta, \theta' \in R^k.$$

Proposition 3 *Let* f *be a bounded function on* $\Re^K$, *and let* (θ_k) *be a sequence of parameter vectors generated by the gradient method* $\theta_{k+1} = \theta_k + \gamma_k \nabla_\theta f(\theta_k)$. *Assume that there exists a constant* $L > 0$ *such that, for all elements* $\theta, \theta' \in R^k$, *we have*

$$\|\nabla f(\theta) - \nabla f(\theta')\| \leq L\|\theta - \theta'\|$$

and that there exists a positive real number ϵ *such that, for all* k, *we have*

$$0 < \epsilon \leq \gamma_k \leq \frac{2-\epsilon}{L}.$$

Then we have that $f(\theta_k)$ *converges and* $\lim_{k\to\infty} \|\nabla f(\theta_k)\| = 0$.

4.2 Estimation of $\nabla\chi(\theta)$

In the previous section, we derived an expression for the gradient of the weighted reward-to-go $\chi(\theta)$ with respect to θ. However, for problems involving a large state space, it is generally impossible to compute this gradient exactly, since this requires to compute, for every state $i \in S$, the steady state probability $\pi_{\bar{\pi},i}(\theta)$ and the reward-to-go $J_i(\theta)$. In this section, we consider an alternative approach, where we employ simulation to estimate the gradient $\nabla\chi(\theta)$.

Throughout this section, we assume that the parameter vector θ is fixed to some value. Let $\bar{\pi}$ be a given probability distribution on the state space S, and let $(i_1, i_2, ...)$ be a sample path of the corresponding Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$. Let t_m be the time of the mth visit at the recurrent state i^*. We refer to the sequence $i_{t_m}, i_{t_m+1}, \ldots, i_{t_{m+1}} - 1$ as the mth renewal cycle.

Note that for a fixed θ, the random variables $T_m = (t_{m+1} - t_m)$, which denotes the length of the mth renewal cycle, are independent and identically distributed, and have a (common) finite mean, which is equal to the mean recurrence time $E_{\bar{\pi},\theta}[T]$.

Our first step is to rewrite the formula for $\nabla\chi(\theta)$ given by Proposition 1 in the form

$$\nabla\chi(\theta) = E_{\bar{\pi},\theta}[T] \sum_{i\in S} \pi_{\bar{\pi},i}(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S_i(\theta)} P_{ij}(\theta) \left(\frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)} J_j(\theta) \right) \right),$$

where the set $S_i(\theta)$ is defined as follows

$$S_i(\theta) = \{ j \in S \mid \nabla P_{ij}(\theta) \neq 0 \}.$$

Assumption 5 (MRP Parameterization) implies that for all states $j \in S_i(\theta)$ the transition probability $P_{ij}(\theta)$ is distinct from 0 and the quotient $\frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)}$ is well defined.

The definition (3.5) of $J_j(\theta)$, suggests that if $t_m < n \leq t_{m+1} - 1$, we can use

$$\tilde{J}_{i_n}(\theta) = \sum_{k=n}^{t_{m+1}-1} g_{i_k}(\theta),$$

to estimate $J_j(\theta)$. As we have that $J_{i^*}(\theta) = 0$, we let

$$\tilde{J}_{i_n}(\theta) = 0, \qquad \text{if } n = t_m.$$

By accumulating the above described estimates over a renewal cycle, we are finally led to an estimate of $\nabla\chi(\theta)$ given by

$$F_m(\theta) = \sum_{n=t_m}^{t_{m+1}-1} \left(\tilde{J}_{i_n}(\theta) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \nabla g_{i_n}(\theta) \right).$$

The denominator $P_{i_{n-1}i_n}(\theta)$ is always positive, since only transitions that have positive probability will be observed. Note that the random variables $F_m(\theta)$ are independent and identically distributed for different values of m, because the transitions during distinct renewal cycles are independent. We define $f_{\bar{\pi}}(\theta)$ to be the expected value of $F_m(\theta)$, namely,

$$f_{\bar{\pi}}(\theta) = E_{\bar{\pi},\theta}[F_m(\theta)].$$

The following proposition confirms that the expectation of $F_m(\theta)$ is an unbiased estimate of $\nabla\chi(\theta)$.

Proposition 4 *We have*

$$f_{\bar{\pi}}(\theta) = \nabla\chi(\theta).$$

Proof: We have

$$\tilde{J}_{i_n}(\theta) = \sum_{k=n}^{t_{m+1}-1} g_{i_k}(\theta),$$

and

$$F_m(\theta) = \sum_{n=t_m+1}^{t_{m+1}-1} \tilde{J}_{i_n}(\theta) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \sum_{n=t_m}^{t_{m+1}-1} \nabla g_{i_n}(\theta).$$

We consider separately the expectations of the two sums above. We consider first the second sum. The expected sum of rewards over a renewal cycle is equal to the steady-state expected reward times the expected length of the renewal cycle. Therefore, the expectation of the second sum is

$$E_{\bar{\pi},\theta}\left[\sum_{n=t_m}^{t_{m+1}-1} \nabla g_{i_n}(\theta)\right] = E_{\bar{\pi},\theta}[T] \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) \nabla g_i(\theta). \tag{4.4}$$

We now focus on the expectation of the first sum. For $n = t_{m+1}, \ldots, t_{m+1} - 1$, let

$$\Delta_n = (\tilde{J}_{i_n}(\theta) - J_{i_n}(\theta)) \frac{\nabla P_{i_{n-1} i_n}(\theta)}{P_{i_{n-1} i_n}(\theta)}.$$

Let $\mathcal{F}_n = \{i_0, \ldots, i_n\}$ stand for the history of the process up to time n. Using the definition (3.5) of $J_{i_n}(\theta)$, we have

$$E_{\bar{\pi},\theta}\left[\tilde{J}_{i_n}(\theta) \mid \mathcal{F}_n\right] = J_{i_n}(\theta).$$

It follows that $E_{\bar{\pi},\theta}[\Delta_n \mid \mathcal{F}_n] = 0$.

Let $\chi_n = 1$ if $n < t_{m+1}$, and $\chi_n = 0$, otherwise. For any $n > t_m$, we have

$$E_{\bar{\pi},\theta}[\chi_n \Delta_n \mid \mathcal{F}_{t_m}] = E_{\bar{\pi},\theta}[E_{\bar{\pi},\theta}[\chi_n \Delta_n \mid \mathcal{F}_n] \mid \mathcal{F}_{t_m}] = E_{\bar{\pi},\theta}[\chi_n E_{\bar{\pi},\theta}[\Delta_n \mid \mathcal{F}_n] \mid \mathcal{F}_{t_m}] = 0.$$

We then have

$$\begin{aligned} E_{\bar{\pi},\theta}\left[\sum_{n=t_m+1}^{t_{m+1}-1} \Delta_n \mid \mathcal{F}_{t_m}\right] &= E_{\bar{\pi},\theta}\left[\sum_{n=t_m+1}^{\infty} \chi_n \Delta_n \mid \mathcal{F}_{t_m}\right] \\ &= \sum_{n=t_m+1}^{\infty} E_{\bar{\pi},\theta}[\chi_n \Delta_n \mid \mathcal{F}_{t_m}] \\ &= 0. \end{aligned}$$

(The interchange of the summation and the expectation can be justified by appealing to the dominated convergence theorem.)

We therefore have

$$E_{\bar{\pi},\theta}\left[\sum_{n=t_m+1}^{t_{m+1}-1} \tilde{J}_{i_n}(\theta) \frac{\nabla P_{i_{n-1} i_n}(\theta)}{P_{i_{n-1} i_n}(\theta)}\right] = E_{\bar{\pi},\theta}\left[\sum_{n=t_m+1}^{t_{m+1}-1} J_{i_n}(\theta) \frac{\nabla P_{i_{n-1} i_n}(\theta)}{P_{i_{n-1} i_n}(\theta)}\right].$$

The right-hand side can be viewed as the total reward over a renewal cycle of a Markov reward process with transition probabilities $P_{\bar{\pi},ij}(\theta)$, where the reward associated with a transition from i to j is equal to $J_j(\theta) \nabla P_{ij}(\theta) / P_{ij}(\theta)$. Recalling that, for $i \neq i^*$, any particular transition from state i to j has steady-state probability $\pi_{\bar{\pi},i}(\theta) P_{ij}(\theta)$, and that $\nabla P_{\bar{\pi},i^* j}(\theta) = 0$, we obtain

$$E_{\bar{\pi},\theta}\left[\sum_{n=t_m+1}^{t_{m+1}-1} \tilde{J}_{i_n}(\theta) \frac{\nabla P_{i_{n-1} i_n}(\theta)}{P_{i_{n-1} i_n}(\theta)}\right] = E_{\bar{\pi},\theta}[T] \sum_{i \in S} \sum_{j \in S_i(\theta)} \pi_{\bar{\pi},i}(\theta) P_{ij}(\theta) \left(\frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)} J_j(\theta)\right). \quad (4.5)$$

Here, we also used the fact that, by Assumption 5 (MRP Parameterization), the transition probability $P_{ij}(\theta)$ is equal to 0 for all states j that are not contained in the set $S_i(\theta)$. By combining Eqs.(4.4) and (4.5), and comparing with the formula for $\nabla \chi(\theta)$, we see that the desired result has been proved. □

4.3 An Algorithm that Updates at Visits to the Terminal State i^*

We now use the estimate of the gradient direction provided by Proposition 4 to propose a simulation-based algorithm that performs updates at visits to the state i^*. We use the variable m to index the times when the recurrent state i^* is visited, and the corresponding updates. The form of the algorithm is the following. At the time t_m that state i^* is visited for the mth time, we have available a current vector θ_m. We then simulate the process according to the transition probabilities $\bar{P}_{\bar{\pi},ij}(\theta_m)$ until the next time t_{m+1} that i^* is visited and update according to

$$\begin{aligned} \theta_{m+1} &= \theta_m + \gamma_m F_m(\theta_m) \\ &= \theta_m + \gamma_m \sum_{n=t_m}^{t_{m+1}-1} \left(\tilde{J}_{i_n}(\theta) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \nabla g_{i_n}(\theta) \right) \end{aligned}$$

where γ_m is a positive step size parameter. We can rewrite this iteration as as

$$\theta_{m+1} = \theta_m + \gamma_m \nabla\chi(\theta_m) + \varepsilon_m,$$

where

$$\varepsilon_m = \gamma_m(F_m(\theta_m) - \nabla\chi(\theta_m)).$$

Proposition 4 the implies that we have

$$E_\theta[\varepsilon_m] = E_\theta[\gamma_m(F_m(\theta_m) - \nabla\chi(\theta_m))] = 0.$$

Therefore, the algorithm we propose here can be interpreted as a gradient algorithm with a stochastic error term ε_m that is a zero mean random vector. It is then not surprising that we have the following convergence result.

Proposition 5 *Let Assumptions 5 (MRP Parameterization), 6 (MRP Termination) and 8 (Step Size 1) hold, and let (θ_m) be the sequence of parameter vectors generated by the above described algorithm. Then, with probability 1, $\chi(\theta_m)$ converges and*

$$\lim_{m\to\infty} \nabla\chi(\theta_m) = 0.$$

Proposition 5 can be proven by using standard arguments of stochastic approximation. We provide a proof in Appendix B.1.

For systems involving a large state space, the interval between visits to the state i^* can be large. As a consequence, for the above proposed algorithm we have that

1. the parameter vector θ gets updated infrequently.

2. the estimate $F_m(\theta)$ can have a large variance.

In the following sections, we will address these two issues. In particular, we will propose two approaches to reduce the variance in the estimate $F_m(\theta)$, and derive an algorithm which updates the parameter vector at each time step.

4.4 Reducing the Variance

In this section, we consider two approaches to reduce the variance in the updates of the algorithm proposed in the previous section. In the first approach, we replace the estimate $\tilde{J}_i(\theta)$ in the approximate gradient direction $F_m(\theta)$ by

$$\tilde{J}_{S^*,i}(\theta) = \sum_{k=0}^{T-1} g_{i_k}(\theta), \tag{4.6}$$

where S^* is a subset of the state space S containing i^*, and $T = \min\{k > 0 \mid i_k \in S^*\}$ is the first future time a state in the set S^* is visited. In the second approach, we use an estimate of the reward-to-go $J_i(\theta)$ of state i given by

$$\tilde{J}_{\alpha,i}(\theta) = \sum_{k=0}^{T-1} \alpha^k g_{i_k}(\theta), \tag{4.7}$$

where α, $0 \leq \alpha < 1$, is a discount factor, and where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time the state i^* is visited.

However, in both of the aforementioned approaches to reduce the variance, we introduce a bias into the estimate of the reward-to-go $J_i(\theta)$. As a consequence, when we use $\tilde{J}_{S^*,i}(\theta)$, or $\tilde{J}_{\alpha,i}(\theta)$, to produce an estimate of the gradient $\nabla\chi(\theta)$, then we are likely to introduce a bias into this estimate of the gradient. We will address this issues in Section 4.5, where we characterize this bias and identify situations for which the bias tends to be small.

In the remainder of this section, we derive a bound on the variance of the estimates of the reward-to-go $J_i(\theta)$ given by the Eqs.(4.6), and (4.7).

4.4.1 The Estimate $\tilde{J}_{S^*,i}(\theta)$

As mentioned above, if the time T until we visit the terminal state i^* is large, then the variance of the estimate $\tilde{J}_i(\theta)$ of the previous section can become quite large. For this reason, it may be desirable to reduce the time T by choosing

$$T = \min\{k > 0 \mid i_k \in S^*\},$$

where S^* is a subset of S containing i^*. In particular, given a set $S^* \subset S$ containing i^*, and a trajectory $(i_0 = i, i_1, ...)$ of the Markov chain with transition probabilities $P_{ij}(\theta)$ which starts at state i, we then estimate the reward-to-go $J_i(\theta)$ of state i by

$$\tilde{J}_{S^*,i}(\theta) = \sum_{k=0}^{T-1} g_{i_k}(\theta),$$

where $T = \min\{k > 0 \mid i_k \in S^*\}$ is the first future time a state in the set S^* is visited. To illustrate this idea, let us consider the following example.

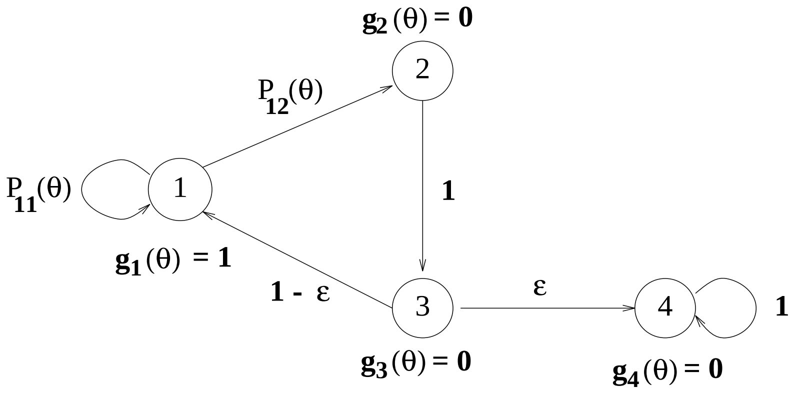

Figure 4.2: Structure of the Markov reward process in Example 4.

Example 4 *Let ϵ be a scalar in $(0, 1]$, and consider the Markov reward process on the state space $S = \{1, 2, 3, 4\}$, with transition probabilities*

$$
\begin{aligned}
P_{11}(\theta) &= \frac{\exp(\theta)}{2(1 + \exp(\theta))}, \\
P_{12}(\theta) &= 1 - P_{11}(\theta), \\
P_{34}(\theta) &= \epsilon, \qquad P_{31}(\theta) = 1 - \epsilon, \\
P_{23}(\theta) &= P_{44}(\theta) = 1,
\end{aligned}
$$

and one-stage rewards

$$g_1(\theta) = 1, \quad \textit{and} \quad g_2(\theta) = g_3(\theta) = g_4(\theta) = 0.$$

The structure of this Markov reward process is given in Figure 4.2. Note that, for this example, Assumption 5 (MRP Parameterization) and 6 (MRP Termination) hold, with state 4 serving as the reward-free terminal state i^*. Let the set S^* be equal to $\{3, 4\}$, and let $(i_0 = 1, i_1, i_2, ...)$ be a trajectory of the Markov chain associated with the parameter θ, which starts at the state 1*. Furthermore, let us consider the estimate*

$$\tilde{J}_{S^*,1}(\theta) = \sum_{k=0}^{T-1} g_{i_k}(\theta)$$

of the reward-to-go $J_1(\theta)$, where $T = \min\{k > 0 \mid i_k \in S^*\}$ is the first future time a state in the set S^* is visited. Note that, in this case, $\tilde{J}_{S^*,1}(\theta)$ has the same distribution as the random variable X_θ given by

$$P(X_\theta = x) = (1 - P_{12}(\theta))^{x-1} P_{12}(\theta), \quad x = 1, 2, ... \quad .$$

It follows that

$$Var_\theta(\tilde{J}_{S^*,1}(\theta)) = Var(X_\theta).$$

Next, let us consider the the estimate

$$\tilde{J}_1(\theta) = \sum_{k=0}^{T-1} g_{i_k}(\theta)$$

of the reward-to-go $J_1(\theta)$, where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time the state i^* is visited. Note that in this case, the estimate $\tilde{J}_1(\theta)$ has the same distribution as the random variable

$$Y_\theta = \sum_{n=1}^{N} X_{\theta,n} \quad ,$$

where $(X_{\theta,n})$ is a sequence of IID random variables with the same distribution as the random variable X_θ above, e.g.

$$P(X_{\theta,n} = x) = (1 - P_{12}(\theta))^{x-1} P_{12}(\theta), \quad x = 1, 2, \dots \quad ,$$

and where N is a random variable, independent of $(X_{\theta,n})$, given by

$$P(N = n) = (1 - \epsilon)^{n-1}\epsilon, \quad n = 1, 2, \dots \quad .$$

It follows that

$$Var_\theta(\tilde{J}_1(\theta)) = Var(Y_\theta) = E[N]\, Var(X_\theta) + E[X_\theta^2]\, Var(N).$$

Therefore, using $\tilde{J}_{S^*,1}(\theta)$, instead of $\tilde{J}_1(\theta)$, to estimate $J_1(\theta)$, reduces the variance by the factor

$$\frac{Var_\theta(\tilde{J}_1(\theta))}{Var_\theta(\tilde{J}_{S^*,1}(\theta))} = \frac{E[N]\, Var(X_\theta) + E[X_\theta^2]\, Var(N)}{Var(X_\theta)} \geq E[N] = \frac{1}{\epsilon},$$

which becomes large when ϵ becomes small.

Example 4 illustrates that the estimate $\tilde{J}_{S^*,i}(\theta)$ can have a significantly lower variance than the estimate $\tilde{J}_i(\theta)$. However, one may suspect that this reduction in the variance will translates into a large bias in the corresponding estimate of the gradient $\nabla\chi(\theta)$. We will revisit this example in the Section 4.5 and show that this is not necessarily the case.

However, while in Example 4 the estimate $\tilde{J}_{S^*,i}(\theta)$ has a lower variance than the estimate $\tilde{J}_i(\theta)$, the opposite can also be true.

Example 5 *Consider the Markov reward process on the state space $S = \{1, 2, 3, 4, 5, 6\}$, with transition probabilities*

$$\begin{aligned} P_{12}(\theta) &= \frac{\exp(\theta)}{1 + \exp(\theta)}, \\ P_{13}(\theta) &= 1 - P_{12}(\theta), \\ P_{24}(\theta) &= P_{46}(\theta) = P_{35}(\theta) = P_{56}(\theta) = 1, \end{aligned}$$

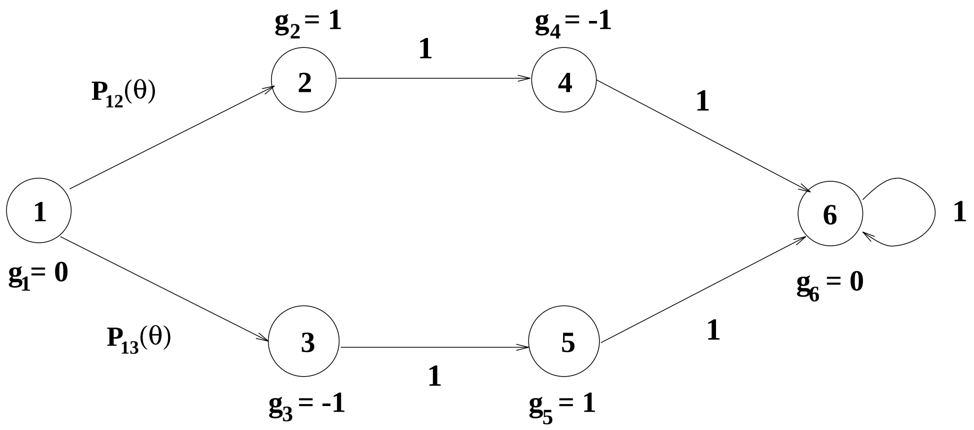

Figure 4.3: Structure of Markov reward process in Example 5.

and one-stage rewards

$$g_1(\theta) = g_6(\theta) = 0, \qquad g_2(\theta) = g_5(\theta) = 1, \qquad g_3(\theta) = g_4(\theta) = -1.$$

The structure of this Markov reward process is given in Figure 4.3. Note that for this example Assumption 5 (MRP Parameterization) and 6 (MRP Termination) hold, with state 6 *serving as the reward-free terminal state* i^*. *Let the set* S^* *be equal to* $\{4, 5, 6\}$, *and let* $(i_0 = 1, i_1, i_2, ...)$ *be a trajectory of the Markov chain associated with the parameter* θ, *which starts at the state* 1. *Furthermore, let us consider the estimate*

$$\tilde{J}_{S^*,1}(\theta) = \sum_{k=0}^{T-1} g_{i_k}(\theta)$$

of the reward-to-go $J_1(\theta)$, *where* $T = \min\{k > 0 \mid i_k = i^*\}$ *is the first future time a state in the set* S^* *is visited. Note that in this case we have*

$$E_\theta[\tilde{J}_{S^*,1}(\theta)] = 2P_{12}(\theta) - 1,$$

and

$$\begin{aligned} Var_\theta(\tilde{J}_{S^*,1}(\theta)) &= P_{12}(\theta)(2 - 2P_{12}(\theta))^2 + (1 - P_{12}(\theta)(2P_{12}(\theta))^2 \\ &= 4P_{12}(\theta)(1 - P_{12}(\theta)). \end{aligned}$$

Next let us consider the estimate

$$\tilde{J}_1(\theta) = \sum_{k=0}^{T-1} g_{i_k}(\theta)$$

of the reward-to-go $J_1(\theta)$, *where* $T = \min\{k > 0 \mid i_k = i^*\}$ *is the first future time the state* i^* *is visited. Note that in this case, the estimate* $\tilde{J}_1(\theta)$ *is equal to* 0 *for every possible trajectory, and that therefore we have*

$$Var(\tilde{J}_1(\theta)) = 0.$$

Therefore, in this example, the estimate $\tilde{J}_{S^*,i}(\theta)$ *has a higher variance than the estimate* $\tilde{J}_i(\theta)$.

Although, it is not generally true that the estimate $\tilde{J}_{S^*,i}(\theta)$ has a lower variance than the estimate $\tilde{J}_1(\theta)$, we can derive a bound on the variance $\tilde{J}_{S^*,i}(\theta)$, which decreases, as the set S^* becomes larger.

Proposition 6 *Let θ be an element of $\Re^K$, and let S^* be a subset of the state space S containing i^*. Furthermore, let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold. In particular, let $g_{\max}$ be a bound on $|g_i(\theta)|$ and let $\rho(\theta) > 0$ be such that*

$$P_\theta(T \leq N \mid i_0 = i) \geq \rho(\theta), \quad \text{for all } i \in S,$$

where N is the number of states in S, and where $T = \min\{k > 0 \mid i_k \in S^*\}$ is the first future time a state in the set S^* is visited by a trajectory that starts at state i. Then, for every state $i \in S$, we have

$$Var_\theta(\tilde{J}_{S^*,i}(\theta)) \leq g_{\max}^2 \left(N^2 \frac{(2 - \rho(\theta))}{\rho(\theta)^2} - 1 \right).$$

Proof: Note that the following is true,

$$\mathrm{Var}_\theta \left(\sum_{k=0}^{T-1} g_{i_k}(\theta) \mid i_0 = i \right) = \mathrm{Var}_\theta \left(\sum_{k=1}^{T-1} g_{i_k}(\theta) \mid i_0 = i \right),$$

as $g_{i_0}(\theta)$ is equal to one-stage reward $g_i(\theta)$ for every trajectory. Furthermore, we have

$$\sum_{k=1}^{T-1} g_{i_k}(\theta) \leq g_{\max}(T-1),$$

which implies that

$$\begin{aligned}
\mathrm{Var}_\theta(\tilde{J}_{S^*,i}(\theta)) &= \mathrm{Var}_\theta \left(\sum_{k=1}^{T-1} g_{i_k}(\theta) \mid i_0 = i \right) \\
&\leq E_\theta \left[\left(\sum_{k=1}^{T-1} g_{i_k}(\theta) \right)^2 \mid i_0 = i \right] \\
&\leq g_{\max}^2 E_\theta[(T-1)^2 \mid i = i_0] \\
&\leq g_{\max}^2 (E_\theta[T^2 \mid i = i_0]) - 1),
\end{aligned}$$

where the last equality follows as $E_\theta[T \mid i = i_0] \geq 1$.

Furthermore, using the assumption that, for all $i \in S$, we have

$$P_\theta(T \leq N \mid i_0 = i) \geq \rho(\theta) > 0,$$

it follows that

$$\begin{aligned}
E_\theta[T^2 \mid i_0 = i] &\leq \sum_{l=1}^{\infty} (lN)^2 (1 - \rho(\theta))^{l-1} \rho(\theta) \\
&= N^2 \frac{1}{1 - \rho(\theta)} \sum_{l=0}^{\infty} l^2 (1 - \rho(\theta))^l \rho(\theta) \\
&= N^2 \frac{(2 - \rho(\theta))}{\rho(\theta)^2}.
\end{aligned}$$

Combining these two results, we obtain

$$\begin{aligned}
\mathrm{Var}_\theta(\tilde{J}_{S^*,i}(\theta)) &= \mathrm{Var}_\theta\left(\sum_{k=1}^{T-1} g_{i_k}(\theta) \mid i_0 = i\right) \\
&\leq E_\theta\left[\left(\sum_{k=1}^{T-1} g_{i_k}(\theta)\right)^2 \mid i_0 = i\right] \\
&\leq g_{\max}^2(E_\theta[T^2 \mid i = i_0] - 1) \\
&\leq g_{\max}^2(N^2\frac{(2-\rho(\theta))}{\rho(\theta)^2} - 1).
\end{aligned}$$

□

Note that the bound on $\mathrm{Var}_\theta(\tilde{J}_{S^*,i}(\theta))$ in Proposition 6 is decreasing in $\rho(\theta)$. Furthermore, note that $\rho(\theta)$ can be chosen to be nondecreasing in the size of the set S^*. It follows that the larger the set S^*, the smaller tends to be the bound on $\mathrm{Var}_\theta(\tilde{J}_{S^*,i}(\theta))$.

4.4.2 The Estimate $\tilde{J}_{\alpha,i}(\theta)$

In this section, we discuss a second approach to reduce the variance in the estimates of the reward-to-go. In particular, given a trajectory $(i_0 = i, i_1, ...)$ of the Markov chain with transition probabilities $P_{ij}(\theta)$ which starts at state i, we estimate the reward-to-go $J_i(\theta)$ by

$$\tilde{J}_{\alpha,i}(\theta) = \sum_{k=0}^{T-1} \alpha^k g_{i_k}(\theta), \tag{4.8}$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that the state i^* is visited, and where α, $0 \leq \alpha < 1$, is a discount factor. We have the following bound on the variance of the estimate $\tilde{J}_{\alpha,i}$.

Proposition 7 *Let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold. In particular, let $g_{\max}$ be a bound on $|g_i(\theta)|$ and let $\rho(\theta) > 0$ be such that*

$$P_{ii^*}^N(\theta) \geq \rho(\theta), \quad \textit{for all } i \in S,$$

where N is the number of states in S. Then, for every $\theta \in \Re^K$ and every discount factor $\alpha \in [0,1)$, we have

$$Var_\theta(\tilde{J}_{\alpha,i}(\theta)) \quad \leq \quad \frac{g_{\max}^2}{(1-\alpha)^2}\left(1 - \frac{2\alpha^N\rho(\theta)}{1-(1-\rho(\theta))\alpha^N} + \frac{\alpha^{2N}\rho(\theta)}{1-(1-\rho(\theta))\alpha^{2N}}\right).$$

Proof: Note that for $\alpha \in [0,1)$, we have

$$\mathrm{Var}_\theta\left(\sum_{k=0}^{T-1} \alpha^k g_{i_k}(\theta) \mid i_0 = i\right) = \mathrm{Var}_\theta\left(\sum_{k=1}^{T-1} \alpha^k g_{i_k}(\theta) \mid i_0 = i\right),$$

as $g_{i_0}(\theta)$ is equal to $g_i(\theta)$ for every trajectory. Furthermore, we have

$$\sum_{k=1}^{T-1} \alpha^k g_{i_k}(\theta) \leq g_{\max} \frac{1-\alpha^{T-1}}{1-\alpha}.$$

It follows that

$$\begin{aligned}
\mathrm{Var}_\theta(\tilde{J}_{\alpha,i}(\theta)) &= \mathrm{Var}_\theta\left(\sum_{k=1}^{T-1} \alpha^k g_{i_k}(\theta) \mid i_0 = i\right) \\
&\leq E_\theta\left[\left(\sum_{k=1}^{T-1} \alpha^k g_{i_k}(\theta)\right)^2 \mid i_0 = i\right] \\
&\leq g_{\max}^2 \frac{1}{(1-\alpha)^2}(1 - 2E_\theta[\alpha^{T-1} \mid i = i_0] + E_\theta[\alpha^{2(T-1)} \mid i = i_0]).
\end{aligned}$$

Furthermore, using the assumption that, for all states $i \in S$, we have $P_{ii^*}^N(\theta) \geq \rho(\theta) > 0$, it follows that

$$\begin{aligned}
E_\theta[\alpha^T \mid i_0 = i] &\leq \sum_{l=1}^{\infty} \alpha^{lN}(1-\rho(\theta))^{l-1}\rho(\theta) \\
&= \sum_{k=0}^{\infty} \alpha^{(l+1)N}(1-\rho(\theta))^{l}\rho(\theta) \\
&= \frac{\rho(\theta)\alpha^N}{1-(1-\rho(\theta))\alpha^N}.
\end{aligned}$$

Using this bound on $E_\theta[\alpha^T \mid i_0 = i]$, we obtain

$$\begin{aligned}
\mathrm{Var}_\theta(\tilde{J}_{\alpha,i}(\theta)) &= \mathrm{Var}\left(\sum_{k=1}^{T-1} \alpha^k g_{i_k}(\theta) \mid i_0 = i\right) \\
&\leq E_\theta\left[\left(\sum_{k=1}^{T-1} \alpha^k g_{i_k}(\theta)\right)^2 \mid i_0 = i\right] \\
&\leq g_{\max}^2 \frac{1}{(1-\alpha)^2}(1 - 2E_\theta[\alpha^{T-1} \mid i = i_0] + E_\theta[\alpha^{2(T-1)} \mid i = i_0]) \\
&\leq g_{\max}^2 \frac{1}{(1-\alpha)^2}\left(1 - \frac{2\rho(\theta)\alpha^{N-1}}{1-(1-\rho(\theta))\alpha^N} + \frac{\rho(\theta)\alpha^{2(N-1)}}{1-(1-\rho(\theta))\alpha^{2N}}\right).
\end{aligned}$$

$\square$

Note that in Proposition 7, the bound for the variance of the estimate $\tilde{J}_{\alpha,i}(\theta)$ is increasing in α and decreasing in $\rho(\theta)$. Furthermore, we have

$$\lim_{\rho(\theta)\to 1} \mathrm{Var}_\theta(\tilde{J}_{\alpha,i}(\theta) \leq g_{\max}\left(\frac{1-\alpha^N}{1-\alpha}\right)^2,$$

which implies that when N, the number of states in the set S is large, then the variance of $\tilde{J}_{\alpha,i}(\theta)$ tends to be large, unless α is small. Therefore, Proposition 7 suggests that in order to obtain a small variance in the estimate $\tilde{J}_{\alpha,i}(\theta)$, we have to choose a small discount factor α. Again, we will identify in the next section situations, for which one can choose a small α to reduce the variance in the estimate of the reward-to-go, without introducing too large a bias in the corresponding estimate of the gradient $\nabla\chi(\theta)$.

4.5 Modified Estimation of $\nabla\chi(\theta)$

In this section, we use the estimates $\tilde{J}_{S^*,i}(\theta)$ and $\tilde{J}_{\alpha,i}(\theta)$ of the reward-to-go of state i introduced in the previous section to produce an estimate of the gradient $\nabla\chi(\theta)$. However, as mentioned earlier, by using $\tilde{J}_{S^*,i}(\theta)$ and $\tilde{J}_{\alpha,i}(\theta)$ we introduce a bias into the estimate of the gradient. Here, we characterize the resulting bias, and identify situations where this bias is small.

4.5.1 Estimation of $\nabla\chi(\theta)$ Using $\tilde{J}_{S^*,i}(\theta)$

Let S^* be a subset of S containing the state i^* that we use to obtain $\tilde{J}_{S^*,i}(\theta)$, and let the parameter $\theta \in \Re^K$ be fixed to some value. Furthermore, let $\bar{\pi}$ be a given probability distribution on the state space S, and let $(i_1, i_2, ...)$ be a sample path of the Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$. As in Section 4.2, let t_m be the time of the mth visit to the terminal state i^*. Here, we also want to mark the times when a state in the set $S^*\backslash\{i^*\}$ is visited. For the interval $k = t_m, ..., t_{m+1}-1$ associated with the mth visit to the state i^*, let $\kappa(m)$ be the number of times a state in the set $S^*\backslash\{i^*\}$ is visited. Furthermore, let the $t_{m,n}$ be the time when for the n-th time a state in the set S^* is visited in the interval $k = t_m, ..., t_{m+1}-1$, and let $t_{m,0}$ and $t_{m,\kappa(m)+1}$ be equal to t_m and t_{m+1}, respectively. Using these definitions, we estimate $\nabla\chi(\theta)$ by

$$F_{S^*,m}(\theta) = \sum_{k=t_m}^{t_{m+1}-1}\left(\tilde{J}_{S^*,i_k}(\theta)\frac{\nabla P_{i_{k-1}i_k}(\theta)}{P_{i_{k-1}i_k}(\theta)} + \nabla g_{i_k}(\theta)\right), \tag{4.9}$$

where for $t_{m,n} \le k < t_{m,n+1}$, and $k \ne t_m$, we set

$$\tilde{J}_{S^*,i_k}(\theta) = \sum_{l=k}^{t_{m,n+1}-1} g_{i_l}(\theta),$$

and for $k = t_m$ we set $\tilde{J}_{S^*,i_k}(\theta) = 0$.

We have the following result for $f_{\bar{\pi},S^*}(\theta)$, which we define to be the expected value of $F_{S^*,m}(\theta)$, namely,

$$f_{\bar{\pi},S^*}(\theta) = E_{\bar{\pi},\theta}[F_{S^*,m}(\theta)].$$

Proposition 8 *Let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold. Then, we have*

$$f_{\bar{\pi},S^*}(\theta) = E_{\bar{\pi},\theta}[T]\sum_{i\in S}\pi_{\bar{\pi},i}(\theta)\left(\nabla g_i(\theta) + \sum_{j\in S}\nabla P_{ij}(\theta)J_{S^*,j}(\theta)\right),$$

where

$$J_{S^*,i}(\theta) = E_\theta\left[\sum_{k=0}^{T-1} g_{i_k}(\theta) \mid i_0 = i\right],$$

with $T = \min\{k > 0 \mid i_k \in S^*\}$ *being the first future time that the set* S^* *is visited.*

Proposition 8 can be proved using the same argument as the one given to prove Proposition 4 in Section 4.2.

Note that the expression for $f_{\tilde{\pi},S^*}(\theta)$ in Proposition 8 is of the same form as the expression for the gradient

$$\nabla\chi(\theta) = E_{\tilde{\pi},\theta}[T]\sum_{i\in S}\pi_{\tilde{\pi},i}(\theta)\left(\nabla g_i(\theta) + \sum_{j\in S_i}\nabla P_{ij}(\theta)J_j(\theta)\right),$$

except that the exact value of the reward-to-go $J_j(\theta)$ is replaced by the approximation $J_{S^*,j}(\theta)$. In the next proposition we characterize the bias

$$\delta_{S^*}(\theta) = f_{\tilde{\pi},S^*}(\theta) - \nabla\chi(\theta)$$

introduced by this modification. We using the following notation. For every state $i \in S$, let S_i be the set given by

$$S_i = \{j \in S \mid \nabla P_{ij}(\theta) \neq 0, \text{ for some } \theta \in \Re^K\},$$

and let N_i be the number of states in the set S_i. We then define $\bar{N}$ by

$$\bar{N} = \max\{N_i \mid i \in S\},$$

which bounds the number of possible transitions from any state $i \in S$ that depend on θ. In addition, let $\hat{J}_{S^*,i}(\theta)$ be defined by

$$\hat{J}_{S^*,i}(\theta) = J_{S^*,i}(\theta) - J_i(\theta),$$

where $J_{S^*,i}(\theta)$ is defined as in Proposition 8. Note that we can interpret $\hat{J}_{S^*,i}(\theta)$ as the bias in the estimate $\tilde{J}_{S^*,i}(\theta)$ of the reward-to-go $J_i(\theta)$. We then have the following proposition.

Proposition 9 *Let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold, and let* ϵ *be such that, for all states* $i \in S$*, we have*

$$|\hat{J}_{S^*,j}(\theta) - \hat{J}_{S^*,j'}(\theta)| \leq \epsilon, \text{ if } j, j' \in S_i.$$

Furthermore, let $T_{\max}$ *be such that, for all* $\theta \in \Re^K$*, we have*

$$E_{\tilde{\pi},\theta}[T] \leq T_{\max}.$$

Then we have

$$\|\delta_{S^*}(\theta)\| = \|f_{\tilde{\pi},S^*}(\theta) - \nabla\chi(\theta_m)\| \leq T_{\max}\bar{N}C\epsilon,$$

where C *is a bound on* $\|\nabla P_{ij}(\theta)\|$.

Note that by Lemma 4 and 5, there exists a bound $T_{\max}$ on $E_{\bar{\pi},\theta}[T]$.

Proof: For every state $i \in S$, let $j_r(i)$ be a state in S_i which we use as a reference state. By assumption, for all states $i \in S$ and for all state $j \in S_i$, we have

$$|\hat{J}_{S^*,j}(\theta) - \hat{J}_{S^*,j_r(i)}(\theta)| \leq \epsilon.$$

Using the fact that for every $i \in S$, and every $\theta \in \Re^K$, we have

$$\sum_{j \in S} \nabla P_{ij}(\theta) = 0,$$

it follows that

$$\begin{aligned}
\|f_{\bar{\pi},S^*}(\theta) - \nabla\chi(\theta_m)\| &= \|E_{\bar{\pi},\theta}[T] \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) \sum_{j \in S_i} \nabla P_{ij}(\theta)(J_{S^*,j}(\theta) - J_j(\theta))\| \\
&= \|E_{\bar{\pi},\theta}[T] \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) \sum_{j \in S_i} \nabla P_{ij}(\theta)\hat{J}_{S^*,j}(\theta)\| \\
&= \|E_{\bar{\pi},\theta}[T] \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) \sum_{j \in S_i} \nabla P_{ij}(\theta)(\hat{J}_{S^*,j}(\theta) - \hat{J}_{S^*,j_r(i)}(\theta))\| \\
&\leq E_{\bar{\pi},\theta}[T] \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) \sum_{j \in S_i} \|\nabla P_{ij}(\theta)\|\epsilon \\
&\leq T_{\max}\bar{N}C\epsilon.
\end{aligned}$$

□

This proposition provides insight for choosing the set S^* so that the resulting bias in the estimates $F_{S^*,m}(\theta)$ of the gradient of $\chi(\theta)$ is small. In particular, S^* should be chosen such that, for all states $i \in S$, the difference $|\hat{J}_{S^*,j}(\theta) - \hat{J}_{S^*,j'}(\theta)|$ is small for all states $j, j' \in S_i$. To illustrate this result, we consider the following example.

Example 6 *Let ϵ be a scalar in $(0,1]$, and consider the Markov reward process on the state space $S = \{1,2,3\}$, with transition probabilities*

$$\begin{aligned}
&P_{11}(\theta) = \frac{1}{2}\frac{\exp(\theta)}{1+\exp(\theta)}, \\
&P_{12}(\theta) = 1 - P_{11}(\theta), \\
&P_{34}(\theta) = \epsilon, \qquad P_{31}(\theta) = 1 - \epsilon, \\
&P_{23}(\theta) = P_{44}(\theta) = 1,
\end{aligned}$$

and one-stage rewards

$$g_1(\theta) = 1, \quad \text{and} \quad g_2(\theta) = g_3(\theta) = g_4(\theta) = 0.$$

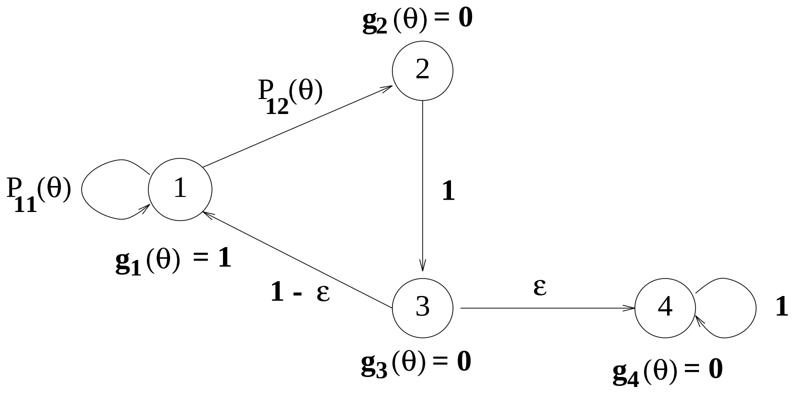

Figure 4.4: Structure of the Markov reward process in Example 6.

The structure of this Markov reward process is given in Figure 4.4. Let the set S^* be equal to $\{3, 4\}$. We have shown in Example 4 that, in this case, the estimate $\tilde{J}_{S^*,1}(\theta)$ has a smaller variance than the estimate $\tilde{J}_1(\theta)$. Furthermore, note that in this case the set

$$S_1 = \{j \in S \mid \nabla P_{1j}(\theta) \neq 0 \textit{ for some } \theta \in \Re\}$$

is equal to $\{1, 2\}$ and that

$$\hat{J}_{S^*,j}(\theta) = J_2(\theta), \quad \textit{for } j \in S_1.$$

Therefore, by Proposition 9, using $\tilde{J}_{S^*,i}(\theta)$ reduces the variance in the estimate of the reward-to-go $J_1(\theta)$ without introducing a bias into the estimate of the gradient $\nabla\chi(\theta)$.

4.5.2 Estimation of $\nabla\chi(\theta)$ Using $\tilde{J}_{\alpha,i}(\theta)$

In this section, we produce an estimate of the gradient $\nabla\chi(\theta)$, by accumulating estimates $\tilde{J}_{\alpha,i}(\theta)$ given by Eq.(4.7) along a trajectory of the Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$.

Let scalar α be in $[0, 1)$, and let the parameter $\theta \in \Re^K$ be fixed to some value. Furthermore, let $\bar{\pi}$ be a given probability distribution over the state space S, and let $(i_1, i_2, ...)$ be a sample path of the Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$. As in Section 4.2, let t_m be the time of the mth visit at the recurrent state i^*. We then estimate the gradient $\nabla\chi(\theta)$ by

$$F_{\alpha,m}(\theta) = \sum_{n=t_m}^{t_{m+1}-1} \left(\tilde{J}_{\alpha,i_n}(\theta) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \nabla g_{i_n}(\theta) \right). \tag{4.10}$$

where for $t_m < n \leq t_{m+1}$, we set

$$\tilde{J}_{\alpha,i_n}(\theta) = \sum_{k=n}^{t_{m+1}-1} \alpha^{k-n} g_{i_k}(\theta),$$

and for $n = t_m$, we set $\tilde{J}_{\alpha,i_n}(\theta) = 0$.

We have the following result for $f_{\bar{\pi},\alpha}(\theta)$, which we define to be the expected value of $F_{\alpha,m}(\theta)$, namely,

$$f_{\bar{\pi},\alpha}(\theta) = E_{\bar{\pi},\theta}[F_{\alpha,m}(\theta)].$$

Proposition 10 *Let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold. Then, we have*

$$f_{\bar{\pi},\alpha}(\theta) = E_{\bar{\pi},\theta}[T] \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S_i} \nabla P_{ij}(\theta) J_{\alpha,j}(\theta) \right),$$

where

$$J_{\alpha,i}(\theta) = E_\theta \left[\sum_{k=0}^{T-1} \alpha^k g_{i_k}(\theta) \right],$$

with $T = \min\{k > 0 \mid i_k = i^*\}$ *being the first future time that the state* i^* *is visited.*

Again, Proposition 10 can be proved by the same argument as the one given to prove Proposition 4 in Section 4.2.

Next, we characterize the bias

$$\delta_\alpha(\theta) = f_{\bar{\pi},\alpha}(\theta) - \nabla \chi(\theta)$$

in the estimate $F_{\alpha,m}(\theta)$ of the gradient. In order to do that, we use the notion of a mixing constant on the set $S \backslash \{i^*\}$. In particular, the scalar $\beta(\theta)$, $0 \le \beta(\theta) < 1$, is a mixing constant on the set $S \backslash \{i^*\}$ for the Markov chain with transition probabilities $P_{ij}(\theta)$, if there exists a constant $A(\theta)$, such that for all states $i, i' \in S \backslash \{i^*\}$, and all $n \ge 1$, we have

$$\sum_{j \in S} |P^n_{ij}(\theta) - P^n_{i'j}(\theta)| \le A(\theta)\beta(\theta)^n.$$

The mixing constant β is a measure on "how fast" two trajectories, which start at the states $i \in S \backslash \{i^*\}$ and $i' \in S \backslash \{i^*\}$, respectively, are going to meet.

Proposition 11 *Let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold, and let* $T_{\max}$ *be such that, for all* $\theta \in \Re^K$*, we have*

$$E_{\bar{\pi},\theta}[T] \le T_{\max}.$$

Furthermore, let the constants β*,* $0 \le \beta < 1$*, and* A*, be such that for all states* $i, i' \in S \backslash \{i^*\}$*, for all transition matrices* $P \in \bar{\mathcal{P}}$ *and all* $n \ge 1$*, we have*

$$\sum_{j \in S} |P^n_{ij} - P^n_{i'j}| \le A\beta^n.$$

Then we have

$$\|\delta_\alpha(\theta)\| = \|f_{\bar{\pi},\alpha}(\theta) - \nabla \chi(\theta_m)\| \le (2 + A) T_{\max} C \bar{N} g_{\max} \left(1 + \frac{\beta}{1-\beta} - \frac{\alpha\beta}{1-\alpha\beta} \right),$$

where C *is a bound on* $\|\nabla P_{ij}(\theta)\|$*, and* $\bar{N}$ *is the same bound on the number of of possible transitions from any state* $i \in S$ *that depend on* θ*, as used in Proposition 9.*

Note that the bound on the bias $\delta_\alpha(\theta)$ is decreasing in the mixing constant β. It is interesting to compare the bound on the bias $\delta_\alpha(\theta)$ given in Proposition 11, with the bound on the variance in $\tilde{J}_{\alpha,i}(\theta)$ given in Proposition 7. The bound on the variance in $\tilde{J}_{\alpha,i}(\theta)$ is a function of the discount factor α and $\rho(\theta) = \min_{i \in S} P_{ii^*}(\theta)$, whereas the bound on the bias $\delta_\alpha(\theta)$ is a function of the discount factor α and the mixing constant β on $S \backslash \{i^*\}$. Therefore, Proposition 11 states that if the mixing constant β is small, then we can choose a small discount factor α to reduce the variance in the estimate of the reward-to-go, without introducing a too large a bias into the estimate of the gradient $\nabla\chi(\theta)$.

Before we proof Proposition 11, we show that there indeed exists β, $0 \leq \beta < 1$, and A, such that for all states $i, i' \in S\backslash\{i^*\}$, for all transition matrices $P \in \bar{\mathcal{P}}$ and all $n \geq 1$, we have

$$\sum_{j \in S} |P^n_{ij} - P^n_{i'j}| \leq A\beta^n.$$

By Assumption 6 (MRP Termination), there exists a constant $\rho > 0$, such that for every state $i \in S$ and every matrix $P \in \overline{\mathcal{P}}$, we have

$$P^N_{ii^*} \geq \rho,$$

where N is the number of states in the state space. To see this, recall that the transition probabilities are assumed to be continuous in θ, and that the set $\overline{\mathcal{P}}$ is compact. Therefore, for every state $i \in S$ and every matrix $P \in \overline{\mathcal{P}}$, we have

$$P^n_{ii^*} \geq 1 - (1-\rho)^{\frac{n}{N}-1}.$$

Using this result we have, for all states $i, i' \in S$, that

$$\begin{aligned}
\sum_{j \in S} |P^n_{ij} - P^n_{i'j}| &= |P^n_{ii^*} - P^n_{i'i^*}| + \sum_{j \in S\backslash\{i^*\}} |P^n_{ij} - P^n_{i'j}| \\
&\leq (P_{ii^*} + P_{ii^*}) - 2\left(1 - (1-\rho)^{\frac{n}{N}-1}\right) + \sum_{j \in S\backslash\{i^*\}} (P^n_{ij} + P^n_{i'j}) \\
&\leq \sum_{j \in S} (P^n_{ij} + P^n_{i'j}) - 2\left(1 - (1-\rho)^{\frac{n}{N}-1}\right) \\
&\leq 2(1-\rho)^{\frac{n}{N}-1} \\
&\leq 2(1-\rho)^{-1}(1-\rho)^{\frac{n}{N}} \\
&\leq A\beta^n,
\end{aligned}$$

where $\beta = (1-\rho)^{\frac{1}{N}}$, and $A = 2\beta^{-1}$.

Proof: First, we show that if the constants β, $0 \leq \beta < 1$, and A, are such that, for all states $i, i' \in S\backslash\{i^*\}$, for all transition matrices $P \in \bar{\mathcal{P}}$, and all $n \geq 1$, we have

$$\sum_{j \in S\backslash\{i^*\}} |P^n_{ij} - P^n_{i'j}| \leq A\beta^n,$$

then we have

$$\|f_{\bar{\pi},\alpha}(\theta) - \nabla\chi(\theta_m)\| \leq (2+A)T_{\max}CNg_{\max}\left(\frac{1}{1-\beta} - \frac{1}{1-\alpha\beta}\right),$$

where C is a bound on $\|\nabla P_{ij}(\theta)\|$, and N is the number of states in the state space.

Observe that, for every for every parameter $\theta \in \Re^K$ and for every state $i \in S$, we have

$$\sum_{j\in S} \nabla P_{ij}(\theta) = 0.$$

Therefore, given a state $i_r \in S\backslash\{i^*\}$, we have, for every parameter $\theta \in \Re^K$ and for every state $i \in S$, that

$$\sum_{j\in S} \nabla P_{ij}(\theta) J_{\alpha,j}(\theta) = \sum_{j\in S} \nabla P_{ij}(\theta) \left(J_{\alpha,j}(\theta) + J_{i_r}(\theta) - J_{\alpha,i_r}(\theta)\right).$$

Combining this relation with the fact that $J_{i*}(\theta) = J_{\alpha,i*}(\theta) = 0$, for all parameters $\theta \in \Re^K$, we obtain

$$\begin{aligned}
&\|f_{\bar{\pi},\alpha}(\theta) - \nabla\chi(\theta_m)\| \\
&\quad \leq \left\| E_{\bar{\pi},\theta}[T] \sum_{i\in S} \pi_{\bar{\pi},i}(\theta) \sum_{j\in S\backslash\{i^*\}} \nabla P_{ij}(\theta) \left(J_{\alpha,j}(\theta) - J_{\alpha,i_r}(\theta) - J_j(\theta) + J_{i_r}(\theta)\right)\right\|.
\end{aligned}$$

Next, we show that for all $\theta \in \Re^K$ and all states $i \in S\backslash\{i^*\}$ we have

$$|J_{\alpha,i}(\theta) - J_{\alpha,i_r}(\theta) - J_i(\theta) + J_{i_r}(\theta)| \leq A g_{\max} \left(\frac{1}{1-\beta} - \frac{1}{1-\alpha\beta}\right).$$

We have

$$\begin{aligned}
|J_{\alpha,i}(\theta) - J_{\alpha,i_r}(\theta) - J_i(\theta) + J_{i_r}(\theta)| &= \Big|\sum_{n=0}^{\infty}(1-\alpha^n)\sum_{j\in S}(P^n_{ij}(\theta) - P^n_{i_r j}(\theta))g_j(\theta)\Big| \\
&\leq \sum_{n=0}^{\infty}(1-\alpha^n)\sum_{j\in S}|P^n_{ij}(\theta) - P^n_{i_r j}(\theta)||g_j(\theta)| \\
&\leq 2g_{\max} + \sum_{n=1}^{\infty}(1-\alpha^n) g_{\max} A\beta^n \\
&\leq 2g_{\max} + g_{\max}A\left[\frac{\beta}{1-\beta} - \frac{\alpha\beta}{1-\alpha\beta}\right] \\
&\leq (2+A)g_{\max}\left[1 + \frac{\beta}{1-\beta} - \frac{\alpha\beta}{1-\alpha\beta}\right].
\end{aligned}$$

Finally, by Assumption 5 (MRP Parameterization), there exists a constant C such that $\|\nabla P_{ij}(\theta)\| \leq C$, for all $i, j \in S$, and all $\theta \in \Re^K$, and we have

$$\begin{aligned}
&\|f_{\bar{\pi},\alpha}(\theta) - \nabla\chi(\theta_m)\| \\
&\quad = \left\| E_{\bar{\pi},\theta}[T] \sum_{i\in S} \pi_{\bar{\pi},i}(\theta) \sum_{j\in S_i\backslash\{i^*\}} \nabla P_{ij}(\theta) \left(J_{\alpha,j}(\theta) - J_{i_r}(\theta) - J_j(\theta) + J_{i_r}(\theta)\right)\right\| \\
&\quad \leq E_{\bar{\pi},\theta}[T] \sum_{i\in S} \pi_{\bar{\pi},i}(\theta) \sum_{j\in S_i\backslash\{i^*\}} \|\nabla P_{ij}(\theta)\| \Big| J_{\alpha,j}(\theta) - J_{i_r}(\theta) - J_j(\theta) + J_{i_r}(\theta)\Big| \\
&\quad \leq (2+A) T_{\max} C \bar{N} g_{\max} \left(1 + \frac{\beta}{1-\beta} - \frac{\alpha\beta}{1-\alpha\beta}\right),
\end{aligned}$$

where the set S_i is defined as follows,

$$S_i = \{j \in S \mid \nabla P_{ij}(\theta) \neq 0, \text{ for some } \theta \in \Re^K\}.$$

□

4.6 A Modified Algorithm that Updates at Visits to the Terminal State i^*

In this section, we use the (biased) estimates of the gradient $\nabla\chi(\theta)$ introduced in the previous section to formulate a simulation-based algorithm, which updates the parameter vector θ at visits to the state i^*.

A Modified Algorithm using $F_{S^*,m}(\theta)$

As in Section 4.3, we use the variable m to index the times when the recurrent state i^* is visited, and the corresponding updates. The form of the algorithm is the following. At the time t_m that state i^* is visited for the mth time, we have available a current vector θ_m. We then simulate the process according to the transition probabilities $P_{\bar{\pi},ij}(\theta_m)$ until the next time t_{m+1} that i^* is visited and update according to

$$\theta_{m+1} = \theta_m + \gamma_m F_{S^*,m}(\theta_m),$$

where γ_m is a positive step size sequence.

Recall how in Proposition 9 we provided a bound on the bias $\delta_{S^*}(\theta)$ of the estimate $F_{S^*,m}(\theta)$. We then have the following result.

Proposition 12 *Let Assumptions 5 (MRP Parameterization), 6 (MRP Termination), and 8 (Step Size 1) hold, and let D be such that, for all $\theta \in \Re^K$, we have*

$$\|\delta_{S^*}(\theta)\| \leq D,$$

where $\delta_{S^*}(\theta)$ is as in Proposition 9. Furthermore, let (θ_m) be the sequence of parameter vectors generated by the above described algorithm. Then, with probability 1, we have

$$\liminf_{m\to\infty} \|\nabla\chi(\theta_m)\| \leq D.$$

We provide the proof for this proposition in Appendix B.2.

Proposition 12 establishes that when the bias $\delta_{S^*}(\theta)$ tends to be small, then the gradient $\nabla\chi(\theta_m)$ is small at infinitely many visits to the recurrent state i^*. However, Proposition 12 does not say anything about the behavior of the sequence $\chi(\theta_m)$. In a worst case, $\chi(\theta_m)$ may oscillate significantly throughout the algorithm. Fortunately, gradient algorithms tend to be fairly robust with respect to small biases, and this modified form algorithm may perform better in practice. In practice, it is often not possible to compute the bias $\delta_{S^*}(\theta)$ associated with a set S^* exactly, and one has to resort to trial and error, as well as intuition on a problem at hand, to chose a set S^*.

A Modified Algorithm using $F_{\alpha,m}(\theta)$

Similarly, we can use the estimate $F_{\alpha,m}(\theta)$ of the gradient $\nabla\chi(\theta)$ to to formulate a simulation-based algorithm, which updates the parameter vector θ at the $m+1$ visit to i^* according to

$$\theta_{m+1} = \theta_m + \gamma_m F_{\alpha,m}(\theta_m), \tag{4.11}$$

where γ_m is a positive step size sequence.

Recalling the bound on the bias $\delta_\alpha(\theta)$ provided by Proposition 11, we can formulate the following result.

Proposition 13 *Let Assumptions 5 (MRP Parameterization), 6 (MRP Termination), and 8 (Step Size 1) hold, and let D be such that, for all $\theta \in \Re^K$, we have*

$$\|\delta_\alpha(\theta)\| \leq D,$$

where $\delta_\alpha(\theta)$ is as in Proposition 11. Furthermore let (θ_m) be a sequence of parameter vectors generated by the above described algorithm. Then, with probability 1, we have that

$$\liminf_{m\to\infty} \|\nabla\chi(\theta_m)\| \leq D.$$

Again, it often won't be possible to compute the bias $\delta_\alpha(\theta)$ exactly, and one has to resort to trial and error, as well as intuition about a problem at hand, to chose a discount factor α. We provide the proof for this proposition to Appendix B.2.

4.7 An Algorithm that Updates at Every Time Step

In Section 4.3, we proposed a simulation-based algorithm which updates the parameter vector θ at visits to the state i^*. However, if the intervals between visits to the state i^* are typically very long, then the parameter vector gets updated very infrequently. As an alternative approach, we propose in this section an algorithm that updates the parameter vector at each time step. We start by formulating a recursive way to compute the update direction $F_m(\theta)$ of the algorithm in Section 4.3. This will allow us to break down the total update over an interval between visits to the state i^* into a sum of incremental updates, carried out at each time step. Finally, we use the aforementioned procedure to formulate modified versions of the simulation-based algorithms proposed in Section 4.6.

4.7.1 A Recursive Formula for $F_m(\theta)$

Recall that in the algorithm proposed in Section 4.3, at the mth visit to the state i^*, we updated the parameter vector θ along the direction $F_m(\theta)$ given by

$$F_m(\theta) = \sum_{n=t_m}^{t_{m+1}-1} \left(\tilde{J}_{i_n}(\theta) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \nabla g_{i_n}(\theta) \right), \tag{4.12}$$

where, for $t_m < n \le t_{m+1} - 1$, we have

$$\tilde{J}_{i_n}(\theta) = \sum_{k=n}^{t_{m+1}-1} g_{i_k}(\theta), \tag{4.13}$$

and, for $n = t_m$, we set $J_{i_n}(\theta)$ equal to 0.

We can rewrite Eq.(4.12) as

$$\begin{aligned}
F_m(\theta) &= \sum_{n=t_m+1}^{t_{m+1}-1} \tilde{J}_{i_n}(\theta) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \sum_{n=t_m}^{t_{m+1}-1} \nabla g_{i_n}(\theta) \\
&= \sum_{n=t_m+1}^{t_{m+1}-1} \left(\nabla g_{i_n}(\theta) + \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} \sum_{k=n}^{t_{m+1}-1} g_{i_k}(\theta) \right) + \nabla g_{i^*}(\theta) \\
&= \sum_{k=t_m+1}^{t_{m+1}-1} \left(\nabla g_{i_k}(\theta) + g_{i_k}(\theta) \sum_{n=t_m+1}^{k} \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} \right) + \nabla g_{i^*}(\theta) \\
&= \nabla g_{i^*}(\theta) + \sum_{k=t_m+1}^{t_{m+1}-1} \Big(\nabla g_{i_k}(\theta) + g_{i_k}(\theta) z_k \Big),
\end{aligned}$$

where

$$z_k = \sum_{n=t_m+1}^{k} \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)}, \qquad k = t_m + 1, \ldots, t_{m+1} - 1,$$

is a vector (of the same dimension as θ) that becomes available at time k. The vector z_k can be updated recursively, with

$$z_{t_m} = 0, \tag{4.14}$$

and

$$z_{k+1} = z_k + \frac{\nabla P_{i_k i_{k+1}}(\theta)}{P_{i_k i_{k+1}}(\theta)}, \qquad k = t_m, \ldots, t_{m+1} - 2. \tag{4.15}$$

Note that in order to implement the algorithm, on the basis of the above equations, we only need to maintain in memory $2K$ scalars, namely the vectors θ, and z.

4.7.2 An Algorithm that Updates at Every Time Step

Using the above proposed recursive formula for computing the update direction $F_m(\theta)$, we can break down the total update $F_m(\theta)$ in the algorithm given in Section 4.3 into a sum of incremental updates carried out at each time step.

At a typical time k, the state is i_k, and the values of θ_k, and z_k, are available from the previous iteration. We then update θ according to

$$\theta_{k+1} \;=\; \theta_k + \gamma_k \Big(\nabla g_{i_k}(\theta_k) + g_{i_k}(\theta_k) z_k \Big). \tag{4.16}$$

Next, we simulate a transition to the next state i_{k+1} using the transition probabilities $P_{\bar{\pi},ij}(\theta_{k+1})$, and update z by letting

$$z_{k+1} = \begin{cases} 0, & \text{if } i_{k+1} = i^*, \\ z_k + \dfrac{\nabla P_{i_k i_{k+1}}(\theta_k)}{P_{i_k i_{k+1}}(\theta_k)}, & \text{otherwise.} \end{cases} \tag{4.17}$$

Since at the kth time step, the next state i_{k+1} is chosen according to the transition probabilities probabilities $P_{ij}(\theta_{k+1})$, the algorithm can be implemented in an online fashion.

To prove convergence of this version of the algorithm, we need to strengthen Assumption 6 (MRP Termination) that we used in Section 4.3. Assumption 6 ensured that if we fix the parameter vector θ, then we will eventually reach the state i^*. Here, we have to make sure that this property still holds for the situation where we change the parameter vector at each time step.

Assumption 10 (MRP Strong Termination) *There exist a state $i^* \in S$ and a positive integer N_0, such that, for every parameter vector $\theta \in \Re^K$, we have*

$$g_{i^*}(\theta) = 0 \qquad \textit{and} \qquad P_{i^*i^*}(\theta) = 1,$$

and, for every state $i \in S$ and every collection $\{P_1, ..., P_{N_0}\}$ of N_0 matrices in the set $\overline{\mathcal{P}}$, we have

$$Q_{ii^*} > 0,$$

where the matrix Q is given by

$$Q = P_1 \cdots P_N.$$

Assumption 10 states that there exists an integer N_0, so that we reach the state i^* in N_0 steps with a positive probability, regardless of the value of the parameter vector θ at each time step. We will discuss this assumption in further detail at the end of this section.

We have the following convergence result for the algorithm described by (4.16) and (4.17).

Proposition 14 *Let Assumptions 5 (MRP Parameterization), 8 (Step Size 1), 9 (Step Size 2), and 10 (MRP Strong Termination) hold, and let (θ_k) be the sequence of parameter vectors generated by the above described algorithm. Then, $\chi(\theta_k)$ converges and*

$$\lim_{k\to\infty} \nabla\chi(\theta_k) = 0,$$

with probability 1.

The algorithm described by Eq.(4.16) and (4.17) is similar to the algorithm in Section 4.3, except that θ is continually updated in the course of a renewal cycle. Because of the diminishing step size, these incremental updates are asymptotically negligible and the difference between the two algorithms is inconsequential. Given that the algorithm in Section 4.3 converges by Proposition 5, Proposition 14 is hardly surprising. We provide a proof of Proposition 14 in Appendix B.3. The technique will use is very similar to the one presented in [CR94]. However, translated into the context of parameterized Markov Reward Processes, [CR94] assumes that the transition probabilities $P_{ij}(\theta)$ are independent of θ (the one-stage rewards $g_i(\theta)$ can still depend on θ). This is not the situation we consider here, and need to extend the proof accordingly.

4.7.3 A Modified Algorithm using $F_{S^*,m}(\theta)$

Recall that in Section 4.5, using a subset S^* of the state space S containing the state i^*, we derived an estimate $F_{S^*,m}(\theta)$ of the gradient $\nabla\chi_{\bar{\pi}}(\theta)$, which we used in Section 4.6 to formulate an algorithm that updates the parameter vector at visits to the state i^*. Using the same idea as in the preceding subsections, we can derive a recursive way to compute the update direction $F_{S^*,m}(\theta_m)$, and formulate a modified version of the algorithm of Section 4.6 where we now update the parameter vector at each time step.

As in Subsection 4.5.1, let $(i_1, i_2, ...)$ be a trajectory of the Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$, and let t_m be the time of the mth time to the terminal state i^*. Furthermore, let $\kappa(m)$ be the number of times a state in the set $S^*\backslash\{i^*\}$ is visited in the interval $k = t_m, ..., t_{m+1}-1$, let the $t_{m,n}$ be the time when for the nth time a state in S^* is visited, and let $t_{m,0}$ and $t_{m,\kappa(m)+1}$ be equal to t_m and t_{m+1}, respectively. Using these definitions, and taking into account that $\tilde{v}_{i^*}(\theta) = 0$, we have

$$
\begin{aligned}
F_{S^*,m}(\theta) &= \sum_{n=t_m}^{t_{m+1}-1}\left(\tilde{J}_{S^*,i_n}(\theta)\frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \nabla g_{i_n}(\theta)\right) \\
&= \nabla g_{i^*}(\theta) + \sum_{l=t_m+1}^{t_{m,1}-1}\left(\nabla g_{i_l}(\theta) + \frac{\nabla P_{i_{l-1}i_l}(\theta)}{P_{i_{l-1}i_l}(\theta)}\sum_{k=l}^{t_{m,1}} g_{i_k}(\theta)\right) + \\
&\quad \sum_{n=1}^{\kappa(m)}\left(\sum_{l=t_{m,n}}^{t_{m,n+1}-1}\left(\nabla g_{i_l}(\theta) + \frac{\nabla P_{i_{l-1}i_l}(\theta)}{P_{i_{l-1}i_l}(\theta)}\sum_{k=l}^{t_{m,n+1}-1} g_{i_k}(\theta)\right)\right) \\
&= \nabla g_{i^*}(\theta) + \sum_{k=t_m+1}^{t_{m,1}-1}\left(\nabla g_{i_k}(\theta) + g_{i_k}(\theta)\sum_{l=t_m+1}^{k}\frac{\nabla P_{i_{l-1}i_l}(\theta)}{P_{i_{l-1}i_l}(\theta)}\right) + \\
&\quad \sum_{n=1}^{\kappa(m)}\left(\sum_{k=t_{m,n}}^{t_{m,n+1}-1}\left(\nabla g_{i_k}(\theta) + g_{i_k}(\theta)\sum_{l=t_{m,n}}^{k}\frac{\nabla P_{i_{l-1}i_l}(\theta)}{P_{i_{l-1}i_l}(\theta)}\right)\right) \\
&= \nabla g_{i^*}(\theta) + \sum_{k=t_m+1}^{t_{m,1}-1}\left(\nabla g_{i_k}(\theta) + g_{i_k}(\theta)z_k\right) + \\
&\quad \sum_{n=1}^{\kappa(m)}\left(\sum_{k=t_{m,n}}^{t_{m,n+1}-1}\left(\nabla g_{i_k}(\theta) + g_{i_k}(\theta)z_k\right)\right),
\end{aligned}
$$

where

$$
z_k = \begin{cases} \sum_{l=t_m+1}^{k}\frac{\nabla P_{i_{l-1}i_l}(\theta)}{P_{i_{l-1}i_l}(\theta)}, & k = t_m+1, \ldots, t_{m,1}-1, \\ \sum_{l=t_{m,n}}^{k}\frac{\nabla P_{i_{l-1}i_l}(\theta)}{P_{i_{l-1}i_l}(\theta)}, & k = t_{m,n}, \ldots, t_{m,n+1}-1, \quad n = 1, ..., \kappa(m). \end{cases}
$$

The vector z_k can be updated recursively, with

$$
z_{t_m} = 0, \tag{4.18}
$$

and

$$z_{k+1} = \begin{cases} \frac{\nabla P_{i_k i_{k+1}}(\theta)}{P_{i_k i_{k+1}}(\theta)}, & k+1 = t_{m,n}, \qquad n = 1, ..., \kappa(m), \\ z_k + \frac{\nabla P_{i_k i_{k+1}}(\theta)}{P_{i_k i_{k+1}}(\theta)}, & k+1 = t_{m,n}+1, \ldots, t_{m,n+1}-1, \qquad n = 1, ..., \kappa(m). \end{cases} \tag{4.19}$$

Accordingly, we can break down the total update $F_{S^*,m}(\theta_m)$ in the algorithm proposed in Section 4.6 into a sum of incremental updates carried out at each time step. At a typical time k, the state is i_k, and the values of θ_k, and z_k, are available from the previous iteration. We update θ according to

$$\theta_{k+1} \quad = \quad \theta_k + \gamma_k \Big(\nabla g_{i_k}(\theta_k) + g_{i_k}(\theta_k) z_k \Big).$$

We then simulate a transition to the next state i_{k+1} according to the transition probabilities $P_{\bar{\pi},ij}(\theta_{k+1})$, and update z by letting

$$z_{k+1} = \begin{cases} 0, & \text{if } i_{k+1} = i^*, \\ \dfrac{\nabla P_{i_k i_{k+1}}(\theta_k)}{P_{i_k i_{k+1}}(\theta_k)}, & \text{if } i_{k+1} \in S^*, \\ z_k + \dfrac{\nabla P_{i_k i_{k+1}}(\theta_k)}{P_{i_k i_{k+1}}(\theta_k)}, & \text{otherwise.} \end{cases}$$

Recall that $F_{S^*,m}(\theta_m)$ is a biased estimate of the gradient $\chi(\theta)$, for which the bias $\delta_{S^*}(\theta)$ is bounded by Proposition 9 given in Section 4.5. We have the following proposition.

Proposition 15 *Let Assumptions 5 (MRP Parameterization), 8 (Step Size 1), 9 (Step Size 2), and 10 (MRP Strong Termination) hold, and let D be such that, for all $\theta \in \Re^K$, we have*

$$\|\delta_{S^*}(\theta)\| \leq D,$$

where $\delta_{S^*}(\theta)$ is as in Proposition 9. Furthermore let (θ_m) be a sequence of parameter vectors generated by the above described algorithm. Then, with probability 1, we have

$$\liminf_{m\to\infty} \|\nabla \chi_{\bar{\pi}}(\theta_m)\| \leq D.$$

This proposition can be proven by the same argument as the one given in Appendix B.3 to prove Proposition 14.

4.7.4 A Modified Algorithm using $F_{\alpha,m}(\theta)$

Similarly, we can derive an algorithm which updates the parameter vector at every time step based on the estimate $F_{\alpha,m}(\theta)$ of the gradient $\nabla\chi(\theta)$ that we introduced in Subsection 4.5.2.

We can rewrite $F_{\alpha,m}(\theta)$ as

$$\begin{aligned} F_{\alpha,m}(\theta) \quad &= \quad \sum_{n=t_m+1}^{t_{m+1}-1} \tilde{J}_{\alpha,i_n}(\theta) \frac{\nabla P_{i_{n-1} i_n}(\theta)}{P_{i_{n-1} i_n}(\theta)} + \sum_{n=t_m}^{t_{m+1}-1} \nabla g_{i_n}(\theta) \\ &= \quad \nabla g_{i^*}(\theta) + \sum_{k=t_m+1}^{t_{m+1}-1} \Big(\nabla g_{i_k}(\theta) + g_{i_k}(\theta) z_k \Big), \end{aligned}$$

where

$$z_k = \sum_{n=t_m+1}^{k} \alpha^{k-n} \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)}, \qquad k = t_m + 1, \ldots, t_{m+1} - 1,$$

is a vector (of the same dimension as θ) that becomes available at time k. The vector z_k can be updated recursively, with

$$z_{t_m} = 0, \tag{4.20}$$

and

$$z_{k+1} = z_k + \alpha \frac{\nabla P_{i_k i_{k+1}}(\theta)}{P_{i_k i_{k+1}}(\theta)}, \qquad k = t_m, \ldots, t_{m+1} - 2. \tag{4.21}$$

Using these relations, we obtain the following algorithm. At a typical time k, the state is i_k, and the values of θ_k, and z_k, are available from the previous iteration. We update θ according to

$$\theta_{k+1} \quad = \quad \theta_k + \gamma_k \Big(\nabla g_{i_k}(\theta_k) + g_{i_k}(\theta_k) z_k \Big).$$

We then simulate a transition to the next state i_{k+1} according to the transition probabilities $P_{\bar{\pi},ij}(\theta_{k+1})$, and update z by letting

$$z_{k+1} = \begin{cases} 0, & \text{if } i_{k+1} \in S^*, \\ z_k + \alpha \dfrac{\nabla P_{i_k i_{k+1}}(\theta_k)}{P_{i_k i_{k+1}}(\theta_k)}, & \text{otherwise.} \end{cases}$$

We have the following proposition, which uses the fact that the bias $\delta_\alpha(\theta)$ in the estimate of the gradient $F_{\alpha,m}(\theta_m)$ is bounded by Proposition 11.

Proposition 16 *Let Assumptions 5 (MRP Parameterization), 8 (Step Size 1), 9 (Step Size 2), and 10 (MRP Strong Termination) hold, and let D be such that, for all $\theta \in \Re^K$, we have*

$$\|\delta_\alpha(\theta)\| \leq D,$$

where $\delta_\alpha(\theta)$ is as in Proposition 11. Furthermore let (θ_m) be a sequence of parameter vectors generated by the above described algorithm. Then, with probability 1, we have

$$\liminf_{m \to \infty} \|\nabla \chi_{\bar{\pi}}(\theta_m)\| \leq D.$$

This proposition can be proven by the same argument as the one given in Appendix B.3 to prove Proposition 14.

4.7.5 Assumption 10 (MRP Strong Termination)

Here, we comment on Assumption 10 (MRP Strong Termination) which states that there exists a reward-free terminal state i^* and a positive integer N_0 such that for every state $i \in S$ and every collection $\{P_1, ..., P_{N_0}\}$ of N_0 matrices from the set $\overline{\mathcal{P}}$, we have

$$Q_{ii^*} > 0,$$

where

$$Q = P_1 \cdots P_N.$$

Let us start by pointing out why we were not able to rely on the weaker Assumption 6 (MRP Termination) to prove convergence for the algorithms proposed in this section. Recall that Assumption 6 states that there exists a reward-free terminal state i^*, such that, for all states $i \in S$ and all matrices $P \in \overline{\mathcal{P}}$, we have

$$P_{ii^*}^N > 0,$$

where N is the number of states in the state space S. Assumption 6, together with Assumption 5 (MRP Parameterization) (the transition probabilities $P_{ij}(\theta)$ are continuous in θ), implies that if the parameter vector θ_k changes by not too much within N time steps, then we return to the state i^* in N time steps with a positive probability, i.e., there exist $\rho' > 0$, and $\delta > 0$, such that if

$$\|\theta_k - \theta_{k+n}\| \leq \delta, \qquad n = 1, ..., N,$$

then we have, for all states $i \in S$,

$$Q_{ii^*} \geq \rho' > 0, \quad \text{for all states } i \in S,$$

where the matrix Q is given by

$$Q = P(\theta_k) \cdots P(\theta_{k+N}).$$

Using this result one may expect that, contrary to our statement, Assumption 6 is sufficient to prove convergence. Indeed, Assumption 6 (MRP Termination) would imply Assumption 10 (MRP Strong Termination), if we are able to show that the changes in θ_k over N time steps are bounded by a constant times the step size γ_k, i.e., there exists a constant C such that

$$\|\theta_k - \theta_{k+n}\| \leq \gamma_k C, \quad n = 1, 2, ..., N.$$

Unfortunately, in the update

$$\theta_{k+1} = \theta_k + \gamma_k\Big(\nabla g_{i_k}(\theta_k) + g_{i_k}(\theta_k) z_k\Big),$$

of the parameter vector θ_k, the vector z_k tends to grow linearly in k, i.e., we have

$$\|z_k\| \leq \gamma_k (t_k - t_m) C_z, \quad t_m \leq k < t_{m+1} - 1,$$

for some constant C_z. This implies that a bound on the changes in θ_k will be of the form

$$\|\theta_k - \theta_{k+n}\| \leq \gamma_k (t_k - t_m)^2 C, \quad t_m \leq k < t_{m+1} - 1, \quad n = 1, 2, ..., N,$$

for some constant C. Therefore, the changes $\|\theta_k - \theta_{k+1}\|$, $k = t_m, ..., t_{m+1} - 2$, in the parameter vector tend to get larger as k increases. This observation leads us to consider Assumption 10 (MRP Strong Termination), which ensures that no matter how large the changes in the parameter vector, there is always a positive probability that the state i^* is reached in N_0 time steps. It is unclear whether a weaker assumption could be used instead.

Verifying Assumption 10 may be difficult in practice. However, using the sets S_i, $i \in S$, given by

$$S_i = \{j \in S \mid \nabla P_{ij}(\theta) \neq 0 \text{ for some } \theta \in \Re^K\},$$

we can formulate the following result.

Lemma 7 *Assume that there exists a scalar $\epsilon > 0$, such that, for every element $P \in \overline{\mathcal{P}}$ and for every state $i \in S$, we have*

$$P_{ij} \geq \epsilon > 0, \qquad \textit{for all } j \in S_i.$$

Furthermore, assume that there exist a state $i^* \in S$, and a matrix $P \in \overline{\mathcal{P}}$, such that for every state $i \in S$ we have

$$P_{ii^*}^N > 0,$$

where N is the number of states in the state space S. Then, for every state $i \in S$, and every collection $\{P_1, ..., P_N\}$ of N matrices in the set $\overline{\mathcal{P}}$, we have

$$Q_{ii^*} > 0,$$

where

$$Q = \Pi_{n=1}^N P_n.$$

Proof: By assumption, there exists a matrix $P \in \overline{\mathcal{P}}$, such that for every state $i \in S$ we have

$$P_{ii^*}^N > 0.$$

Therefore, for every state $i \in S$, there exists a path $(i_1 = i, i_2, ..., i_N = i^*)$, such that

$$\Pi_{n=1}^{N-1} P_{i_n i_{n+1}} > 0.$$

And for every matrix $P' \in \overline{\mathcal{P}}$, we have

$$P'_{i_n i_{n+1}} \geq \delta_i = \min\{\epsilon_i, \epsilon\} > 0,$$

where ϵ_i is given by

$$\epsilon_i = \min_{n \in \{1,...,N-1\}} P_{i_n i_{n+1}}$$

and is therefore positive. This implies that for every collection $\{P_1, ..., P_N\}$ of N matrices in the set $\overline{\mathcal{P}}$, we have

$$Q_{ii^*} \geq \delta_i^N > 0,$$

where

$$Q = P_1 \cdot ... \cdot P_N.$$

□

Lemma 7 provides a condition which can easily be enforced in practice. To see this, let us consider the following example.

Example 7 *Consider a parameterized Markov reward process on the state space $S = \{1, ..., N\}$ with transition probabilities $P_{ij}(\theta)$, $\theta \in \Re^K$. Assume that there exist a state $i^* \in S$, a matrix $P \in \overline{\mathcal{P}}$, such that for every state $i \in S$ we have*

$$P^N_{ii^*} > 0.$$

Furthermore, let $\epsilon > 0$ be a scalar such that $N\epsilon \leq 1$, and, for every state $i \in S$, let the scalar ϵ_i be equal to $N_i\epsilon$, where N_i is the number of states in the set $S_i = \{j \in S \mid \nabla P_{ij}(\theta) \neq 0 \text{ for some } \theta \in \Re^K\}$.

Next consider the parameterized Markov reward process on the same state space, whose transition probabilities are given as follows,

$$P'_{ij}(\theta) = \begin{cases} \epsilon + (1 - \epsilon_i)P_{ij}(\theta), & \text{if } j \in S_i, \\ P_{ij}(\theta), & \text{otherwise.} \end{cases}$$

Note that, for every $\theta \in \Re^K$, the matrix $P'(\theta)$ with components $P'_{ij}(\theta)$, is again a stochastic matrix. Let $\mathcal{P}' = \{P'(\theta) \mid \theta \in \Re^K\}$ be the set of all such matrices, and let $\overline{\mathcal{P}}'$ be its closure. Then it can be easily verified that for every element $P \in \overline{\mathcal{P}}'$, and for every state $i \in S$, we have $P'_{ij} \geq \epsilon$, $j \in S_i$.

Chapter 5

Optimizing the Average Reward

In this chapter, we consider Markov reward processes which depend on a parameter $\theta \in \Re^K$, and for which Assumption 7 (MRP Recurrence) holds, i.e., there exists a state $i^* \in S$ such that, for every parameter $\theta \in \Re^K$, i^* is eventually reached from every state $i \in S$. Furthermore, we consider the case where a decision maker tunes the parameter θ so as to optimize the average reward

$$\lambda(\theta) = \sum_{i \in S} \pi_i(\theta) g_i(\theta),$$

where $\pi_i(\theta)$ is the steady state probability of being in state i in the Markov chain with transition probabilities $P_{ij}(\theta)$, and where $g_i(\theta)$ is the immediate reward associated with state i and parameter θ. The program that we follow is similar to the one of Chapter 4. We will use estimates of the differential reward

$$v_i(\theta) = E_\theta\left[\sum_{k=0}^{T-1} \left(g_{i_k}(\theta) - \lambda(\theta)\right) \mid i_0 = i\right], \tag{5.1}$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited, to produce an estimate of the gradient $\lambda(\theta)$ and to update the parameter vector θ. More precisely, given a trajectory $(i_0 = i, i_1, i_2, ...)$ of the Markov chain with transition probabilities $P_{ij}(\theta)$ which starts at the state i, we use the expression

$$\tilde{v}_i(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} (g_{i_k}(\theta) - \tilde{\lambda}) \tag{5.2}$$

to estimate $v_i(\theta)$. Here, $\tilde{\lambda}$ is an estimate of the average reward $\lambda(\theta)$, and T is the first future time that state i^* is visited. An estimate of the gradient can then be produced by accumulating such estimates along the trajectory. However, unless $\tilde{\lambda}$ is equal to $\lambda(\theta)$, the expression given by Eq.(5.2) yields a biased rather than an unbiased estimate of the differential reward $v_i(\theta)$. As a consequence, we will obtain a biased estimate of the gradient $\nabla\lambda(\theta)$. This fact will complicate the analysis in this chapter. In particular, we will not be able to use standard results of stochastic approximations to establish the desired convergence results, but have to devise proofs especially tailored to this problem. Unbiased estimates of the gradient $\lambda(\theta)$ exist [Gly86], but it does not appear possible to employ them in an algorithm that updates the parameter vector at each step. We comment on this issue in more detail in Section 5.6 at the end of this chapter.

The remainder of this chapter is structured similarly to Chapter 4. In Section 5.1, we derive an expression for the derivative $\nabla\lambda(\theta)$ of the average reward, and propose a gradient algorithm for tuning θ so as to improve $\lambda(\theta)$. Again, for problems involving a large state space, it is not feasible to compute $\nabla\lambda(\theta)$ exactly. In Section 5.2 we consider the approach where we employ simulation to obtain an estimate of the derivative $\nabla\lambda(\theta)$ as described above. Using this estimate, we then propose a simulation based algorithm which updates the parameter vector at visits to the recurrent state i^*. If the time T until we visit the termination state i^* is large, then the variance of the estimate $\tilde{v}_i(\theta, \tilde{\lambda})$ can become quite large. In Section 5.4, we consider two approaches to reduce the the variance in the estimate of the differential reward, which we then use to formulate a modified algorithm to update θ. Finally, in Section 5.5, we derive an algorithm, that updates the parameter vector at every time step.

5.1 The Gradient of $\lambda(\theta)$

In this section we derive an expression for the the gradient of the average reward

$$\lambda(\theta) = \sum_{i \in S} \pi_i(\theta) g_i(\theta),$$

with respect to θ, and propose a gradient algorithm for tuning θ so as to improve $\lambda(\theta)$.

Recall from Chapter 3 that, when Assumption 7 (MRP Recurrence) holds, the differential reward $v_i(\theta)$ of state $i \in S$ is defined by

$$v_i(\theta) = E_\theta \left[\sum_{k=0}^{T-1} (g_{i_k}(\theta) - \lambda(\theta)) \mid i_0 = i \right],$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. Using this definition, we obtain the following expression for the gradient of the average reward $\lambda(\theta)$, with respect to θ.

Proposition 17 *Let Assumptions 5 (MRP Parameterization) and 7 (MRP Recurrence) hold. Then,*

$$\nabla\lambda(\theta) = \sum_{i \in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta) v_j(\theta) \right).$$

This expression for the gradient of the average reward $\lambda(\theta)$ with respect to θ has been presented in various places [CC97, CW98, FH94, JSJ95]. Proposition 17 has a nice intuitive interpretation: in order to increase the average reward, one should increase the probabilities of transitions to states which have a high differential reward. We provide a proof of this result which is due to [Tsi98].

Proof: Note that, for all parameter vectors $\theta \in \Re^K$, we have

$$\sum_{i \in S} \nabla \pi_i(\theta) = 0,$$

as the sum of the steady state probabilities over all states $i \in S$ is equal to 1. Recall the following relation for the differential reward $v_i(\theta)$,

$$v_i(\theta) = g_i(\theta) - \lambda(\theta) + \sum_{j \in S} P_{ij}(\theta) v_j(\theta).$$

Combining these two results, we obtain

$$\begin{aligned}
\nabla\lambda(\theta) &= \sum_{i \in S} \pi_i(\theta)\nabla g_i(\theta) + \sum_{i \in S} \nabla\pi_i(\theta) g_i(\theta) \\
&= \sum_{i \in S} \pi_i(\theta)\nabla g_i(\theta) + \sum_{i \in S} \nabla\pi_i(\theta)(g_i(\theta) - \lambda(\theta)) \\
&= \sum_{i \in S} \pi_i(\theta)\nabla g_i(\theta) + \sum_{i \in S} \nabla\pi_i(\theta)\left(v_i(\theta) - \sum_{j \in S} P_{ij}(\theta) v_j(\theta)\right).
\end{aligned}$$

We make the following observations. Using the chain rule, we obtain

$$\nabla\Big(\pi_i(\theta)P_{ij}(\theta)\Big) = \nabla\pi_i(\theta)P_{ij}(\theta) + \pi_i(\theta)\nabla P_{ij}(\theta),$$

and using the balance equations given in Chapter 3, we have

$$\pi_j(\theta) = \sum_{i \in S} \pi_i(\theta) P_{ij}(\theta).$$

It then follows, that

$$\begin{aligned}
\nabla\lambda(\theta) &= \sum_{i \in S} \pi_i(\theta)\nabla g_i(\theta) + \sum_{i \in S} \nabla\pi_i(\theta)\left(v_i(\theta) - \sum_{j \in S} P_{ij}(\theta) v_j(\theta)\right) \\
&= \sum_{i \in S} \pi_i(\theta)\nabla g_i(\theta) + \sum_{i \in S} \nabla\pi_i(\theta) v_i(\theta) + \\
&\quad \sum_{i,j \in S} \Big(\pi_i(\theta)\nabla P_{ij}(\theta) - \nabla\Big(\pi_i(\theta)P_{ij}(\theta)\Big)\Big) v_j(\theta) \\
&= \sum_{i \in S} \pi_i(\theta)\nabla g_i(\theta) + \sum_{i \in S} \nabla\pi_i(\theta) v_i(\theta) + \\
&\quad \sum_{i,j \in S} \pi_i(\theta)\nabla P_{ij}(\theta) v_j(\theta) - \sum_{j} \nabla\left(\sum_{i \in S} \pi_i(\theta) P_{ij}(\theta)\right) v_j(\theta) \\
&= \sum_{i \in S} \pi_i(\theta)\nabla g_i(\theta) + \sum_{i} \nabla\pi_i(\theta) v_i(\theta) + \\
&\quad \sum_{i,j \in S} \pi_i(\theta)\nabla P_{ij}(\theta) v_j(\theta) - \sum_{j \in S} \nabla\pi_j(\theta) v_j(\theta) \\
&= \sum_{i \in S} \pi_i(\theta)\left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta) v_j(\theta)\right).
\end{aligned}$$

$\square$

It is interesting to note the the expression

$$\lambda(\theta) = \sum_{i \in S} \pi_i(\theta) g_i(\theta)$$

for the average reward $\lambda(\theta)$ suggests that the gradient of $\nabla\lambda(\theta)$ would involve gradients of the steady state probabilities $\nabla\pi_i(\theta)$. However, the expression for $\nabla\lambda(\theta)$ given by Proposition 17 involves no such terms. This property of the expression for the gradient given by Proposition 17 will turn out to be helpful in producing simulation-based estimates of $\nabla\lambda(\theta)$.

5.1.1 An idealized gradient algorithm

Using the gradient of the average as given by Proposition 17, we can formulate an idealized gradient algorithm of the following form. Starting with an initial parameter vector $\theta_0 \in \Re^K$, the parameter vector gets iteratively updated according to the rule

$$\begin{aligned}\theta_{k+1} &= \theta_k + \gamma_k \nabla\lambda(\theta_k) \\ &= \theta_k + \gamma_k \left(\sum_{i\in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j\in S} \nabla P_{ij}(\theta) v_j(\theta) \right) \right),\end{aligned}$$

where γ_k is a step size parameter.

We have the following convergence result.

Proposition 18 *Let Assumptions 5 (MRP Parameterization) and 7 (MRP Recurrence) hold, and let $\theta_0 \in R^k$ be an initial parameter vector. Consider the iteration defined by the rule*

$$\begin{aligned}\theta_{k+1} &= \theta_k + \gamma_k \nabla\lambda(\theta_k) \\ &= \theta_k + \gamma_k \left(\sum_{i\in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j\in S} \nabla P_{ij}(\theta) v_j(\theta) \right) \right).\end{aligned}$$

Then, there exists an interval $[a,b]$, with $0 < a < b$, such that if $\gamma_k \in [a,b]$ for every k, then $\lambda(\theta_k)$ converges and

$$\lim_{k\to+\infty} \|\nabla\lambda(\theta_k)\| = 0.$$

Proof: From Lemma 4, it follows that the average reward $\lambda(\theta)$ is bounded, and that there exists a constant L, such that

$$\|\nabla\lambda(\theta) - \nabla\lambda(\theta')\| \le L\|\theta - \theta'\|, \qquad \text{for all } \theta, \theta' \in R^k.$$

The proposition then follows immediately from the convergence result for gradient methods given by Proposition 3 in Section 4.1. □

5.2 Estimation of $\nabla\lambda(\theta)$

In the previous section we derived an algorithm where the parameter vector θ gets updated along a (exact) gradient direction. In this section we consider an alternative approach, where the decision maker employs simulation to obtain estimates of the gradient $\nabla\lambda(\theta)$. We will follow a similar program to the one of Section 4.2. However, while in Section 4.2 we were able produced an unbiased estimate of the gradient, here we will no longer enjoy this property.

We rewrite the formula for $\nabla\lambda(\theta)$ in the form

$$\nabla\lambda(\theta) = \sum_{i\in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j\in S_i(\theta)} P_{ij}(\theta) \left(\frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)} v_j(\theta) \right) \right), \tag{5.3}$$

where the set $S_i(\theta)$ is defined by

$$S_i(\theta) = \{j \in S \mid \nabla P_{ij}(\theta) \neq 0\}.$$

Again, Assumption 5 (MRP Parameterization) ensures that the quotient $\frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)}$ in the above expression is well defined.

Let $(i_0 = i, i_1, i_2, ...)$ be a trajectory of the Markov chain with transition probabilities $P_{ij}(\theta)$, and let t_m be the time when the recurrent state i^* is visited for the m-th time in this trajectory. The definition (5.1) of $v_j(\theta, \tilde{\lambda})$, suggests that if $t_m < n \leq t_{m+1} - 1$, we can use

$$\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{t_{m+1}-1} \left(g_{i_k}(\theta) - \tilde{\lambda} \right), \tag{5.4}$$

to estimate $v_{i_n}(\theta)$, where $\tilde{\lambda}$ is an estimate of $\lambda(\theta)$. Note that $v_{i^*}(\theta) = 0$ and does not need to be estimated. For this reason, we let

$$\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = 0, \qquad \text{if } n = t_m.$$

By accumulating the above described estimates over a renewal cycle, we obtain an estimate of $\nabla\lambda(\theta)$ given by

$$F_m(\theta, \tilde{\lambda}) = \sum_{n=t_m}^{t_{m+1}-1} \left(\tilde{v}_{i_n}(\theta, \tilde{\lambda}) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \nabla g_{i_n}(\theta) \right). \tag{5.5}$$

The random variables $F_m(\theta, \tilde{\lambda})$ are independent and identically distributed for different values of m, because the transitions during distinct renewal cycles are independent.

We define $f(\theta, \tilde{\lambda})$ to be the expected value of $F_m(\theta, \tilde{\lambda})$, namely,

$$f(\theta, \tilde{\lambda}) = E_\theta[F_m(\theta, \tilde{\lambda})]. \tag{5.6}$$

The following proposition confirms that the expectation of $F_m(\theta, \tilde{\lambda})$ is aligned with $\nabla\lambda(\theta)$, to the extent that $\tilde{\lambda}$ is close to $\lambda(\theta)$.

Proposition 19 *We have*

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

where

$$G(\theta) = E_\theta \left[\sum_{n=t_m+1}^{t_{m+1}-1} (t_{m+1} - n) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} \right]. \tag{5.7}$$

The proof of Proposition 19 is similar that of Proposition 4 in Section 4.2. As this result plays a central role in this chapter, we provide a detailed proof.

Proof: Note that for $n = t_m + 1, \ldots, t_{m+1} - 1$, we have

$$\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{t_{m+1}-1} (g_{i_k}(\theta) - \lambda(\theta)) + (t_{m+1} - n)(\lambda(\theta) - \tilde{\lambda}).$$

Therefore,

$$\begin{aligned} F_m(\theta, \tilde{\lambda}) \;\; = \;\; & \sum_{n=t_m+1}^{t_{m+1}-1} a_n \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \sum_{n=t_m+1}^{t_{m+1}-1} (t_{m+1} - n)(\lambda(\theta) - \tilde{\lambda}) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \\ & \sum_{n=t_m}^{t_{m+1}-1} \nabla g_{i_n}(\theta), \end{aligned}$$

where

$$a_n = \sum_{k=n}^{t_{m+1}-1} (g_{i_k}(\theta) - \lambda(\theta)). \tag{5.8}$$

We consider separately the expectations of the three sums above. Using the definition of $G(\theta)$, the expectation of the second sum is equal to $G(\theta)(\lambda(\theta) - \tilde{\lambda})$. We then consider the third sum. The expected sum of rewards over a renewal cycle is equal to the steady-state expected reward times the expected length of the renewal cycle. Therefore, the expectation of the third sum is

$$E_\theta \left[\sum_{n=t_m}^{t_{m+1}-1} \nabla g_{i_n}(\theta) \right] = E_\theta[T] \sum_{i \in S} \pi_i(\theta) \nabla g_i(\theta). \tag{5.9}$$

We now focus on the expectation of the first sum. For $n = t_{m+1}, \ldots, t_{m+1} - 1$, let

$$\Delta_n = (a_n - v_{i_n}(\theta)) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)}.$$

Let $\mathcal{F}_n = \{i_0, \ldots, i_n\}$ stand for the history of the process up to time n. By comparing the definition (5.8) of a_n with the definition (5.1) of $v_{i_n}(\theta)$, we obtain

$$E_\theta [a_n \mid \mathcal{F}_n] = v_{i_n}(\theta). \tag{5.10}$$

It follows that $E_\theta[\Delta_n \mid \mathcal{F}_n] = 0$.

Let $\chi_n = 1$ if $n < t_{m+1}$, and $\chi_n = 0$, otherwise. For any $n > t_m$, we have

$$E_\theta[\chi_n \Delta_n \mid \mathcal{F}_{t_m}] = E_\theta[E_\theta[\chi_n \Delta_n \mid \mathcal{F}_n] \mid \mathcal{F}_{t_m}] = E_\theta[\chi_n E_\theta[\Delta_n \mid \mathcal{F}_n] \mid \mathcal{F}_{t_m}] = 0.$$

We then have

$$\begin{aligned} E_\theta \left[\sum_{n=t_m+1}^{t_{m+1}-1} \Delta_n \mid \mathcal{F}_{t_m} \right] &= E_\theta \left[\sum_{n=t_m+1}^{\infty} \chi_n \Delta_n \mid \mathcal{F}_{t_m} \right] \\ &= \sum_{n=t_m+1}^{\infty} E_\theta[\chi_n \Delta_n \mid \mathcal{F}_{t_m}] \\ &= 0. \end{aligned}$$

(The interchange of the summation and the expectation can be justified by appealing to the dominated convergence theorem.)

We therefore have

$$E_\theta\left[\sum_{n=t_m+1}^{t_{m+1}-1} a_n \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)}\right] = E\left[\sum_{n=t_m+1}^{t_{m+1}-1} v_{i_n}(\theta)\frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)}\right].$$

The right-hand side can be viewed as the total reward over a renewal cycle of a Markov reward process, where the reward associated with a transition from i to j is $v_j(\theta)\nabla P_{ij}(\theta)/P_{ij}(\theta)$. Recalling that any particular transition has steady-state probability $\pi_i(\theta)P_{ij}(\theta)$ of being from i to j, we obtain

$$E_\theta\left[\sum_{n=t_m+1}^{t_{m+1}-1} a_n \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)}\right] = E_\theta[T]\sum_{i\in S}\sum_{j\in S}\pi_i(\theta)P_{ij}(\theta)\left(\frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)}v_j(\theta)\right). \tag{5.11}$$

Here, we also used the fact that, by Assumption 5 (MRP Parameterization), the transition probability $P_{ij}(\theta)$ is equal to 0 for all states j that are not contained in the set $S_i(\theta)$. By combining Eqs.(5.9) and (5.11), and comparing with the formula for $\nabla\lambda(\theta)$, we see that the desired result has been proved. □

5.3 An Algorithm that Updates at Visits to the Recurrent State i^*

We now use the approximate gradient direction provided by Proposition 19, and propose a simulation-based algorithm that performs updates at visits to the recurrent state i^*. At the time t_m that state i^* is visited for the mth time, we have available a current vector θ_m and an average reward estimate $\tilde{\lambda}_m$. We then simulate the process according to the transition probabilities $P_{ij}(\theta_m)$ until the next time t_{m+1} that i^* is visited and update according to

$$\theta_{m+1} = \theta_m + \gamma_m F_m(\theta_m, \tilde{\lambda}_m), \tag{5.12}$$

$$\tilde{\lambda}_{m+1} = \tilde{\lambda}_m + \eta\gamma_m \sum_{n=t_m}^{t_{m+1}-1}(g_{i_n}(\theta_m) - \tilde{\lambda}_m), \tag{5.13}$$

where η is a positive scalar, and where γ_m is a step size parameter.

To see the rationale behind Eq.(5.13) note that the expected update direction for $\tilde{\lambda}$ is

$$E_\theta\left[\sum_{n=t_m}^{t_{m+1}-1}(g_{i_n}(\theta) - \tilde{\lambda})\right] = E_\theta[T](\lambda(\theta) - \tilde{\lambda}), \tag{5.14}$$

which moves $\tilde{\lambda}$ closer to $\lambda(\theta)$.

It can be shown that if θ is held fixed, but $\tilde{\lambda}$ keeps being updated according to Eq.(5.13), then $\tilde{\lambda}$ converges to $\lambda(\theta)$. However, if θ is also updated according to Eq.(5.12), then the estimate $\tilde{\lambda}_m$ can "lag behind" $\lambda(\theta_m)$. As a consequence, the expected update direction for θ will not be aligned with the gradient $\nabla\lambda(\theta)$.

An alternative approach that we do not pursue is to use different step sizes for updating $\tilde{\lambda}$ and θ. If the step size used to update θ is, in the limit, much smaller than the step size used to update $\tilde{\lambda}$, the algorithm exhibits a two-time scale behavior of the form studied in [Bor97]. In the limit, $\tilde{\lambda}_m$ is an increasingly accurate estimate of $\lambda(\theta_m)$, and the algorithm is effectively a stochastic gradient algorithm. However, such a method would make slower progress, as far as θ is concerned. Our convergence results indicate that this alternative approach is not necessary.

We can now state following convergence result.

Proposition 20 *Let Assumptions 5 (MRP Parameterization), 7 (MRP Recurrence), and 8 (Step Size 1) hold, and let (θ_m) be the sequence of parameter vectors generated by the above described algorithm. Then, $\lambda(\theta_m)$ converges and*

$$\lim_{m\to\infty} \nabla\lambda(\theta_m) = 0,$$

with probability 1.

We provide the proof of this proposition in Appendix C.1.

5.3.1 A Heuristic Argument

In this subsection, we approximate the algorithm by a suitable ODE (as in [Lju77]), and establish the convergence properties of the ODE. While this argument does not constitute a proof, it illustrates the rationale behind our convergence result.

We replace the update directions by their expectations under the current value of θ. The update equations for θ and $\tilde{\lambda}$ take the form

$$\begin{aligned}
\theta_{m+1} &= \theta_m + \gamma_m f(\theta_m, \tilde{\lambda}_m),\\
\tilde{\lambda}_{m+1} &= \tilde{\lambda}_m + \eta\gamma_m E_{\theta_m}[T](\lambda(\theta_m) - \tilde{\lambda}_m),
\end{aligned}$$

where $f(\theta, \tilde{\lambda})$ is given by Proposition 2. With an asymptotically vanishing step size, and after rescaling time, this deterministic iteration behaves similar to the following system of differential equations:

$$\dot{\theta}_t = \nabla\lambda(\theta_t) + \frac{G(\theta_t)}{E_{\theta_t}[T]}(\lambda(\theta_t) - \tilde{\lambda}_t), \tag{5.15}$$

$$\dot{\tilde{\lambda}}_t = \eta\left(\lambda(\theta_t) - \tilde{\lambda}_t\right). \tag{5.16}$$

Note that $\tilde{\lambda}_t$ and $\lambda(\theta_t)$ are bounded functions since the one-stage reward $g_i(\theta)$ is finite-valued and, therefore, bounded. We will now argue that $\tilde{\lambda}_t$ converges.

We first consider the case where the initial conditions satisfy $\tilde{\lambda}_0 \le \lambda(\theta_0)$. We then claim that

$$\tilde{\lambda}_t \le \lambda(\theta_t), \qquad \forall\ t > 0. \tag{5.17}$$

Indeed, suppose that at some time t_0 we have $\tilde{\lambda}_{t_0} = \lambda(\theta_{t_0})$. If $\nabla\lambda(\theta_{t_0}) = 0$, then we are at an equilibrium point of the differential equations, and we have $\tilde{\lambda}_t = \lambda(\theta_t)$ for all subsequent times. Otherwise, if $\nabla\lambda(\theta_{t_0}) \neq 0$, then $\dot{\theta}_{t_0} = \nabla\lambda(\theta_{t_0})$, and $\dot{\lambda}(\theta_{t_0}) > 0$. At the same time, we have $\dot{\tilde{\lambda}}_{t_0} = 0$, and this implies that $\tilde{\lambda}_t < \lambda(\theta_t)$ for t slightly larger than t_0. The validity of the claim (5.17) follows. As long as $\tilde{\lambda}_t \leq \lambda(\theta_t)$, $\tilde{\lambda}_t$ is nondecreasing and since it is bounded, it must converge.

Suppose now that the initial conditions satisfy $\tilde{\lambda}_0 > \lambda(\theta_0)$. As long as this condition remains true, $\tilde{\lambda}_t$ is non-increasing. There are two possibilities. If this condition remains true for all times, then $\tilde{\lambda}(t)$ converges. If not, then there exists a time t_0 such that $\tilde{\lambda}_{t_0} = \lambda(\theta_{t_0})$, which takes us back to the previously considered case.

Having concluded that $\tilde{\lambda}_t$ converges, we can use Eq.(5.16) to argue that $\lambda(\theta_t)$ must also converge to the same limit. Then, in the limit, θ_t evolves according to $\dot{\theta}_t = \nabla\lambda(\theta_t)$, from which it follows that $\nabla\lambda(\theta_t)$ must go to zero.

We now comment on the nature of a rigorous proof. There are two common approaches for proving the convergence of stochastic approximation methods. One method relies on the existence of a suitable Lyapunov function and a martingale argument. In our context, $\lambda(\theta)$ could play such a role. However, as long as $\tilde{\lambda}_m \neq \lambda(\theta_m)$, our method cannot be expressed as a stochastic gradient algorithm and this approach does not go through. (Furthermore, it is unclear whether another Lyapunov function would do.) The second proof method, the so-called ODE approach, shows that the trajectories followed by the algorithm converge to the trajectories of a corresponding deterministic ODE, e.g., the ODE given by Eqs.(5.15)-(5.16). This line of analysis generally requires the iterates to be bounded functions of time. In our case, such a boundedness property is not guaranteed to hold. For example, if θ stands for the weights of a neural network, it is certainly possible that certain "neurons" asymptotically saturate, and the corresponding weights drift to infinity. We conclude that we need a line of argument specially tailored to our particular algorithm. In rough terms, it proceeds along the same lines as the above provided deterministic analysis, except that we need to ensure that the stochastic terms are not significant.

5.4 Reducing the Variance

If the intervals between visits to the recurrent state i^* are typically large, then the variance of the updates in the algorithm proposed in the previous section may be quite large. Similar to Section 4.4 in Chapter 4, we consider in this section two approaches to reduce this variance: one where we replace the estimate $\tilde{v}_i(\theta, \tilde{\lambda})$ by

$$\tilde{v}_{S^*,i}(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} \Big(g_{i_k}(\theta) - \tilde{\lambda}\Big),$$

where S^* is a subset of the state space S containing i^*, and $T = \min\{k > 0 \mid i_k \in S^*\}$ is the first future time a state in the set S^* is visited, and one where we estimate the reward-to-go $v_i(\theta)$ of state i by

$$\tilde{v}_{\alpha,i}(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} \alpha^k \Big(g_{i_k}(\theta) - \tilde{\lambda}\Big),$$

where α, $0 \leq \alpha < 1$, is a discount factor, and where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time the state i^* is visited.

Again, when using the above approaches to reduce the variance, we are likely to introduce a bias into the estimate of the differential reward $v_i(\theta)$. As a consequence, when we use $\tilde{v}_{S^*,i}(\theta, \tilde{\lambda})$, or $\tilde{v}_{\alpha,i}(\theta, \tilde{\lambda})$, to produce an estimate of the gradient $\nabla\lambda(\theta)$, then we are likely to introduce a bias into this estimate of the gradient. We will characterize this bias, and identify situations for which the bias tends to be small.

5.4.1 A Modified Algorithm using $\tilde{v}_{S^*,i}(\theta, \tilde{\lambda})$

As mentioned above, if the time T until we visit the termination state i^* is large, then the variance of the estimate $\tilde{v}_i(\theta, \tilde{\lambda})$ of the previous section can become quite large. For this reason, it may be desirable to reduce the time T by choosing

$$T = \min\{k > 0 \mid i_k \in S^*\},$$

where S^* is a subset of S containing i^*.

In particular, given a set $S^* \subset S$ containing the recurrent i^*, and a trajectory $(i_0 = i, i_1, ...)$ of the Markov chain with transition probabilities $P_{ij}(\theta)$ which starts at state i, we then estimate the differential reward $v_i(\theta)$ of state i by

$$\tilde{v}_{S^*,i}(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} \Big(g_{i_k}(\theta) - \tilde{\lambda}\Big),$$

where $T = \min\{k > 0 \mid i_k \in S^*\}$ is the first future time that the set S^* is visited, and where $\tilde{\lambda}$ is an estimate of the average reward $\lambda(\theta)$. We can then derive the following bound on the variance of the estimate $\tilde{v}_{S^*,i}(\theta)$.

Proposition 21 *Let θ be an element of $\Re^K$, and let S^* be a subset of the state space S containing i^*. Furthermore, let Assumptions 5 (MRP Parameterization) and 7 (MRP Recurrence) hold. In particular, let $g_{\max}$ be a bound on $|g_i(\theta)|$, and let $\rho(\theta) > 0$ be such that*

$$P_\theta(T \leq N \mid i_0 = i) \geq \rho(\theta), \quad \textit{for all } i \in S,$$

where N is the number of states in S, and where $T = \min\{k > 0 \mid i_k \in S^*\}$ is the first future time a state in the set S^* is visited in a a trajectory that starts at state i. Then, for every state $i \in S$, we have

$$Var_\theta(\tilde{J}_{S^*,i}(\theta)) \leq (g_{\max} + |\tilde{\lambda}|)^2 \left(N^2 \frac{(2-\rho(\theta))}{\rho(\theta)^2} - 1\right).$$

Proof: Note that, for all states $i \in S \backslash \{i^*\}$ the estimate

$$\tilde{v}_i(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} \Big(g_{i_k}(\theta) - \tilde{\lambda}\Big),$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited, can interpreted as an estimate of the reward-to-go of state i in the Markov reward process on the state space S with transition probabilities

$$P_{i^*,ij}(\theta) = \begin{cases} P_{ij}(\theta) & \text{if } i \in S\backslash\{i^*\} \\ 1 & \text{if } i = j = i^* \\ 0 & \text{otherwise} \end{cases},$$

and immediate rewards

$$g_{i^*,i}(\theta) = \begin{cases} g_i(\theta) - \tilde{\lambda} & \text{if } i \in S\backslash\{i^*\} \\ 0 & \text{if } i = i^* \end{cases}$$

We use the notation $P_{i^*,\theta}(\cdot)$, to denote the probability distribution induced by the Markov chain with transition probabilities $P_{I^*,ij}(\theta)$. Note that the state i^* is a reward-free termination state in this new Markov reward process, and that, for all states $i \in S$, we have

$$P_{i^*,\theta}(T \leq N \mid i_0 = i) = P_\theta(T \leq N \mid i_0 = i) \geq \rho(\theta) > 0,$$

where N is the number of states in S, and where $T = \min\{k > 0 \mid i_k \in S^*\}$ is the first future time a state in the set S^* is visited in a a trajectory that starts at state i.

Therefore, the same argument as that in the proof of Proposition 6 in Section 4.4.1 holds, showing that

$$\mathrm{Var}_\theta(\tilde{J}_{S^*,i}(\theta)) \leq (g_{\max} + |\tilde{\lambda}|)^2 \left(N^2 \frac{(2-\rho(\theta))}{\rho(\theta)^2} - 1 \right).$$

□

Note that the bound on $\mathrm{Var}_\theta(\tilde{v}_{S^*,i}(\theta, \tilde{\lambda}))$ in the proposition above is decreasing in $\rho(\theta)$. Furthermore, note that $\rho(\theta)$ can be chosen to be nondecreasing in the size of the set S^*. It follows that the larger the set S^*, the smaller tends to be the bound on $\mathrm{Var}_\theta(\tilde{J}_{S^*,i}(\theta))$.

Estimation of $\nabla\lambda(\theta)$ using $\tilde{v}_{S^*,i}(\theta, \tilde{\lambda})$

In this section, we consider the estimate of the gradient $\nabla\lambda(\theta)$ produced by accumulating estimates $\tilde{v}_{S^*,i}(\theta, \tilde{\lambda})$ of the differential reward of states along a trajectory of the Markov chain with transition probabilities $P_{ij}(\theta)$. In particular, we are interested in characterizing the bias of this estimate.

Let S^* be a subset of S containing i^*, and let the parameter $\theta \in \Re^K$ be fixed to some value. As in Section 5.2, let $(i_1, i_2, ...)$ be a trajectory of the Markov chain with transition probabilities $P_{ij}(\theta)$, and let t_m be the time of the mth time to the termination state i^*. Furthermore, let $\kappa(m)$ be the number of times a state in the set $S^*\backslash\{i^*\}$ is visited, let the $t_{m,n}$ be the time when for the $n$the time in this interval a state in this set is visited, and let $t_{m,0}$ and $t_{m,\kappa(m)+1}$ be equal to t_m and t_{m+1}, respectively. Using these definitions, we obtain the following estimate $F_{S^*,m}(\theta, \tilde{\lambda})$ of the gradient $\nabla\lambda(\theta)$,

$$F_{S^*,m}(\theta, \tilde{\lambda}) = \sum_{k=t_m}^{t_{m+1}-1} \left(\tilde{v}_{S^*,i_k}(\theta, \tilde{\lambda}) \frac{\nabla P_{i_{k-1}i_k}(\theta)}{P_{i_{k-1}i_k}(\theta)} + \nabla g_{i_k}(\theta) \right), \tag{5.18}$$

where, for $t_{m,n} \leq k < t_{m,n+1}$, and $i_k \neq i^*$, we set

$$\tilde{v}_{S^*,i_k}(\theta, \tilde{\lambda}) = \sum_{l=k}^{t_{m,n+1}-1} \left(g_{i_l}(\theta) - \tilde{\lambda} \right),$$

and, for $k = t_m$, we set $\tilde{v}_{i_k}(\theta, \tilde{\lambda}) = 0$.

We have the following result for $f_{S^*}(\theta, \tilde{\lambda})$, which we define to be the expected value of $F_{S^*,m}(\theta, \tilde{\lambda})$, namely,

$$f_{S^*}(\theta, \tilde{\lambda}) = E_\theta[F_{S^*,m}(\theta, \tilde{\lambda})].$$

Proposition 22 *We have*

$$f_{S^*}(\theta, \tilde{\lambda}) = E_\theta[T] \sum_{i \in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta) v_{S^*,j}(\theta) \right) + G_{S^*}(\theta)(\lambda(\theta) - \tilde{\lambda}),$$

where

$$G_{S^*}(\theta) = E_\theta \left[\sum_{n=t_m+1}^{t_{m,1}-1} (t_{m,1} - n) \frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)} + \sum_{k=1}^{\kappa(m)} \sum_{n=t_{m,k}}^{t_{m,k+1}} (t_{m,k+1} - n) \frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)} \right],$$

and where

$$v_{S^*,j}(\theta) = E_\theta \left[\sum_{k=0}^{T-1} (g_{i_k}(\theta) - \lambda(\theta)) \mid i_0 = i \right],$$

with $T = \min\{k > 0 \mid i_k \in S^*\}$ being the first future time that the set S^* is visited.

Proposition 22 can be proven by the same argument as the one given for Proposition 19.

Note that the expression for $f_{S^*}(\theta)$ in Proposition 22 is of the same form as the expression for the gradient $\nabla \lambda(\theta)$ given in Section 5.1 multiplied with $E_\theta[T]$,

$$E_\theta[T] \nabla \lambda(\theta) = E_\theta[T] \sum_{i \in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta) v_j(\theta) \right),$$

except for the following two changes. There is an additional bias term $G_{S^*}(\theta)(\lambda(\theta) - \tilde{\lambda})$ and the exact value of the differential reward $v_j(\theta)$ is replaced by the approximation $v_{S^*,j}(\theta)$. Let us comment briefly on these two changes and the resulting biases. The first term will vanish when the estimate $\tilde{\lambda}$ approaches the true value of the average reward $\lambda(\theta)$. Replacing $v_j(\theta)$ by the approximation $v_{S^*,j}(\theta)$ introduces the bias

$$\begin{aligned} \sigma_{S^*}(\theta) &= E_\theta[T] \sum_{i \in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta) v_{S^*,j}(\theta) \right) - E_\theta[T] \nabla \lambda(\theta) \\ &= E_\theta[T] \sum_{i \in S} \pi_i(\theta) \left(\sum_{j \in S} \nabla P_{ij}(\theta) (v_{S^*,j}(\theta) - v_j(\theta)) \right), \end{aligned}$$

which we characterize in the next proposition. We will use the following notation. For every state $i \in S$, let $\hat{v}_{S^*,i}(\theta)$ be defined by

$$\hat{v}_{S^*,i}(\theta) = v_{S^*,i}(\theta) - v_i(\theta),$$

where $v_{S^*,i}(\theta)$ is defined as in Proposition 8. Also, as in the previous chapter, let the set S_i, $i \in S$ be defined by

$$S_i = \left\{ j \in S \mid \nabla P_{ij}(\theta) \neq 0 \text{ for some } \theta \in \Re^K \right\},$$

and let $\bar{N}$ be given by

$$\bar{N} = \max\{N_i \mid i \in S\},$$

where N_i is the number of states in the set S_i. $\bar{N}$ bounds the number of possible transitions from any state $i \in S$ that depend on θ.

Proposition 23 *Let Assumptions 5 (MRP Parameterization) and 7 (MRP Recurrence) hold, and let $T_{\max}$ be such that, for all $\theta \in \Re^K$, we have*

$$E_\theta[T] \leq T_{\max}.$$

and let ϵ be such that, for all states $i \in S$, we have

$$|\hat{v}_{S^*,j}(\theta) - \hat{v}_{S^*,j'}(\theta)| \leq \epsilon, \text{ if } j, j' \in S_i.$$

Then we have

$$\|\sigma_{S^*}(\theta)\| = \left\| E_\theta[T] \sum_{i \in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta)(v_{S^*,j}(\theta) - v_j(\theta)) \right) \right\| \leq T_{\max} \bar{N} C \epsilon,$$

where C is a bound on $\|\nabla P_{ij}(\theta)\|$.

In the proof of Proposition 21 we noted that $v_i(\theta)$ can be interpreted as the reward-to-go in a modified Markov reward process. Using this observation, Proposition 23 can be proved by the same argument as the one given to prove Proposition 9 in Section 4.5.1.

Proposition 23 states in order to keep the bias $\sigma_{S^*}(\theta)$ small, one should choose S^* such that, for all states $i \in S$ and for all states $j, j' \in S_i$, the difference $|\hat{v}_{S^*,j}(\theta) - \hat{v}_{S^*,j'}(\theta)|$ is small.

A Modified Algorithm using $F_{S^*,m}(\theta, \tilde{\lambda})$

In this section, we use the estimate $F_{S^*,m}(\theta)$ of the gradient $\nabla\lambda(\theta)$ to to formulate a simulation-based algorithm, which updates the parameter vector θ at visits to the state i^*. At the time t_m that state i^* is visited for the mth time, we have available a current vector θ_m. We then simulate the process according to the transition probabilities $P_{\bar{\pi},ij}(\theta_m)$ until the next time t_{m+1} that i^* is visited and update according to

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

where η is a positive scalar, and γ_k is a step size parameter.

We have the following result.

Proposition 24 *Let Assumptions 5 (MRP Parameterization), 7 (MRP Recurrence) and 8 (Step Size 1) hold, and let $T_{\min}$ and D be such that, for all $\theta \in \Re^K$, we have*

$$1 \le T_{\min} \le E_\theta[T],$$
$$\|\sigma_{S^*}(\theta)\| \le D,$$

where $\sigma_{S^*}(\theta)$ is as in Proposition 23. Furthermore, let (θ_m) be the sequence of parameter vectors generated by the above described algorithm. Then, with probability 1, we have

$$\liminf_{m\to\infty} \|\nabla\lambda(\theta_m)\| \le \frac{D}{T_{\min}}.$$

Note that a constant $T_{\min}$ such that, for all $\theta \in \Re^K$, we have $E_\theta[T] \ge T_{\min} > 0$ exists, as 1 is a lower bound of $E_\theta[T]$.

Proposition 24 establishes that if the bias $\|\sigma_m\|$ tends to be small, then the gradient $\nabla\chi(\theta_m)$ is small at infinitely many visits to the recurrent state i^*. We defer the proof of this proposition to Appendix C.2.

Proposition 24 does not say anything about the behavior of the sequence $\lambda(\theta_m)$. In a worst case, $\lambda(\theta_m)$ may oscillate significantly throughout the algorithm. However, in the next proposition, we show that if the bias $\|\sigma_{S^*}(\theta)\|$ is small enough, then $\tilde{\lambda}_m$ overestimates the average reward $\lambda(\theta_m)$ at most by a little. This result has an interesting application. It suggests that, while $\lambda(\theta_m)$ may oscillate significantly, the estimate $\tilde{\lambda}_m$ can be used to detect instances where the policy $\mu(\theta_m)$ leads to a good performance $\lambda(\theta_m)$. Also, this result suggests that, in this context, it is not generally true that $\tilde{\lambda}_m$ is an accurate estimate of $\lambda(\theta_m)$.

We will use the following lemma which gives a bound on $G_{S^*}(\theta)$.

Lemma 8 *Let Assumptions 5 (MRP Parameterization) and 7 (MRP Recurrence) hold, then there exists a constant L such that, for all $\theta \in \Re^K$, we have*

$$\|G_{S^*}(\theta)\| = \left\| E_\theta \left[\sum_{n=t_m+1}^{t_{m,1}-1} (t_{m,k+1} - n) \frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)} + \sum_{k=1}^{\kappa(m)} \sum_{n=t_{m,k}}^{t_{m,k+1}} (t_{m,k+1} - n) \frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)} \right] \right\| \le L.$$

Proof: Let C be a bound on $\|\frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)}\|$, which exists by Assumption 5 (MRP Parameterization). We then have

$$\begin{aligned} \|G_{S^*}(\theta)\| &= \left\| E_\theta \left[\sum_{n=t_m+1}^{t_{m,1}-1} (t_{m,k+1} - n) \frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)} + \sum_{k=1}^{\kappa(m)} \sum_{n=t_{m,k}}^{t_{m,k+1}} (t_{m,k+1} - n) \frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)} \right] \right\| \\ &\le C E_\theta \left[(t_{m+1} - t_m)^2 \right]. \end{aligned}$$

By Lemma 4, the expectation $E_\theta\left[(t_{m+1} - t_m)^2\right]$ is bounded and the result follows. $\square$

We then have the following result.

Proposition 25 *Let Assumptions 5 (MRP Parameterization), 7 (MRP Recurrence) and 8 (Step Size 1) hold, and let $T_{\min}$ L, and D be such that, for all $\theta \in \Re^K$, we have*

$$1 \le T_{\min} \le E_\theta[T],$$
$$\|G_{S^*}(\theta)\| \le L,$$
$$\|\sigma_{S^*}(\theta)\| \le D < \frac{\eta T_{\min}}{4L},$$

where $G_{S^*}(\theta)$, and $\sigma_{S^*}(\theta)$, are as in Lemma 8, and Proposition 23, respectively. Furthermore, let (θ_m) be the sequence of parameter vectors generated by the above described algorithm. Then, with probability 1, we have

$$\limsup_{m\to\infty}(\tilde{\lambda}_m - \lambda(\theta_m)) \le \frac{\eta T_{min}}{2L^2}\left[\left(1 - \frac{2LD}{\eta T_{min}}\right) - \sqrt{1 - \frac{4LD}{\eta T_{min}}}\right].$$

We defer the proof of this proposition to Appendix C.2.

5.4.2 A Modified Algorithm using $\tilde{v}_{\alpha,i}(\theta, \tilde{\lambda})$

In this section, we discuss a second approach to reduce the variance in the estimates of the differential reward. In particular, given a trajectory $(i_0 = i, i_1, \ldots)$ of the Markov chain with transition probabilities $P_{ij}(\theta)$ which starts at state i, we estimate the differential reward $v_i(\theta)$ by

$$\tilde{v}_{\alpha,i}(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} \alpha^k \Big(g_{i_k}(\theta) - \tilde{\lambda}\Big),$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that the state i^* is visited, and where α, $0 \le \alpha < 1$, is a discount factor. We have the following result for variance of the estimate $\tilde{v}_{\alpha,i}(\theta, \tilde{\lambda})$.

Proposition 26 *Let Assumptions 5 (MRP Parameterization) and 7 (MRP Recurrence) hold. In particular, let $g_{\max}$ be a bound on $|g_i(\theta)|$ and, for every $\theta \in \Re^K$, let $\rho(\theta) > 0$ be such that*

$$\sum_{n=1}^{N} P_{ii^*}^n(\theta) \ge \rho(\theta), \quad \text{for all } i \in S,$$

where N is the number of states in S. Then, for every $\theta \in \Re^K$ and every discount factor $\alpha \in [0, 1)$, we have

$$Var_\theta(\tilde{v}_{\alpha,i}(\theta, \tilde{\lambda})) \;\le\; \frac{(g_{\max} + |\tilde{\lambda}|)^2}{(1-\alpha)^2}\left(1 - \frac{2\alpha^N \rho(\theta)}{1 - (1-\rho(\theta))\alpha^N} + \frac{\alpha^{2N}\rho(\theta)}{1-(1-\rho(\theta))\alpha^{2N}}\right).$$

Proof: In the proof of Proposition 21 in Subsection 5.4.1, we noted that, for all states $i \in S\backslash\{i^*\}$ the estimate

$$\tilde{v}_i(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} \Big(g_{i_k}(\theta) - \tilde{\lambda}\Big),$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited, can be interpreted as an estimate of the reward-to-go of state i a new Markov reward process on the state space S. Furthermore, note that for this new Markov reward process with transition probabilities $P_{i^*,ij}(\theta)$ as given in the proof of Proposition 21, we have, for all states $i \in S$, that

$$P^N_{i^*,ii^*}(\theta) = \sum_{n=1}^{N} P^n_{ii^*}(\theta) \geq \rho(\theta) > 0,$$

where N is the number of states in the states space.

Therefore, the same argument as that proving Proposition 21 in Subsection 5.4.1 holds to show that

$$\mathrm{Var}_\theta(\tilde{J}_{S^*,i}(\theta)) \leq (g_{\max} + |\tilde{\lambda}|)^2 (N^2 \frac{(2-\rho(\theta))}{\rho(\theta)^2} - 1).$$

□

Note that in Proposition 26, the bound for the variance of the estimate $\tilde{v}_{\alpha,i}(\theta, \tilde{\lambda})$ is increasing in α and decreasing in $\rho(\theta)$. Furthermore, we have

$$\lim_{\rho(\theta)\to 1} \mathrm{Var}_\theta(\tilde{v}_{\alpha,i}(\theta, \tilde{\lambda}) \leq g_{\max} \left(\frac{1-\alpha^N}{1-\alpha} \right)^2,$$

which implies that when N, the number of states in the S, is large, that then the variance of $\tilde{J}_{\alpha,i}(\theta)$ tends to be large, unless α is small. Therefore, Proposition 26 suggests that in order to obtain a small variance in the estimate $\tilde{J}_{\alpha,i}(\theta)$, we have to choose a small discount factor α.

Estimation of $\nabla\lambda(\theta)$ using $\tilde{v}_{\alpha,i}(\theta, \tilde{\lambda})$

In this section, we produce an estimate of the gradient $\nabla\lambda(\theta)$, by accumulating estimates $\tilde{v}_{\alpha,i}(\theta)$ of the differential reward of states along a trajectory of the Markov chain with transition probabilities $P_{ij}(\theta)$.

Let scalar α be in $[0, 1)$, and let the parameter $\theta \in \Re^K$ be fixed to some value. Furthermore, let $(i_1, i_2, ...)$ be a sample path of the corresponding Markov chain with transition probabilities $P_{ij}(\theta)$. As in Section 5.2, let t_m be the time of the mth visit at the recurrent state i^*. Using these definitions, we obtain the following estimate $F_{\alpha,m}(\theta, \tilde{\lambda})$ of the gradient $\nabla\lambda(\theta)$,

$$F_{\alpha,m}(\theta, \tilde{\lambda}) = \sum_{n=t_m}^{t_{m+1}-1} \left(\tilde{v}_{i_n}(\theta, \tilde{\lambda}) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \nabla g_{i_n}(\theta) \right), \tag{5.19}$$

where, $t_m < n \leq t_{m+1} - 1$, we set

$$\tilde{v}_{\alpha,i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{t_{m+1}-1} \alpha^{k-n} \left(g_{i_k}(\theta) - \tilde{\lambda} \right),$$

and, for $n = t_m$, we set $\tilde{v}_{\alpha,i_n}(\theta, \tilde{\lambda}) = 0$.

We have the following result for $f_\alpha(\theta, \tilde{\lambda})$, which we define to be the expected value of $F_{\alpha,m}(\theta, \tilde{\lambda})$, namely,

$$f_\alpha(\theta, \tilde{\lambda}) = E_\theta[F_{\alpha,m}(\theta, \tilde{\lambda})].$$

Proposition 27 *We have*

$$f_\alpha(\theta,\tilde{\lambda}) = E_\theta[T]\sum_{i\in S}\pi_i(\theta)\left(\nabla g_i(\theta) + \sum_{j\in S}\nabla P_{ij}(\theta)v_{\alpha,j}(\theta)\right) + G_\alpha(\theta)(\lambda(\theta)-\tilde{\lambda}),$$

where

$$G_\alpha(\theta) = E_\theta\left[\sum_{n=t_m+1}^{t_{m+1}-1}\sum_{k=n}^{t_{m+1}-1}\alpha^{k-n}\frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)}\right],$$

and where

$$v_{\alpha,j}(\theta) = E_\theta\left[\sum_{k=0}^{T-1}\alpha^k(g_{i_k}(\theta)-\lambda(\theta)) \mid i_0 = i\right],$$

with $T = \min\{k>0 \mid i_k \in S^*\}$ *being the first future time that the state* i^* *is visited.*

The expression for the expected value of $F_{\alpha,m}(\theta,\tilde{\lambda})$ has the same form as the expected value of original estimate $F_m(\theta,\tilde{\lambda})$ of the gradient $\nabla\lambda(\theta)$ given in Proposition 19. However, the exact value of the differential reward $v_j(\theta)$ is now replaced by the approximation $v_{\alpha,j}(\theta)$. In the next proposition we characterize the resulting bias

$$\begin{aligned}\sigma_\alpha(\theta) &= E_\theta[T]\sum_{i\in S}\pi_i(\theta)\left(\nabla g_i(\theta) + \sum_{j\in S}\nabla P_{ij}(\theta)v_{\alpha,j}(\theta)\right) - E_\theta[T]\nabla\lambda(\theta)\\ &= E_\theta[T]\sum_{i\in S}\pi_i(\theta)\left(\nabla g_i(\theta) + \sum_{j\in S}\nabla P_{ij}(\theta)(v_{\alpha,j}(\theta)-v_j(\theta))\right).\end{aligned}$$

In order to do that, we use, as in Subsection 4.5.2 the notion of mixing constant on the set $S\backslash\{i^*\}$ for the Markov chain with transition probabilities

$$P_{i^*,ij}(\theta) = \begin{cases} P_{ij}(\theta), & \text{if } i\in S\backslash\{i^*\},\\ 1, & \text{if } i=j=i^*,\\ 0, & \text{otherwise.}\end{cases}$$

In particular, the scalar $\beta(\theta)$, $0\le\beta(\theta)<1$, is a mixing constant on the set $S\backslash\{i^*\}$ for the Markov chain with transition probabilities $P_{i^*,ij}(\theta)$, if there exists a constant $A(\theta)$, such that for all states $i,i'\in S\backslash\{i^*\}$, and all $n\ge 1$, we have

$$\sum_{j\in S}|P^n_{i^*,ij}(\theta) - P^n_{i^*,i'j}(\theta)| \le A(\theta)\beta(\theta)^n.$$

We have the following proposition.

Proposition 28 *Let Assumptions 5 (MRP Parameterization) and 7 (MRP Recurrence) hold, and let* $T_{\max}$ *be such that, for all* $\theta\in\Re^K$*, we have*

$$E_\theta[T]\le T_{\max}.$$

Furthermore, let the constants β, $0 \leq \beta < 1$, and A, be such that for all states $i, i' \in S \backslash \{i^*\}$, for all transition matrices $P \in \bar{\mathcal{P}}$ and all $n \geq 1$, we have

$$\sum_{j \in S} |P^n_{i^*,ij} - P^n_{i^*,i'j}| \leq A\beta^n.$$

Then we have

$$\begin{aligned} \|\sigma_\alpha(\theta)\| &= \left\| E_\theta[T] \sum_{i \in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta)(v_j(\theta) - v_{\alpha,j}(\theta)) \right) \right\| \\ &\leq T_{\max}(2 + A) C \bar{N} 2 g_{\max} \left(1 + \frac{\beta}{1-\beta} - \frac{\alpha\beta}{1-\alpha\beta} \right), \end{aligned}$$

where C is a bound on $\|\nabla P_{ij}(\theta)\|$, and $\bar{N}$ is the same bound on the number of of possible transitions from any state $i \in S$ that depend on θ, as used in Proposition 23.

Similar to Proposition 23, using the observation that $v_i(\theta)$ can be interpreted as the reward-to-go in a modified Markov Reward Process, Proposition 28 can be proved by the same argument as the one given to prove Proposition 11 in Subsection 4.5.2.

Proposition 28 states that if the mixing constant β is small, then we can choose a small discount factor α to reduce the variance in the estimate of the reward-to-go, without introducing a too large a bias $\|\sigma_\alpha(\theta)\|$.

A Modified Algorithm using $F_{\alpha,m}(\theta, \tilde{\lambda})$

Again, we can use the estimate $F_{\alpha,m}(\theta, \tilde{\lambda})$ of the gradient $\nabla\lambda(\theta)$ to formulate a simulation-based algorithm, which updates the parameter vector θ at visits to the state i^*. At the time t_m that state i^* is visited for the mth time, we have available a current vector θ_m. We then simulate the process according to the transition probabilities $P_{ij}(\theta_m)$ until the next time t_{m+1} that i^* is visited and update according to

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

where η is a positive scalar, and where γ_m is a step size parameter.

We have the following result.

Proposition 29 *Let Assumptions 5 (MRP Parameterization), 7 (MRP Recurrence) and 8 (Step Size 1) hold, and let $T_{\min}$ and D be such that, for all $\theta \in \Re^K$, we have*

$$\begin{aligned} &1 \leq T_{\min} \leq E_\theta[T], \\ &\|\sigma_\alpha(\theta)\| \leq D, \end{aligned}$$

where $\sigma_\alpha(\theta)$ is as in Proposition 28. Furthermore, let (θ_m) be the sequence of parameter vectors generated by the above described algorithm. Then, with probability 1, we have

$$\liminf_{m \to \infty} \|\nabla\lambda(\theta_m)\| \leq \frac{D}{T_{\min}}.$$

Proposition 29 establishes that if the bias term $\|\sigma_m\|$ tends to be small, then the gradient $\nabla\chi(\theta_m)$ is small at infinitely many visits to the recurrent state i^*. We defer the proof of this proposition to Appendix C.2.

Again, we show that if the bias $\|\sigma_\alpha(\theta)\|$ is small enough, then $\tilde{\lambda}_m$ overestimates the average reward $\lambda(\theta_m)$ at most by a little. We will use the following lemma, which provides a bound $G_\alpha(\theta)$.

Lemma 9 *Let Assumptions 5 (MRP Parameterization) and 7 (MRP Recurrence) hold, then there exists a constant L such that, for all $\theta \in \Re^K$, we have*

$$\|G_\alpha(\theta)\| = \left\| E_\theta \left[\sum_{n=t_m+1}^{t_{m+1}-1} \sum_{k=n}^{t_{m+1}-1} \alpha^{k-n} \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} \right] \right\| \leq L.$$

Proof: Let C be a bound on $\|\frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)}\|$, which exists by Assumption 5 (MRP Parameterization). We then have

$$\begin{aligned} \|G_\alpha(\theta)\| &= \left\| E_\theta \left[\sum_{n=t_m+1}^{t_{m+1}-1} \sum_{k=n}^{t_{m+1}-1} \alpha^{k-n} \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} \right] \right\| \\ &\leq C E_\theta \left[(t_{m+1} - t_m)^2 \right]. \end{aligned}$$

By Lemma 4, the expectation $E_\theta\left[(t_{m+1} - t_m)^2\right]$ is bounded and the result follows. $\square$

Proposition 30 *Let Assumptions 5 (MRP Parameterization), 7 (MRP Recurrence) and 8 (Step Size 1) hold, and let $T_{\min}$ L, and D be such that, for all $\theta \in \Re^K$, we have*

$$\begin{aligned} &1 \leq T_{\min} \leq E_\theta[T], \\ &\|G_\alpha(\theta)\| \leq L, \\ &\|\sigma_\alpha(\theta)\| \leq D < \frac{\eta T_{\min}}{4L}, \end{aligned}$$

where $G_\alpha(\theta)$, and $\sigma_\alpha(\theta)$, are as in Lemma 9, and Proposition 28, respectively. Furthermore, let (θ_m) be the sequence of parameter vectors generated by the above described algorithm. Then, with probability 1, we have

$$\limsup_{m\to\infty} (\tilde{\lambda}_m - \lambda(\theta_m)) \leq \frac{\eta T_{min}}{2L^2} \left[\left(1 - \frac{2LD}{\eta T_{min}}\right) - \sqrt{1 - \frac{4LD}{\eta T_{min}}} \right].$$

We defer the proof of this proposition to Appendix C.2.

5.5 An Algorithm that Updates at Every Time Step

In Section 5.2, we proposed a simulation-based algorithm which updates the parameter vector θ along the approximate gradient direction $F_m(\theta, \tilde{\lambda})$ at visits to the state i^*. Similar to Chapter 4, we can use the update direction $F_m(\theta, \tilde{\lambda})$ to derive an algorithm which updates the parameter vector at every time step.

We start out by we indicating an economical way of computing the direction $F_m(\theta, \tilde{\lambda})$ of update of the vector θ. In particular, taking into account that $\tilde{v}_{i_{t_m}}(\theta, \tilde{\lambda}) = 0$, Eq.(4.12) becomes

$$\begin{aligned} F_m(\theta, \tilde{\lambda}) &= \sum_{n=t_m+1}^{t_{m+1}-1} \tilde{v}_{i_n}(\theta, \tilde{\lambda}) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} + \sum_{n=t_m}^{t_{m+1}-1} \nabla g_{i_n}(\theta) \\ &= \nabla g_{i^*}(\theta) + \sum_{k=t_m+1}^{t_{m+1}-1} \Big(\nabla g_{i_k}(\theta) + (g_{i_k}(\theta) - \tilde{\lambda}) z_k \Big), \end{aligned}$$

where

$$z_k = \sum_{n=t_m+1}^{k} \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)}, \qquad k = t_m + 1, \ldots, t_{m+1} - 1,$$

is a vector (of the same dimension as θ) that becomes available at time k. It can be updated recursively, with

$$z_{t_m} = 0, \tag{5.20}$$

and

$$z_{k+1} = z_k + \frac{\nabla P_{i_k i_{k+1}}(\theta)}{P_{i_k i_{k+1}}(\theta)}, \qquad k = t_m, \ldots, t_{m+1} - 2. \tag{5.21}$$

Using this result, we obtain the following algorithm. At a typical time k, the state is i_k, and the values of θ_k, z_k, and $\tilde{\lambda}_k$ are available from the previous iteration. We update θ and $\tilde{\lambda}$ according to

$$\begin{aligned} \theta_{k+1} &= \theta_k + \gamma_k \Big(\nabla g_{i_k}(\theta_k) + (g_{i_k}(\theta_k) - \tilde{\lambda}_k) z_k \Big), \\ \tilde{\lambda}_{k+1} &= \tilde{\lambda}_k + \eta \gamma_k (g_{i_k}(\theta_k) - \tilde{\lambda}_k), \end{aligned}$$

where η is a positive scalar, and γ_k is a step size parameter.

We then simulate a transition to the next state i_{k+1} according to the transition probabilities $P_{ij}(\theta_{k+1})$, and update z by letting

$$z_{k+1} = \begin{cases} 0, & \text{if } i_{k+1} = i^*, \\ z_k + \dfrac{\nabla P_{i_k i_{k+1}}(\theta_k)}{P_{i_k i_{k+1}}(\theta_k)}, & \text{otherwise.} \end{cases}$$

Note that this algorithm can be implemented in an online fashion.

Similar to Chapter 4, to prove convergence of this algorithm, we use the following assumption.

Assumption 11 (MRP Strong Recurrence) *There exist a state $i^* \in S$ and a positive integer N_0, such that, for every state $i \in S$ and every collection $\{P_1, ..., P_{N_0}\}$ of N_0 matrices in the set $\overline{\mathcal{P}}$, we have*

$$\sum_{n=1}^{N_0} Q_{ii^*}^n > 0,$$

where the matrix Q^n is given by

$$Q^n = \Pi_{l=1}^n P_l.$$

By a similar argument as the one given to prove Lemma 7, we can show that Assumption 11 holds, if there exist a scalar $\epsilon > 0$, such that, for every element $P \in \overline{\mathcal{P}}$ and for every state $i \in S$, we have

$$P_{ij} \geq \epsilon > 0, \qquad \text{for all } j \in S_i,$$

and a matrix $P \in \overline{\mathcal{P}}$, such that for every state $i \in S$ we have

$$\sum_{n=1}^{N} P_{ii^*}^n > 0,$$

where N is the number of states in the state space S. The set S_i, $i \in S$, is given by

$$S_i = \{j \in S \mid \nabla P_{ij}(\theta) \neq 0 \text{ for some } \theta \in \Re^K\}.$$

We have the following convergence result.

Proposition 31 *Let Assumptions 5 (MRP Parameterization), 8 (Step Size 1), 9 (Step Size 2), and 11 (MRP Strong Recurrence) hold, and let (θ_k) be the sequence of parameter vectors generated by the above described algorithm. Then, $\lambda(\theta_k)$ converges and*

$$\lim_{k \to \infty} \nabla \lambda(\theta_k) = 0,$$

with probability 1.

The algorithm of this section is similar to the algorithm of Section 5.2, except that θ is continually updated in the course of a renewal cycle. Because of the diminishing step size, these incremental updates are asymptotically negligible and the difference between the two algorithms is inconsequential. Given that the algorithm of the preceding section converges, Proposition 31 is hardly surprising. In particular, this proposition can be proven, using Proposition 20, by the same argument that we gave to prove Proposition 14 in Chapter 4.

Also, similar to Chapter 4, we can derive algorithms which update the parameter vector at each time step, based on the estimates $F_{S^*,m}(\theta)$, and $F_{\alpha,m}(\theta)$, of Section 5.4.1, and Section 5.4.2, respectively. Again, for these two cases, the results for the corresponding algorithms given in Section 5.4.1, and Section 5.4.2, respectively, remain valid in this context.

5.6 Closing Remarks

In this chapter, we relate the results obtained in this chapter to other available work.

At the heart of this chapter lies the expression

$$\nabla\lambda(\theta) = \sum_{i\in S}\pi_i(\theta)\left(\nabla g_i(\theta) + \sum_{j\in S}\nabla P_{ij}(\theta)v_j(\theta)\right) \tag{5.22}$$

for the gradient of the average reward. Eq.(5.22) has been presented, and independently derived, in various contexts: sensitivity analysis of Markov systems [CC97], likelihood-ratio method [Gly86, Gly87], neuro-dynamic programming/reinforcement learning [JSJ95], and smoothed perturbation analysis [FH94]. While these methods are quite different, in the sense that each is suited for problems that the others can not address, they overlap in the area of parameterized Markov reward processes, where they all produce the same result.

Based on Eq.(5.22), various estimators of $\nabla\lambda(\theta)$ have been proposed. Roughly, there are two main approaches, for which Eq.(5.22) is used to produce

(a) an unbiased estimate $F(\theta)$ based on observations of two renewal cycles.

(b) a biased estimate $F_n(\theta)$ based on a trajectory $(i_0, i_1, ..., i_n)$ of length n. Typically $F_n(\theta)$ is strongly consistent, i.e. we have, with probability 1,

$$\lim_{n\to\infty} F_n(\theta) = \nabla\lambda(\theta).$$

Let us discuss these two approaches in some more detail.

In [Gly86], an unbiased estimate of $\nabla\lambda(\theta)$ is produced by simulating two recurrent cycles. To illustrate the basic idea, let us consider the following situation. Assume that we use the first renewal cycle $(i_0, ..., i_{T^{(1)}-1})$ of length $T^{(1)}$ to obtain an unbiased estimate of the average reward by

$$\tilde{\lambda} = \frac{1}{T^{(1)}}\sum_{k=0}^{T^{(1)}-1} g_{i_k}(\theta).$$

Using $\tilde{\lambda}$ in the second renewal cycle $(i_{T^{(1)}}, ..., i_{T^{(1)}+T^{(2)}-1})$ of length $T^{(2)}$, for $i_n = i$, the expression

$$\tilde{\lambda}v_{i_n}(\theta,\tilde{\lambda}) = \sum_{k=n}^{T^{(1)}+T^{(2)}-1}(g_{i_k}(\theta) - \tilde{\lambda}), \qquad n = T^{(1)}+1, ..., T^{(1)}+T^{(2)}-1,$$

then yields an unbiased estimate of $v_i(\theta)$. This approach translates into an algorithm where θ gets updated after the completion of two renewal cycles. Convergence can easily be proven using standard techniques of stochastic approximation as shown in [Gly86]. However, it is unclear if this approach lends itself to derive an algorithm which updates the parameters θ, and $\tilde{\lambda}$ at each time step.

Various forms of an estimator of type (b) have been proposed [CW98, FH94, JSJ95, Gly87]. Typically, using a trajectory $(i_0, ..., i_n)$ of the Markov chain with transition probabilities $P_{ij}(\theta)$, the estimate $F_n(\theta)$ of the gradient $\nabla\lambda(\theta)$ is obtained by setting

$$F_n(\theta) = \sum_{i\in S}\tilde{\pi}_i(\theta)\nabla P_{ij}(\theta)\tilde{v}_i(\theta,\tilde{\lambda}),$$

where

$$\tilde{\pi}_i(\theta) = \frac{\kappa_i}{n},$$
$$\tilde{\lambda} = \frac{1}{n}\sum_{k=1}^{n} g_{i_k}(\theta).$$

Here κ_i is the number of times the state i is visited in the trajectory, and $\tilde{v}_i(\theta, \tilde{\lambda})$ is of the form

$$\tilde{v}_i(\theta, \tilde{\lambda}) = \frac{1}{\kappa_i}\sum_{l=1}^{\kappa_i} \tilde{v}_{i,l}(\theta, \tilde{\lambda}),$$

where the estimate $v_{i,l}(\theta, \tilde{\lambda})$ is obtained using observation after the l visit to state i. Proposed expressions for $\tilde{v}_{i,l}(\theta, \tilde{\lambda})$ include

$$v_{i,l}(\theta, \tilde{\lambda}) = \sum_{k=t_{i,l}}^{n} (g_{i_k}(\theta) - \tilde{\lambda}),$$
$$v_{i,l}(\theta, \tilde{\lambda}) = \sum_{k=t_{i,l}}^{n} g_{i_k}(\theta),$$
$$v_{i,l}(\theta, \tilde{\lambda}) = \sum_{k=t_{i,l}}^{T_{i,l}-1} (g_{i_k}(\theta) - \tilde{\lambda}),$$

where $t_{i,l}$ is the time of the lth visit to state i, and $T_{i,l} = \min\{k > t_{i,l} \mid i_k = i^*\}$ is the first future time after $t_{i,k}$ that state i^* is visited. We make the following observations for this approach.

(1) The estimator $F_n(\theta)$ is biased for finite n.

(2) To obtain $F_n(\theta)$, an estimate $\tilde{v}_i(\theta, \tilde{\lambda})$ has to be computed and stored for each state i, which is impossible for problems involving a large state space.

(3) The corresponding gradient algorithm updates the parameter vector after every n iteration steps. Again, for problems involving a large state space, n may have to be large to obtain a sufficiently small bias, which leads to infrequent updates of θ.

(1) may not be a major concern, as gradient algorithms are typically robust with respect to a small bias. To avoid (2) variations of the above estimator have been suggested in [CW98, Gly87] which are in the spirit of our derivations in Section 5.2. In particular, [Gly87] contains the same expression for $F_n(\theta)$ as the one given by Eq.(5.3) in Section 5.2. Interestingly, of the aforementioned papers only [JSJ95] addressed (3), and proposed an algorithm that updates at each time step an estimate of the differential reward vector, which is then used to update θ.

Chapter 5 draws on many ideas presented in the aforementioned papers. We share with [Gly86] the approach of using renewal cycles as a basis for our analysis. The form of the estimate $F_m(\theta, \tilde{\lambda})$ that we used in Section 5.1, and Section 5.2 has been presented in [Gly87]. The motivation for identifying simple iteration rules that update θ at each iteration step comes from the methodology of neuro-dynamic programming/reinforcement Learning, which is the approach used in [JSJ95]. The theory developed in Chapter 5 extends these earlier work in the following way. We showed that in order to obtain convergence, one does not have to rely on an unbiased estimate of the gradient, and estimators $F_n(\theta)$, which are based on a trajectory of length n, can be used to derive algorithms that update the parameter vector at each time step. In this sense, a main contribution of Chapter 5 is to combine ideas of earlier, and to obtain an algorithm which

1. updates the parameter vector θ at each time step.

2. converges, i.e., with probability 1, we have that $\lambda(\theta_k)$ converges and

$$\lim_{k\to\infty} \|\nabla\lambda(\theta_k)\| = 0.$$

Chapter 6

Markov Decision Processes

In this chapter, we apply the theory of parameter Markov reward processes that we developed in Chapter 4 and 5 to Markov decision processes. In particular we consider Markov decision processes that involve a randomized control policy which is described in terms of a parameter vector. As that case initial motivated us to study parameterized Markov reward processes, this chapter brings the theoretical part of the thesis to a closure.

Recall how in Chapter 3 we considered Markov decision processes that are defined on a finite state space $S = \{1, \ldots, N\}$ and a finite action space $U = \{1, \ldots, L\}$. At any state i, the choice of a control action $u \in U$ determines the transition probabilities $P_{ij}(u)$, and the one-stage rewards $g_i(u)$. Furthermore, we introduced randomized control policies that are described in terms of a parameter vector by associating with each parameter vector $\theta \in \Re^K$ a mapping

$$\mu(\theta) : S \mapsto [0,1]^L.$$

A randomized control policy $\mu(\theta)$ specifies, for any given state i, the probability $\mu_u(i,\theta)$ that control action u is applied at state i. The corresponding transition probabilities are given by

$$P_{ij}(\theta) = \sum_{u \in U} \mu_u(i,\theta) P_{ij}(u), \tag{6.1}$$

and the expected reward per stage is given by

$$g_i(\theta) = \sum_{u \in U} \mu_u(i,\theta) g_i(u). \tag{6.2}$$

In this context we can restate Assumption 5 (MRP Parameterization) that we introduced in Chapter 3 for Markov reward processes as follows.

Assumption 12 (MDP Parameterization) *For every state $i \in S$, and every control action $u \in U$, $\mu_u(i,\theta)$ is, as a function of θ, bounded and has bounded first and second derivatives. Furthermore, for every state $i \in S$, and every control action $u \in U_i$, we have*

$$\nabla \mu_u(i,\theta) = \mu_u(i,\theta) f_u(i,\theta),$$

where the function $f_u(i,\theta)$ is bounded, differentiable and has bounded first derivatives.

Let us comment on Assumption 12 and how it relates to Assumption 5 (MRP Parameterization). It can easily be verified by differentiating Eqs.(6.1), and (6.2) that Assumption 12 implies that, for all states $i, j \in S$, the transition probabilities $P_{ij}(\theta)$ and the one-stage reward $g_i(\theta)$ are, as functions of θ, bounded, and have bounded first and second derivatives. Furthermore, Assumption 12 implies, that, for all states $i, j \in S$, we have

$$\nabla P_{ij}(\theta) = P_{ij}(\theta) f_{ij}(\theta),$$

where the function

$$f_{ij}(\theta) = \frac{\sum_{u \in U} \nabla \mu_u(i,\theta) P_{ij}(u)}{\sum_{u \in U} \mu_u(i,\theta) P_{ij}(u)}$$

is bounded, because the functions

$$f_u(i,\theta) = \frac{\nabla \mu_u(i,\theta)}{\mu_u(i,\theta)}, \qquad \text{for } i \in S, \text{ and } u \in U,$$

are bounded. Note however that Assumption 12 does not imply that the functions $f_{ij}(\theta)$ of Assumption 5 (MRP Parameterization) have bounded first derivatives. It will become clear in Section 6.1 why Assumption 12 is sufficient for the context of Markov decision processes. Let us consider the following examples.

Example 8 *We consider the call admission control problem that we introduced in Section 2.4, where a link provider sells bandwidth to customers of different service types so as to maximize the long term revenue. In particular, we consider the randomized control policy given in Example 1 in Chapter 3, which is described in terms of a parameter vector $\theta \in \Re^M$, where M is the number of service types supported by the link. Recall that in this example the link provider accepts at state $i = (s, \omega)$ a new call, i.e., chooses the control action u_a, with probability*

$$\mu_{u_a}(i,\theta) = \frac{1}{1 + \exp(s \cdot b - \theta_m)}, \qquad \textit{if } \omega = e_m,$$

and decides to reject a new call, i.e., chooses the control action u_r, with probability

$$\mu_{u_r}(i,\theta) = 1 - \mu_{u_a}(i,\theta).$$

where s denotes the current configuration of the link, e_m denotes the m-th unit vector, $\omega = e_m$ indicates an arrival of a new call of service type m and $b = (b(1), ..., b(M))$ is a vector that specifies the bandwidth demands of the different service types.

It is then straightforward to verify that the functions $\mu_{u_a}(i,\theta)$ and $\mu_{u_r}(i,\theta)$ satisfy Assumption 12 .

Example 9 *More generally, let $r_u(i,\theta)$, $\theta \in \Re^K$, be a parameterized function that maps states and actions to real numbers, such that for, every state $i \in S$ and every control action $u \in U$, $r_u(i,\theta)$ has bounded first and second derivatives. Furthermore, consider the case where the decision maker chooses at state i control action u with probability*

$$\mu_u(i,\theta) := \frac{\exp\left(r_u(i,\theta)\right)}{\sum\limits_{u' \in U} \exp\left(r_{u'}(i,\theta)\right)}.$$

Obviously, we have $\mu_u(i,\theta) \geq 0$, and $\sum_{u \in U} \mu_u(i,\theta) = 1$. In addition, it is easily verified that the functions $\mu_u(i,\theta)$ satisfies Assumption 12 .

Example 9 illustrates that Assumption 12 is not restrictive, in the sense that there is a rich class of functions for which Assumption 12 holds.

We introduce the following notation. For every parameter vector $\theta \in \Re^K$, we define $P(\theta)$ to be the transition matrix with entries $P_{ij}(\theta)$ as given by Eq.(6.1). Furthermore, let $\mathcal{P} = \{P(\theta) \mid \theta \in \Re^K\}$ be the set of all possible transition matrices, and let $\bar{\mathcal{P}}$ be its closure.

In the remainder of this section, we restate the results given in Chapter 5 for the case where a decision maker tunes the parameter of the policy $\mu(\theta)$ to optimize the average reward $\lambda(\theta)$. The same program that we use here can also be applied to the case where the decision maker optimizes the weighted reward $\chi(\theta)$.

6.1 Optimizing the Average Reward

In this section, we study Markov decision process which satisfy Assumption 4 (MDP Recurrence) that we introduced in Section 2.2, i.e., there exists a state i^* such that, for every pure policy $\mu : S \mapsto U$ and for every state $i \in S$, we have

$$\sum_{n=1}^{N} P_{ii^*}^n(\mu) > 0,$$

where N is the number of states in the state space S.

When Assumption 4 (MDP Recurrence) holds, then there exists a pure control policy μ^* such that, for every randomized control policy, and therefore for every element $P \in \bar{\mathcal{P}}$, we have

$$\sum_{n=1}^{N} P_{ii^*}^n \geq \sum_{n=1}^{N} P_{ii^*}^n(\mu^*) > 0, \qquad i \in S.$$

This is exactly the content of Assumption 7 (MRP Recurrence) that we introduced in Chapter 3 for parameterized Markov reward processes. It follows that, in this context, Assumption 4 (MDP Recurrence) implies Assumption 7 (MRP Recurrence), and therefore, under Assumption 4, the steady state probabilities $\pi_i(\theta)$, and the average reward

$$\lambda(\theta) = \sum_{i \in S} \pi_i(\theta) g_i(\theta) = \sum_{i \in S} \pi_i(\theta) \sum_{u \in U} \mu_u(i,\theta) g_i(u),$$

are well defined for every randomized policy $\mu(\theta)$. Recall how we defined in Chapter 3 the differential reward vector $v(\theta)$ and mean-recurrence time $E_\theta[T]$ for parameterized Markov reward processes. Note that when Assumption 4 holds, then Lemma 4 applies, i.e., $\pi(\theta)$, $\lambda(\theta)$, $E_\theta[T]$ and $v(\theta)$ are, as functions of θ, bounded, twice differentiable, and have bounded first and second derivatives.

6.1.1 The Gradient $\nabla\lambda(\theta)$

In this subsection, we restate the expression for the gradient $\lambda(\theta)$ given in Chapter 5 in the context of Markov Decision Processes.

Let us start with a few observations. Since $\sum_{u\in U}\mu_u(i,\theta) = 1$, for every θ, we have $\sum_{u\in U}\nabla\mu_u(i,\theta) = 0$, and

$$\nabla g_i(\theta) = \sum_{u\in U}\nabla\mu_u(i,\theta)\left(g_i(u) - \lambda(\theta)\right).$$

Furthermore, we have

$$\sum_{j\in S}\nabla P_{ij}(\theta)v_j(\theta) = \sum_{j\in S}\sum_{u\in U}\nabla\mu_u(i,\theta)P_{ij}(u)v_j(\theta).$$

Using these relations in the formula for $\nabla\lambda(\theta)$ provided by Proposition 17, we obtain

$$\begin{aligned}
\nabla\lambda(\theta) &= \sum_{i\in S}\pi_i(\theta)\left(\nabla g_i(\theta) + \sum_{j\in S}\nabla P_{ij}(\theta)v_j(\theta)\right) \\
&= \sum_{i\in S}\pi_i(\theta)\left(\sum_{u\in U}\nabla\mu_u(i,\theta)g_i(u) + \sum_{j\in S}\sum_{u\in U}\nabla\mu_u(i,\theta)P_{ij}(u)v_j(\theta)\right) \\
&= \sum_{i\in S}\pi_i(\theta)\sum_{u\in U}\nabla\mu_u(i,\theta)\left((g_i(u)-\lambda(\theta)) + P_{ij}(u)v_j(\theta)\right) \\
&= \sum_{i\in S}\sum_{u\in U}\pi_i(\theta)\nabla\mu_u(i,\theta)q_{i,u}(\theta),
\end{aligned}$$

where

$$\begin{aligned}
q_{i,u}(\theta) &= (g_i(u) - \lambda(\theta)) + \sum_{j\in S}P_{ij}(u)v_j(\theta) \\
&= E_\theta\left[\sum_{k=0}^{T-1}(g_{i_k}(u_k) - \lambda(\theta)) \mid i_0 = i, u_0 = u\right],
\end{aligned}$$

and where i_k and u_k is the state and control at time k, and $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that the recurrent state i^* is visited. Thus, $q_{i,u}(\theta)$ is the differential reward if control action u is first applied in state i, and policy $\mu(\theta)$ is followed thereafter. It is the same as Watkins' Q-factor [Wat89], suitably modified for the average reward case. Note that while the differential reward $v_{i^*}(\theta)$ is equal to 0 for all $\theta \in \Re^K$, it is generally not true that $q_{i^*,u}(\theta) = 0$, $u \in U$.

6.1.2 An Algorithm that Updates at Visits to the Recurrent State i^*

Using the expression for the gradient $\nabla\lambda(\theta)$ given in the previous subsection, we can formulate a simulation-based algorithm which updates the parameter vector at visits to the recurrent state i^*.

We start by rewriting the expression for the gradient $\nabla\lambda(\theta)$ as

$$\nabla\lambda(\theta) = \sum_{i\in S}\sum_{u\in U_i(\theta)} \pi_i(\theta)\mu_u(i,\theta)q_{i,u}(\theta)\frac{\nabla\mu_u(i,\theta)}{\mu_u(i,\theta)},$$

where the set $U_i(\theta)$ is defined by

$$U_i(\theta) = \{u \in U \mid \nabla\mu_u(i,\theta) \neq 0\}.$$

Assumption 12 (MDP Parameterization) then ensures that the quotient $\frac{\nabla\mu_u(i,\theta)}{\mu_u(i,\theta)}$ in the above expression is well defined.

We can now proceed as in Chapter 5 and obtain an algorithm that updates θ at visits to the state i^*. At the time T_{m+1} of the $(m+1)$th visit to the state i^*, we update θ and $\tilde{\lambda}$ according to is

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

where

$$F_m(\theta_m, \tilde{\lambda}_m) = \sum_{n=t_m}^{t_{m+1}-1} \tilde{q}_{i_n,u_n}(\theta_m, \tilde{\lambda}_m)\frac{\nabla\mu_{u_n}(i_n,\theta_m)}{\mu_{u_n}(i_n,\theta_m)},$$

$$\tilde{q}_{i_n,u_n}(\theta_m, \tilde{\lambda}_m) = \sum_{k=n}^{t_{m+1}-1} (g_{i_k}(u_k) - \tilde{\lambda}_m),$$

and where η is a positive scalar, and γ_m is a step size parameter.

We have the following convergence result.

Proposition 32 *Let Assumptions 4 (MDP Recurrence), 8 (Step Size 1), and 12 (MDP Parameterization) hold, and let (θ_m) be the sequence of parameter vectors generated by the above described algorithm. Then, $\lambda(\theta_m)$ converges and*

$$\lim_{m\to\infty} \nabla\lambda(\theta_m) = 0,$$

with probability 1.

The argument given to prove Proposition 20 in Section 5 remains valid to establish Proposition 32, with only notation changes in the proof.

Similar to Chapter 5, as an approach to reduce the variance of the updates, modified versions of the above mentioned algorithm can be formulated. In particular, given a trajectory $((i_0, u_0), (i_1, u_1), ...)$ for which $i_0 = i$ and $u_0 = u$, we can estimate the function $q_{iu}(\theta)$ by

$$\tilde{q}_{S^*,i,u}(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} (g_{i_k}(u_k) - \tilde{\lambda}_m),$$

where S^* is a set containing the recurrent state i^* and $T = \min\{k > 0 \mid i_k \in S^*\}$ is the first future time that a state in the set S^* is visited, and by

$$\tilde{q}_{\alpha,i,u}(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} \alpha^k (g_{i_k}(u_k) - \tilde{\lambda}_m),$$

where α, $0 \leq \alpha < 1$, is a discount factor and $T = \min\{k > 0 \mid i_k = i^*\}$, is the first future time that the recurrent state i^* is visited. Based on these estimates, we can proceed as in Chapter 5 and formulate modified versions of the algorithm mentioned above. Also, the corresponding results given in Sections 5.4.1 and 5.4.2 still apply here, with only notational changes in the proofs.

6.1.3 An Algorithm that Updates at Every Time Step

In this subsection, we derive a version of the above mentioned algorithm where we now update the parameter vector at each time-step. In particular, by rewriting the estimate $F_m(\theta_m, \tilde{\lambda}_m)$ as

$$\begin{aligned}
F_m(\theta_m, \tilde{\lambda}_m) &= \sum_{n=t_m}^{t_{m+1}-1} \tilde{q}_{i_n,u_n}(\theta_m, \tilde{\lambda}_m) \frac{\nabla \mu_{u_n}(i_n, \theta_m)}{\mu_{u_n}(i_n, \theta_m)} \\
&= \sum_{n=t_m}^{t_{m+1}-1} \frac{\nabla \mu_{u_n}(i_n, \theta_m)}{\mu_{u_n}(i_n, \theta_m)} \sum_{k=n}^{t_{m+1}-1} (g_{i_k}(u_k) - \tilde{\lambda}_m) \\
&= \sum_{k=t_m}^{t_{m+1}-1} (g_{i_k}(u_k) - \tilde{\lambda}_m) z_{k+1},
\end{aligned}$$

where

$$z_{k+1} = \begin{cases} \frac{\nabla \mu_{u_k}(i_{k+1}, \theta_m)}{\mu_{u_k}(i_k, \theta_m)}, & \text{if } k = t_m, \\ z_k + \frac{\nabla \mu_{u_k}(i_k, \theta_m)}{\mu_{u_k}(i_n, \theta_m)}, & k = t_m + 1, ..., t_{m+1} - 1 \end{cases}$$

we obtain the following algorithm. At a typical time k, the state is i_k, and the values of θ_k, z_k, and $\tilde{\lambda}_k$ are available from the previous iteration. We then generate a control action $u_k \in U$ using the randomized policy $\mu(\theta_k)$, and the next state i_{k+1} according to the transition probabilities $P_{ij}(u_k)$. Finally, we update z_k, θ_k and $\tilde{\lambda}_k$ according to

$$\begin{aligned}
z_{k+1} &= \begin{cases} \frac{\nabla \mu_{u_k}(i_k, \theta_k)}{\mu_{u_k}(i_k, \theta_k)}, & \text{if } i_k = i^*, \\ z_k + \frac{\nabla \mu_{u_k}(i_k, \theta_k)}{\mu_{u_k}(i_k, \theta_k)}, & \text{otherwise,} \end{cases} \\
\theta_{k+1} &= \theta_k + \gamma_k (g_{i_k}(\theta_k) - \tilde{\lambda}_k) z_{k+1}, \\
\tilde{\lambda}_{k+1} &= \tilde{\lambda}_k + \eta \gamma_k (g_{i_k}(\theta_k) - \tilde{\lambda}_k).
\end{aligned}$$

Note that, unlike in Chapter 5, here we do not need to strengthen Assumption 4 (MDP Recurrence) to prove convergence of this algorithm. When Assumption 4 holds, then there exists a pure control policy $\mu^* : S \mapsto U$ such that, for every randomized non stationary control policy, i.e., every policy for which the probability of applying control $u \in U$ at state i_n can depend on i_n and the time n, and therefore for every collection $\{P_1, ..., P_{N_0}\}$ of N matrices in the set $\overline{\mathcal{P}}$, we have

$$\sum_{n=1}^{N} Q^n_{ii^*} \geq \sum_{n=1}^{N} P^n_{ii^*}(\mu^*) > 0, \qquad i \in S,$$

where the matrix Q^n is given by

$$Q^n = \Pi_{l=1}^{n} P_l.$$

Therefore, Assumption 4 (MDP Recurrence) implies Assumption 11 (MRP Strong Recurrence) in this context.

We have the following convergence result.

Proposition 33 *Let Assumptions 4 (MDP Recurrence), 8 (Step Size 1), 9 (Step Size 2), and 12 (MDP Parameterization) hold, and let (θ_m) be the sequence of parameter vectors generated by the above described algorithm. Then, $\lambda(\theta_m)$ converges and*

$$\lim_{m \to \infty} \nabla \lambda(\theta_m) = 0,$$

with probability 1.

The argument given to prove Proposition 15 in Chapter 4 remains valid to establish Proposition 33, with only notational changes in the proof.

Also, similar to Chapter 5, we can formulate an algorithm which updates the parameter vector at each time step, based on the estimates $\tilde{q}_{S^*,iu}(\theta, \tilde{\lambda})$, and $\tilde{q}_{\alpha,iu}(\theta, \tilde{\lambda})$, respectively, given above. We omit a detailed derivation here. However, we will consider these algorithms in the next chapter, where we present a computational study.

Chapter 7

Call Admission Control

In this chapter, we illustrate the algorithms that we proposed in Chapter 6 by considering the call admission control problem introduced in Section 2.4. We shall study a small instance of this problem for which it is possible to compute an optimal call admission control policy using methods of dynamic programming, and to compute exactly the gradient of the average reward. This will allow us to compare the control policies obtained by the simulation-based gradient algorithms with the one obtained by Dynamic Programming and by the idealized gradient algorithm.

For the reader's convenience we repeat here briefly the statement of the problem. The call admission control problem introduced in Section 2.4 arises in the context where a telecommunication provider sells bandwidth of a communication link to customers so as to maximize the long term revenue. Customers are divided into different classes, called service types. In particular, we considered a single communication link with a total bandwidth of B units, which supports a finite set $\{1, 2, .., M\}$ of different service types. Each service type is characterized by its bandwidth demand $b(m)$, its call arrival rate $\alpha(m)$, and its average holding time $1/\beta(m)$, where we assume that the calls arrive according to independent Poisson Processes, and that the holding times are exponentially (and independently) distributed. When a new customer requests a connection, the link provider can decide to reject, or, if enough bandwidth is available, to accept the customer (*Call Admission Control*). Once accepted, a customer of class m seizes $b(m)$ units of bandwidth for the duration of the call. Whenever a call of service type m gets accepted, the provider receives an immediate reward of $c(m)$ units. The reward $c(m)$ can be interpreted as the price customers of service type m are paying for using $b(m)$ units of bandwidth of the link for the duration of the call. The goal of the link provider is to exercise call admission control such as to maximize the long term revenue.

7.1 Formulation of the Simulation-Based Algorithm

In this Section, we specialize the algorithm given in Chapter 6 to the call admission control problem.

Let us briefly restate the formulation of the call admission control problem as given in Section 2.4. Recall how we denoted the state i of the system by $i = (s, \omega)$, where s stands for the current configuration of the link, and ω identifies the event that triggers the next transition to a new configuration s'. The configuration s of the communication link is defined by an $M-$tuple $s = (s(1), ..., s(M))$, where $s(m)$ denotes the number of customers of class m currently using the link. The set of all possible configurations S_0 of the link is then given by

$$S_0 = \left\{ s \in \Re^M \;\middle|\; \sum_{m=1}^{M} s(m)b(m) \leq B, \quad s(m) \in \{0, 1, 2, ...\} \right\}.$$

The event ω is given by a M-tuple $\omega = (\omega(1), ..., \omega(M))$ where $\omega(m)$ equals 1 if the event corresponds to an arrival of a new customer of class m, $\omega(m)$ equals -1 if the event corresponds to a departure of a customer of class m, and $\omega(m)$ equals 0 otherwise. We allowed the trivial event $\omega = (0, ..., 0)$, which we denoted by ω^*. The set of all possible Ω is then given by

$$\Omega = \left\{ \omega \;\middle|\; \omega \in \{-1, 0, 1\}^N, \;\; \sum_{n=1}^{N} |\omega(n)| \leq 1 \right\},$$

and the state space S is given by

$$S = \{i = (s, \omega) \mid s \in S_0, \omega \in \Omega, \text{ and } \omega_m \geq 0 \text{ if } s(m) = 0.\}.$$

At a state $i = (s, \omega)$, for which the event ω corresponds to an arrival of a new call, the link provider can decide to accept or reject the call. Formally, we defined the set of possible control actions as $U = \{u_a, u_r\}$, where u_a stands for accepting a new call, and u_r stand for rejecting a new call. If the event ω does not correspond to an arrival of a new call, the link provider can still chose a control action, which however does not affect the transition probabilities and the one-stage reward.

The main result of this section is the following. Given a randomized call admission control policy $\mu(\theta)$, we can rewrite the expression of the gradient $\nabla\lambda(\theta)$ given in Chapter 6 as

$$\begin{aligned}
\nabla\lambda(\theta) &= \sum_{i \in S} \sum_{u \in U} \pi_i(\theta) \nabla\mu_u(i, \theta) q_{i,u}(\theta) \\
&= \sum_{i \in S} \sum_{u \in U} \pi_i(\theta) \nabla\mu_u(i, \theta) q'_{i,u}(\theta) \qquad (7.1)
\end{aligned}$$

where $q'_{iu}(\theta)$ is given by

$$q'_{i,u}(\theta) = E_\theta \left[\sum_{k=0}^{T-1} \left(g_{i_k}(u_k) - \lambda(\theta) \right) \mid i_0 = i, u_0 = u \right],$$

with $T = \min\{k > 0 \mid s_k = s^*\}$ being the first future time that the link is empty (which is different from the first future time that the recurrent state $i^* = (s^*, \omega^*)$ is visited).

Using Eq.(7.1), we can formulate the simulation-based algorithm of Section 6.1 as follows. At a typical time k, the state is i_k, and the values of θ_k, z_k, and $\tilde{\lambda}_k$ are available from the previous iteration. We then update z_k, θ_k and $\tilde{\lambda}_k$ according to

$$\begin{aligned}
z_{k+1} &= \begin{cases} \frac{\nabla\mu_{u_k}(i_k, \theta_k)}{\mu_{u_k}(i_k, \theta_k)}, & \text{if } s_k = s^*, \\ z_k + \frac{\nabla\mu_{u_k}(i_k, \theta_k)}{\mu_{u_k}(i_k, \theta_k)}, & \text{otherwise,} \end{cases} \\
\theta_{k+1} &= \theta_k + \gamma_k (g_{i_k}(\theta_k) - \tilde{\lambda}_k) z_{k+1}, \\
\tilde{\lambda}_{k+1} &= \tilde{\lambda}_k + \eta\gamma_k (g_{i_k}(\theta_k) - \tilde{\lambda}_k).
\end{aligned}$$

where u_k is the control action applied at state i_k. Note that in this variant of the algorithm proposed in Section 6.1 as we reset the vector z_k not only at visits to the state i^*, but at visits to all states $i = (s, \omega)$ for which $s = s^*$.

To prove this result, we derive a new Markov decision process on the state space $S' = S \cup S_0$. In particular, we split a transition from state $i = (s, \omega)$ to state $j = (s', \omega')$ into two transitions: a first transition which takes the $i = (s, \omega)$ to the new state $j' = s'$, and a second transition which takes state $j' = s'$ to the state $j = (s', \omega')$. Using r and $r(s)$, which we defined in Section 2.4 by

$$r = \max_{s \in S} \left\{ \sum_{m=1}^{M} (\alpha(m) + s(m)\beta(m)) \right\},$$

and

$$r(s) = \sum_{m=1}^{M} (\alpha(m) + s(m)\beta(m)), \qquad s \in S_0,$$

respectively, we obtain the following transition probabilities $P_{ij}(u)$, $u \in U$, for this new process:

1. if $i = (s, \omega) \in S$, $j = s' \in S_0$, ω corresponds to an arrival of a new call and $u = u_a$, then
$$P'_{ij}(u) = \begin{cases} 1, & \text{if } s' = s + \omega. \\ 1, & \text{if } s' = s \text{ and } s + \omega \notin S_0. \end{cases}$$

2. if $i = (s, \omega) \in S$, $j = s' \in S_0$, ω corresponds to an arrival of a new call and $u = u_r$, then
$$P'_{ij}(u) = \ 1, \quad \text{if } s' = s.$$

3. if $i = (s, \omega) \in S$, $j = s' \in S_0$, and ω corresponds to a departure of a call or $\omega = \omega^*$, then
$$P'_{ij}(u) = \ 1, \quad \text{if } s' = s + \omega.$$

4. if $j = s \in S_0$, and $j = (s', \omega') \in S$, $j = s' \in S_0$, then
$$P'_{ij}(u) = \begin{cases} \alpha(m)/r, & \text{if } s' = s \text{ and } \omega'_m = 1, \\ s'(m)\beta(m)/r, & \text{if } s' = s \text{ and } \omega'_m = -1. \\ 1 - r(s')/r, & \text{if } s' = s \text{ and } \omega'_m = \omega^*. \end{cases}$$

For all other cases, the transition probabilities $P_{ij}(u)$ are equal to 0.

Note that for this new process, only transition from states $i = (s, \omega) \in S$ to states $j = s' \in S_0$, and transitions from states $i = s \in S_0$ to states $j = (s, \omega) \in S$ occur. Furthermore, observe that for a transition which originates at a state $i = (s, \omega) \in S$, the next state $j = s'$ is a deterministic function of the state i and of the control action u applied at state i. On the other hand, for a transition which originates from a state $i = s \in S_0$, the probability that the next state j is a particular state in the set S does not depend on the control action, but only on s and the corresponding probabilities that a given event ω occurs at the configuration s.

We define the one-stage reward $g_i'(u)$ for this new process by

$$g_i'(u) = \begin{cases} c(m), & \text{if } i = (s,\omega) \in S,\ u = u_a,\ \omega_m = 1, \text{ and } s+\omega \in S_0, \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, for a given randomized call admission control policy $\mu(\theta)$, we define a new policy $\mu'(\theta)$ by

$$\mu_u'(i,\theta) = \begin{cases} \mu_u(i,\theta) & \text{if } i \in S, \\ 1 & \text{if } i \in S_0, \text{ and } u = u_r. \end{cases}$$

Note that for the case where $i \in S_0$ we can choose the value of the function $\mu_u'(i,\theta)$ arbitrarily, as the transition probabilities $P_{ij}(u)$ are independent of u. Here, we chose $\mu_u'(i,\theta)$ such that we have

$$\nabla \mu_u'(i,\theta) = 0, \text{ for all } i \in S_0.$$

From our definitions it follows that, for $i, j \in S$, the two-step transition probability ${P_{ij}'}^2(\theta)$ is equal to the transition probability $P_{ij}(\theta)$ of the original process. Furthermore, the state $s^* = (0,...,0) \in S_0$ is a recurrent state in the Markov chain with transition probabilities $P_{ij}'(\theta)$, which is eventually reached from each initial state $i \in S'$. We denote the steady-state probabilities for the Markov chain with transition probabilities $P_{ij}'(\theta)$ as $\pi_i'(\theta)$. It can then be verified, by using the balance equations, that the following relations hold,

$$\pi_i'(\theta) = \frac{1}{2}\pi_i(\theta), \quad i \in S, \quad \text{and,} \quad \pi_s'(\theta) = \frac{1}{2}\sum_{\omega \in \Omega} \pi_{(s,\omega)}(\theta), \quad s \in S_0,$$

where $\pi_i(\theta)$ is the steady-state probability of the Markov chain on the state space S with transition probabilities $P_{ij}(\theta)$.

As, for all states $i \in S_0$ and all control actions $u \in U$, the one-stage reward $g_i'(u)$ is equal to 0, we have the following relation

$$\lambda'(\theta) = \sum_{i \in S'} \pi_i'(\theta) g_i'(\theta) = \sum_{i \in S} \pi_i'(\theta) g_i'(\theta) = \sum_{i \in S} \frac{1}{2}\pi_i(\theta) g_i(\theta) = \frac{1}{2}\lambda(\theta).$$

Combining this relation, we the fact that $\nabla\mu_u'(i,\theta) = 0$, for all states $i \in S_0$, and $\nabla\mu_u'(i,\theta) = \nabla\mu_u(i,\theta)$, for all $i \in S$, we obtain

$$\begin{aligned} \nabla\lambda(\theta) = 2\nabla\lambda'(\theta) &= 2\sum_{i \in S}\sum_{u \in U} \pi_i'(\theta)\nabla\mu_u(i,\theta) q_{i,u}'(\theta) \\ &= 2\sum_{i \in S}\sum_{u \in U} \frac{1}{2}\pi_i(\theta)\nabla\mu_u(i,\theta) q_{i,u}'(\theta) \\ &= \sum_{i \in S}\sum_{u \in U} \pi_i(\theta)\nabla\mu_u(i,\theta) q_{i,u}'(\theta). \end{aligned}$$

7.2 Numerical Results

We consider the call admission control problem for an communication link with a total bandwidth of 10 units, which supports three different service types. The parameters associated

Table 7.1: Service Types.

SERVICE TYPE m	1	2	3
BANDWIDTH DEMAND $b(m)$	1	1	1
ARRIVAL RATE $\alpha(m)$	1.8	1.6	1.4
AVERAGE HOLDING TIME $1/\beta(m)$	0.6	0.5	0.4
IMMEDIATE REWARD $c(m)$	1	2	4

with the service types are given in Table 7.1. The configuration of the communication link is then defined by the vector $s = (s(1), ..., s(3))$, where $s(m)$, $m = 1, ..., 3$, denotes the number of customers of service type m currently using the link. The corresponding set of all possible configurations S_0 is given by

$$S_0 = \left\{ i \in \Re^3 \;\middle|\; \sum_{m=1}^{3} s(m)b(m) \leq 10, \quad s(m) \in \{0, 1, 2, ...\} \right\}.$$

For this particular choice of parameters, there are 286 different configurations.

Furthermore, we consider the situation where the provider uses a parameterized call admission control policy as given in Example 1 in Chapter 3. Given a parameter vector $\theta \in \Re^M$, the link provider accepts a new call at state $i = (s, \omega)$, i.e., chooses the control action u_a, with probability

$$\mu_{u_a}(i, \theta) = \begin{cases} \frac{1}{1+\exp\left(1.5(s \cdot b - \theta_m)\right)}, & \text{if } \omega_m = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Accordingly, the provider rejects a new call at state i, i.e., chooses the control action u_r, with probability $\mu_{u_r}(i, \theta) = 1 - \mu_{u_a}(i, \theta)$.

Note that we have

$$\mu_{u_a}(i,\theta) \geq 0.5, \qquad \text{if } s \cdot b \leq \theta_m,$$
$$\mu_{u_a}(i,\theta) \leq 0.5, \qquad \text{if } s \cdot b \geq \theta_m,$$

which implies that the parameter θ_m can be interpreted as a ("fuzzy") threshold value associated with the service type m, i.e., calls of service type m are likely to get rejected if the currently used bandwidth $s \cdot b$ exceeds the threshold θ_m, and are likely to get accepted otherwise.

For this parameterization of the control policy, the goal of the link provider is to tune the parameters $(\theta_1, ..., \theta_3)$ so that the corresponding admission control policy optimizes the long term revenue.

7.2.1 Optimal Control Policy

As the call admission control problem that we consider here involves a small state space, it is possible to compute an optimal call admission control policy using methods of dynamic programming. In particular, using Policy Iteration we obtain an optimal control policy which has an average reward of 0.8868. It is worth noting that this optimal admission control policy is a threshold policy, i.e., it can be characterized by a threshold value for each service type, such that calls of a specific service type get accepted, if the currently used bandwidth does not exceed the corresponding threshold value, and rejected otherwise. In particular, under the optimal admission control policy, customers of service type 1 get only accepted if the currently used bandwidth does not exceed the threshold value of 7 units, while customers of customers of service type 2 and 3 get always accepted (if enough bandwidth is available).

7.2.2 Idealized Gradient Algorithm

For this example, we can compute the gradient $\nabla\lambda(\theta)$ of the average reward with respect to θ. This allows us to implement an idealized gradient algorithm where the parameter vector gets updated according to the rule

$$\theta_{k+1} = \theta_k + \gamma_k \nabla\lambda(\theta_k),$$

where γ_k is a step size parameter. Here, we chose a constant step size parameter equal to 30. Starting with an initial parameter $\theta = (8, 8, 8)$, we obtained parameter vectors, and average rewards of the corresponding control policies, as given in Figure 7.1. Again, the average reward under the admission control policy associated with a parameter $\theta \in \Re^3$ can be exactly computed. After 100 iteration steps, the parameter vector θ_{100} is equal to $(7.5459, 11.7511, 12.8339)$, which corresponds to a call admission control policy with an average reward of 0.8808. The probabilities of accepting a new call under this control policy are given in Figure 7.2. Note that (as one expects from the optimal admission control policy obtained with dynamic programming), this control policy essentially always accepts calls of service types 2 and 3, and is likely to accept calls of service type 1 if the currently used bandwidth does not exceed the threshold value of 7. Compared with the optimal call admission control policy, whose average reward is equal to 0.8868, the control policy

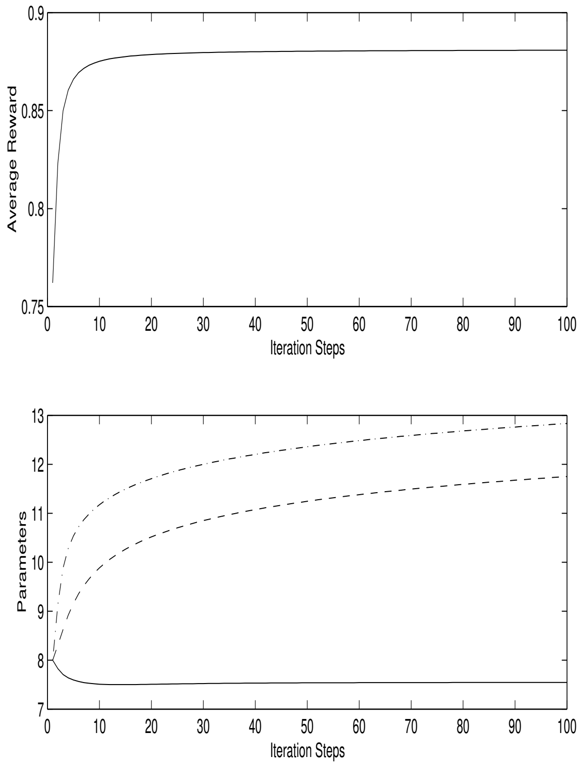

Figure 7.1: Parameter vectors, and average rewards of the corresponding admission control policies, obtained by the idealized gradient algorithm. The solid, dashed, and dash-dot line correspond to the threshold values θ_1, θ_2, and θ_3, associated with service types 1, 2, and 3, respectively.

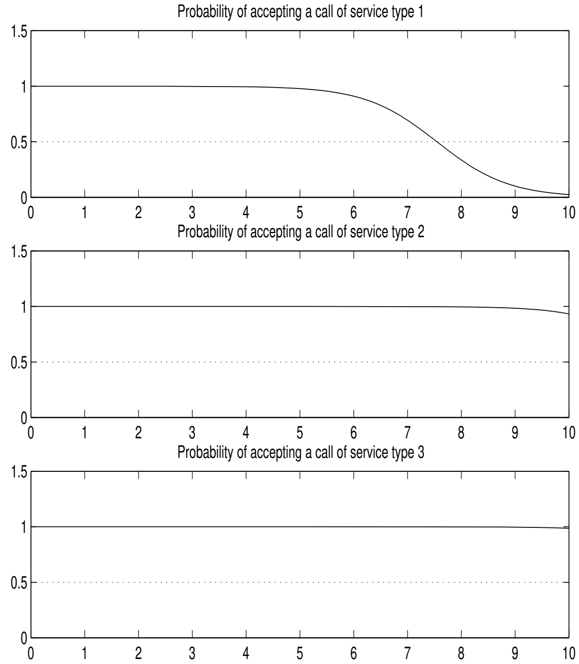

Figure 7.2: Probabilities of accepting a new call, given as a function of the used bandwidth on the link, under the control policy associated with the parameter vector $(7.5459, 11.7511, 12.8339)$ obtained by the idealized gradient algorithm.

corresponding the parameter vector θ_{100} has a lower average reward of 0.8808. This is due to the fact that the "sharp" threshold values of the optimal control policy as described above, have been replaced by "fuzzy" thresholds as given in Figure 7.2.

7.2.3 Simulation-Based Optimization

Instead of computing the gradient of the average reward with respect to θ exactly, we consider here the simulation-based algorithm given in Section 7.1, where the link provider employs simulation to get an estimate of the gradient and updates the parameter vector along an approximate gradient direction. At a typical time k, the state is i_k, and the values of θ_k, z_k, and $\tilde{\lambda}_k$ are available from the previous iteration. We then update z_k, θ_k and $\tilde{\lambda}_k$ according to

$$\begin{aligned} z_{k+1} &= \begin{cases} \frac{\nabla \mu_{u_k}(i_k,\theta_k)}{\mu_{u_k}(i_k,\theta_k)}, & \text{if } s_k = s^*, \\ z_k + \frac{\nabla \mu_{u_k}(i_k,\theta_k)}{\mu_{u_k}(i_k,\theta_k)}, & \text{otherwise,} \end{cases} \\ \theta_{k+1} &= \theta_k + \gamma_k (g_{i_k}(\theta_k) - \tilde{\lambda}_k) z_{k+1}, \\ \tilde{\lambda}_{k+1} &= \tilde{\lambda}_k + \eta\gamma_k (g_{i_k}(\theta_k) - \tilde{\lambda}_k). \end{aligned}$$

where u_k is the control action applied at state i_k.

We implemented this algorithm by starting with an initial parameter $\theta = (8, 8, 8)$, an initial estimate of the average reward $\tilde{\lambda}_0 = 0.78$, an initial value of $z_0 = 0$, and an initial step size $\gamma_0 = 0.0003$, which we adjusted after every $500,000$ iteration steps according to the rule

$$\gamma_{k+1} = 0.9\gamma_k.$$

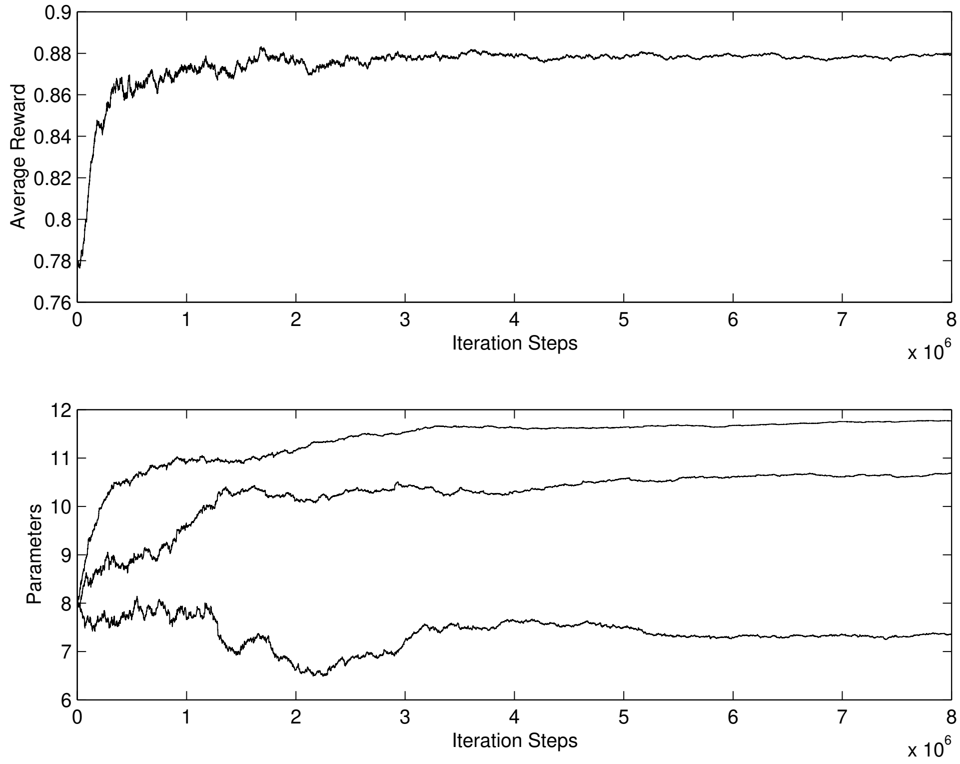

Figure 7.3: Parameter vectors, and estimates of the average reward, obtained by the simulation-based algorithm.

The parameter η was set to 0.1. The corresponding trajectories of the parameter vectors and average reward are given in Figure 7.3.

We make here the following observations:

1. The algorithm makes rapid progress in the beginning, improving the average reward from 0.78 to 0.87 within the first $1 \cdot 10^6$ iteration steps.

2. After reaching an average reward of 0.87, the algorithm makes only slow progress obtaining after $8 \cdot 10^6$ iterations the parameter vector

$$\theta_{8 \cdot 10^6} = (7.3540, 10.6850, 11.7713)$$

   which corresponds to an admission control policy with an average reward of 0.8789. This average reward still slightly below the average reward of 0.8808 obtained by the idealized gradient algorithm.

3. While the algorithm makes slow progress and fluctuates in the later iteration steps, the fluctuations remain small and the performance of the corresponding control policies never deteriorates.

Note that this behavior is not unlike the one of the idealized algorithm (see Figure 7.1), where the average reward also gets improved dramatically in the beginning, but makes only slow progress in the later iteration steps.

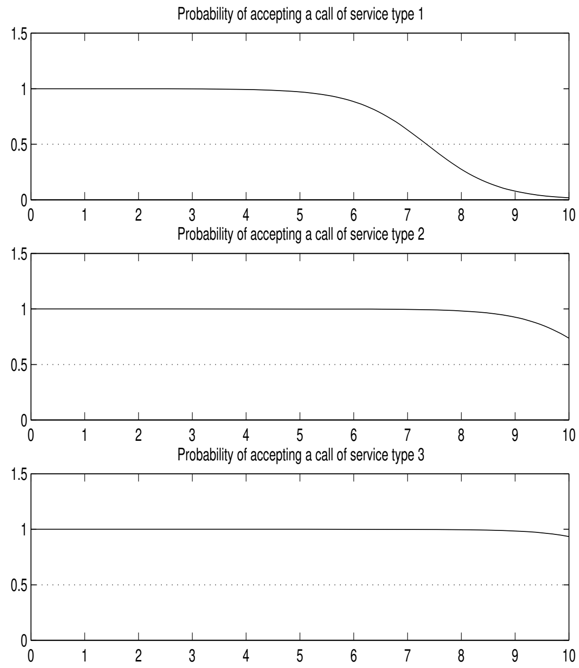

Figure 7.4: Probabilities of accepting a new call, given as a function of the used bandwidth on the link, under the control policy associated with the parameter vector $(7.3540, 10.6850, 11.7713)$ obtained by the simulation-based algorithm.

The probabilities of accepting a new call under the control policy obtained with the simulation-based algorithm are given in Figure 7.4.

7.2.4 Simulation-Based Optimization using $\tilde{q}_{S^*,i,u}(\theta, \tilde{\lambda})$

As mentioned in Chapter 6, we can derive variants of the algorithm in the previous subsection, which attempt to reduce the variance of the updates. In this subsection we consider a variant of the simulation-based algorithm proposed in Section 7.1, which is based on estimates of the differential reward $q_{i,u}(\theta)$ of state i, and control action u, given by

$$\tilde{q}_{S^*,i,u}(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} \Big(g_{i_k}(\theta) - \tilde{\lambda}\Big),$$

where S^* is a subset of S_0 containing s^*, $\tilde{\lambda}$ is an estimate of the average reward $\lambda(\theta)$, and $T = \min\{k > 0 \mid s_k = s^*\}$ is the first future time k in a trajectory that starts at state $i_0 = i$, and uses control action $u_0 = u$ at state i_0, for which the configuration of the link belongs to the set S^*. In particular, at time k, we then update z_k, θ_k and $\tilde{\lambda}_k$ according to

$$\begin{aligned}
z_{k+1} &= \begin{cases} \frac{\nabla \mu_{u_k}(i_k,\theta_k)}{\mu_{u_k}(i_k,\theta_k)}, & \text{if } s_k \in S^*, \\ z_k + \frac{\nabla \mu_{u_k}(i_k,\theta_k)}{\mu_{u_k}(i_k,\theta_k)}, & \text{otherwise,} \end{cases} \\
\theta_{k+1} &= \theta_k + \gamma_k(g_{i_k}(\theta_k) - \tilde{\lambda}_k) z_{k+1}, \\
\tilde{\lambda}_{k+1} &= \tilde{\lambda}_k + \eta\gamma_k(g_{i_k}(\theta_k) - \tilde{\lambda}_k).
\end{aligned}$$

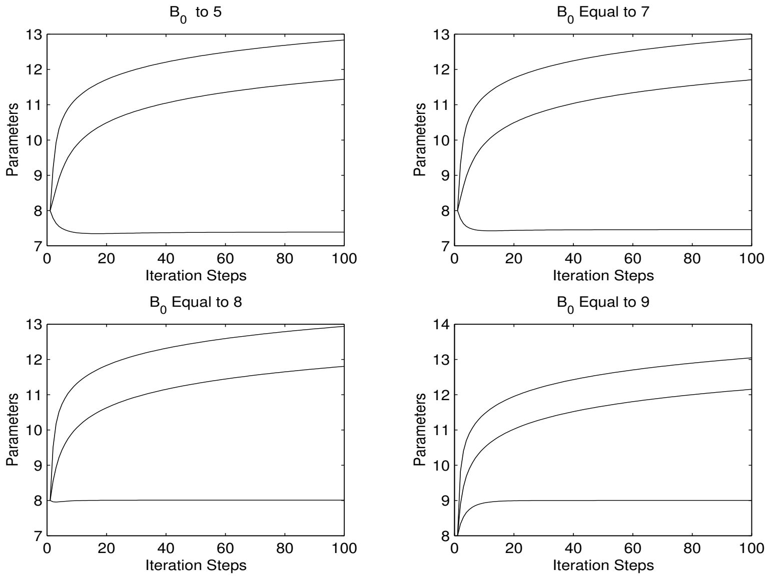

Figure 7.5: Parameter vectors obtained by modified idealized gradient method using the set $S^*_{B_0}$ for values of B_0 equal to 5, 7, 8 and 9.

where u_k is the control action applied at state i_k.

Here, we consider the following choice for the set S^*,

$$S^* = \left\{ i = (s, \omega) \in S \mid \sum_{m=1}^{3} s(m)b(m) \leq B_0 \right\},$$

where B_0 is a parameter in the set $\{1, ..., 10\}$. For a particular choice of B_0, we denote this set as $S^*_{B_0}$. The set $S^*_{B_0}$ contains all states for which the used bandwidth of the link does not exceed B_0 units.

As discussed in Chapter 5, the above mentioned algorithm can reduce the variance of the updates, but introduces a bias into the estimate of the gradient $\nabla\lambda(\theta)$. To study the effect of this bias on the performance of the algorithm, we consider its deterministic version given by the update rule

$$\theta_{k+1} = \theta_k + \gamma_k \sum_{i \in S} \sum_{u \in U} \pi_i(\theta) \nabla \mu_u(i, \theta) q_{S^*,i,u}(\theta),$$

where

$$q_{S^*,i,u}(\theta) \quad = \quad E_\theta \left[\sum_{k=0}^{T-1} (g_{i_k}(u_k) - \lambda(\theta)) \mid i_0 = i, u_0 = u \right].$$

and where i_k and u_k is the state and control at time k, and $T = \min\{k > 0 \mid s_k \in S^*\}$ is the first future time for which the configuration s_k is in contained in S^*. The resulting iterations for values of B_0 equal to 5, 7, 8, and 9, are given in Figure 7.5. We note that the deterministic version of the algorithm is robust with respect to the bias introduced by the modified values of the differential reward function. In particular, for the sets S^*_5 and S^*_7 the effect of the bias is small.

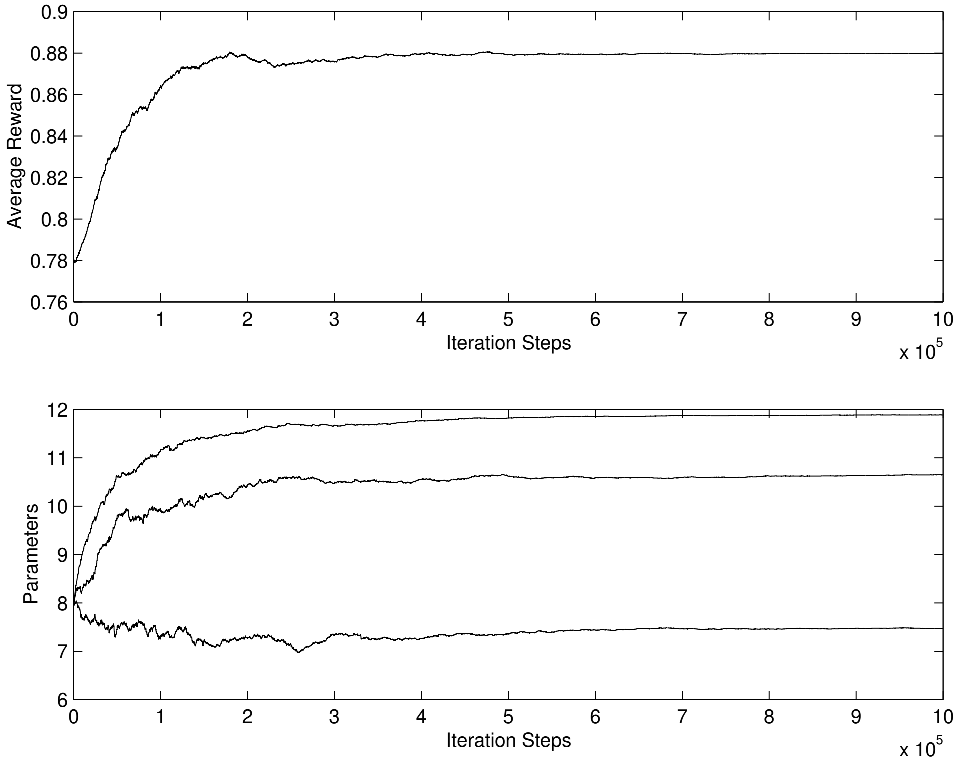

Figure 7.6: Parameter vectors, and estimates of the average reward, obtained by the modified simulation-based algorithm using the set S_7^*.

Using the set S_7^*, we then implemented the simulation-based version of the algorithm given above. We started with an initial parameter $\theta = (8, 8, 8)$, an initial estimate of the average reward $\tilde{\lambda} = 0.78$, an initial value of $z = 0$, and an initial step size $\gamma_0 = 0.003$, which we adjusted after every $50,000$ iteration steps according to the rule

$$\gamma_{k+1} = 0.8\gamma_k.$$

The parameter η was set to 0.01. The resulting parameter vectors are given in Figure 7.6.

Note that this algorithm makes much faster progress than the simulation-based algorithm proposed in Section 7.1, obtaining after 150,000 iterations steps a control policy with an average reward around 0.87. Furthermore, the parameter vector of $(7.4743, 10.6494, 11.8870)$ obtained after $1 \cdot 10^6$ iterations corresponds to a control policy with an average reward of 0.8792, which is even slightly higher than the one obtained with the algorithm in the previous subsection.

7.2.5 Simulation-Based Optimization using $\tilde{q}_{\alpha,i,u}(\theta, \tilde{\lambda})$

In this subsection, we discuss a second approach to reduce the variance in the estimate of the gradient. Here we consider an algorithm where we estimate the differential reward $q_{i,u}(\theta)$

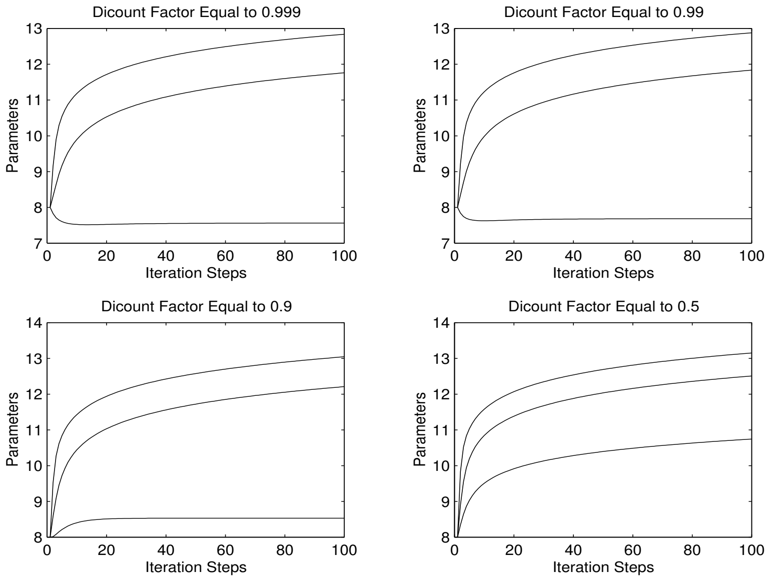

Figure 7.7: Parameter vectors obtained by modified idealized gradient method using a discount factor α with values equal to 0.5, 0.9, 0.99 and 0.999.

of state i, and control action u, by

$$\tilde{q}_{\alpha,i,u}(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} \alpha^k \Big(g_{i_k}(\theta) - \tilde{\lambda} \Big)$$

where $\alpha \in [0, 1)$ is a discount factor, $\tilde{\lambda}$ is an estimate of the average reward $\lambda(\theta)$, and $T = \min\{k > 0 \mid s_k = s^*\}$ is the first future time k in a trajectory that starts at state $i_0 = i$, and uses control action $u_0 = u$ at state i_0, at which the configuration of the link is equal to s^*.

Again, this approach will introduce a bias into the gradient direction. To study the effect of this bias on the algorithm, we consider its deterministic version given by the update rule

$$\theta_{k+1} = \theta_k + \gamma_k \sum_{i \in S} \sum_{u \in U} \pi_i(\theta) \nabla \mu_u(i, \theta) q_{\alpha,i,u}(\theta),$$

where

$$q_{\alpha,i,u}(\theta) \quad = \quad E_\theta \left[\sum_{k=0}^{T-1} \Big(\alpha^k g_{i_k}(u_k) - \lambda(\theta) \Big) \mid i_0 = i, u_0 = u \right].$$

and where i_k and u_k is the state and control at time k, and $T = \min\{k > 0 \mid i_k = i^*\}$ is first future time k at which the state i_k is equal to i^*.

The resulting iterations for values of the discount factor α equal to 0.5, 0.9, 0.99, and 0.999, are given in Figure 7.7. Again, the deterministic version of the algorithm is robust with respect to the bias introduced by the modified values of the differential reward function.

Using the value of α equal to 0.99, we then implemented the simulation-based version of the algorithm, which is given as follows. At a typical time k, the state is i_k, and the values

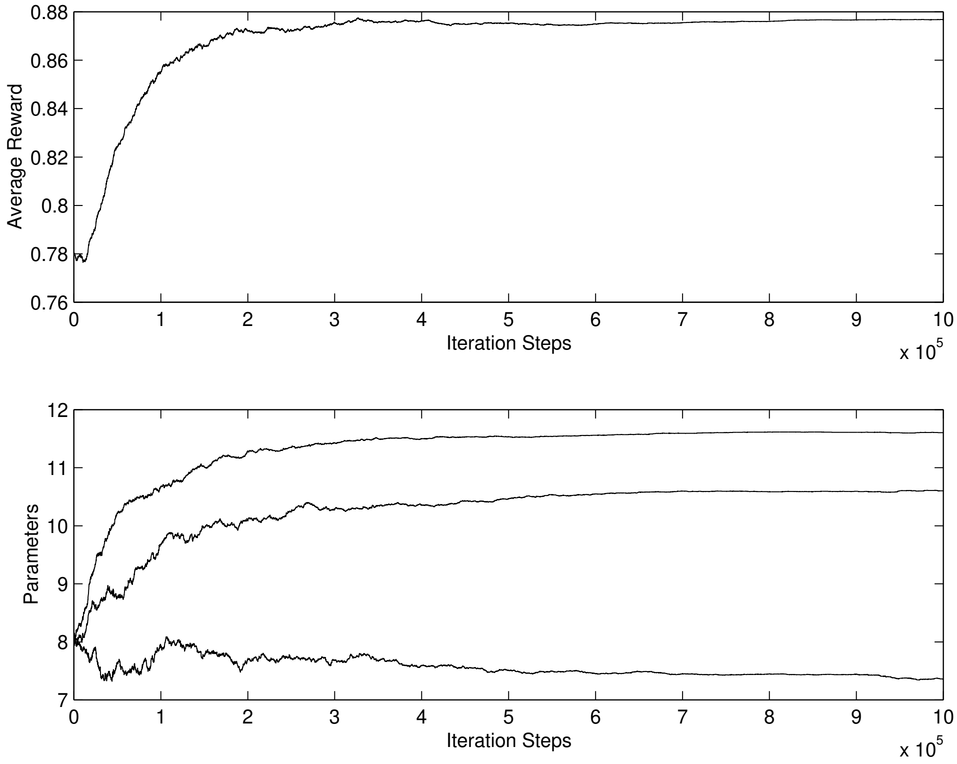

Figure 7.8: Parameter vectors, and estimates of the average reward, obtained by modified simulation-based algorithm using a discount factor $\alpha = 0.99$.

of θ_k, z_k, and $\tilde{\lambda}_k$ are available from the previous iteration. We then update z_k, θ_k and $\tilde{\lambda}_k$ according to

$$
\begin{aligned}
z_{k+1} &= \begin{cases} \frac{\nabla\mu_{u_k}(i_k,\theta_k)}{\mu_{u_k}(i_k,\theta_k)}, & \text{if } s_k = s^*, \\ \alpha z_k + \frac{\nabla\mu_{u_k}(i_k,\theta_k)}{\mu_{u_k}(i_k,\theta_k)}, & \text{otherwise,} \end{cases} \\
\theta_{k+1} &= \theta_k + \gamma_k(g_{i_k}(\theta_k) - \tilde{\lambda}_k)z_{k+1}, \\
\tilde{\lambda}_{k+1} &= \tilde{\lambda}_k + \eta\gamma_k(g_{i_k}(\theta_k) - \tilde{\lambda}_k).
\end{aligned}
$$

where u_k is the control action applied at state i_k.

We started the algorithm with an initial parameter $\theta = (8, 8, 8)$, an initial estimate of the average reward $\tilde{\lambda}_0 = 0.78$, an initial value of $z_0 = 0$, and an initial step size $\gamma_0 = 0.002$, which we adjusted after every $50,000$ iteration steps according to the rule

$$\gamma_{k+1} = 0.8\gamma_k.$$

The parameter η was set to 0.015. The resulting parameter vectors are given in Figure 7.8.

Again, this algorithm makes much faster progress than the simulation-based algorithm proposed in Section 7.1, obtaining after $1{\cdot}10^6$ iterations a parameter vector of $(7.3553, 10.6034, 11.6073)$ which corresponds to a control policy with an average reward of 0.8785 which is only slightly lower that the one of Section 7.2.3.

7.3 Closing Remarks

In this chapter, we illustrated the algorithms proposed in Chapter 6 by considering the call admission control problem. We limited our study to a small instance of the problem, for which the gradient of the average reward can be computed exactly and an idealized gradient algorithm can be implemented. We observed that for this case, the behavior of the simulation-based algorithms is similar to the one of the idealized gradient algorithm. The method tends to make fast progress in the beginning obtaining a parameter vector for which the corresponding performance is close to the one obtained with the idealized gradient methods. Afterwards, it continues to improve the performance, however at a slow rate. This is due to the fact that a very small step size is needed to "average out" the noise term and to obtain an accurate gradient direction. Furthermore, we observe that versions of the algorithm which reduce the variance in the updates, but introduce a small bias, do better in practice. This confirms the empirical result that gradient-type algorithms are robust with respect to a small bias, and earlier numerical studies [Miz98].

The purpose of this chapter was to illustrate the algorithms on a specific application, not to advocate our methodology as an approach to solve the call admission control problem. Nevertheless, we refer here to related work which helps to put the method and its performance in perspective.

The call admission control problem can be formulated as a Markov Decision Process, in particular a multiple-service, single-resource system, or stochastic knapsack. The literature on this type of problem (see [JV94, Ros95] for an overview and references) has focused on characterizing optimal control policies [FGH81, RT89], finding efficient ways to compute the steady state probability distributions (see [Ros95] for references), which allows the computation of the average reward, and analyzing certain classes of control policies [JV94]. In particular, coordinate convex policies have been studied which have the following nice properties [JV94]: the steady state distribution has a product from, optimal coordinate convex policies can be readily characterized, and the performance of optimal coordinate convex policies tends to be close to optimal. In [JV94], Jordan and Varaiya compared optimal coordinate convex policies with optimal policies obtained by dynamic programming and noted, *".. It seems however, that the two solutions, although sometimes very different, generate average rates of revenue that are not very different. Since the Dynamic Programming technique becomes burdensome to calculate as the size of the state space increases, it may be valuable to write an algorithm to find the optimal coordinate convex policy, or a close approximation to it...."*. The methodology proposed here could be used to do this.

For the call admission control problem, trunk-reservation and dynamic trunk-reservation policies have been studied in [LZM89, Lin87, Rob83], and efficient algorithms for their evaluation have been proposed. These numerical methods do not rely on simulation of the systems. However, the arrival and departure rates of the different service types have to be known, unlike in our approach.

In [MT97], different methods of neuro-dynamic programming/reinforcement learning have been applied to the call admission control problem [MT97]. In this approach, for a small problem with 140 possible configurations of the link, $3 \cdot 10^7$ simulation steps were needed, and an optimal control policy was obtained. For larger problems, the resulting policies performed slightly worse than a well-designed threshold policy. The disadvantage of this method, relative to ours, is that its theoretical foundations are still somewhat incomplete.

Chapter 8

Summary and Future Work

In this chapter we provide a summary of the thesis and indicate possible directions of future research. We will keep the discussion at a high level, and refer for more detailed information to the corresponding chapters.

Summary

We presented a methodology for optimizing Markov reward processes that depend on a parameter vector $\theta \in \Re^K$ (Section 3.2). We limited our discussion to finite state Markov reward processes for which one of the following properties holds.

1. There exits a state i^* such that, for every parameter vector $\theta \in \Re^K$, i^* is a reward-free terminal state that is eventually reached from every state in the state space (Chapter 4).

2. There exists a state i^* such that, for every parameter vector $\theta \in \Re^K$, i^* is eventually reached from every state in the state space (Chapter 5).

Both are standard assumptions for Markov chains and hold for many problems. Furthermore, we imposed a regularity assumption on the transition probabilities (Assumption 5 (MRP Parameterization)). We revisited this regularity condition in the context of Markov decision processes and showed that there is a rich class of functions that satisfies this assumption (Chapter 6).

We proposed gradient-type algorithms for updating θ based on the simulation of a single sample path, so as to improve a given performance measure. As possible performance measures, we considered the weighted reward-to-go (Chapter 4) and the average reward (Chapter 5). The corresponding algorithms

(a) can be implemented online and update the parameter vector either at visits to a certain state, or at every time step.

(b) have the property that the gradient (with respect to θ) of the performance measure converges to 0 with probability 1. This is the strongest possible result for gradient-related stochastic approximation algorithms.

As a special case, the proposed methodology can be applied to finite-state, finite-action Markov decision processes that involve a randomized control policy described in terms of a parameter vector. In this case, Assumption 5 (MRP Parameterization) on the transition probabilities translates into a regularity assumption on the parameterized control policy (Assumption 12 (MDP Parameterization)). We showed that there exists a rich class of functions that satisfies Assumption 12 (Chapter 6).

To illustrate the algorithms that we proposed, and to test their behavior on a "real-world" problem, we carried out a case study (Chapter 7). We considered the call admission control problem where a telecommunication provider sells bandwidth of a communication link to customers so as to maximize the average reward. We showed how a randomized call admission control policy can be described in terms of parameter vector $\theta \in \Re^K$, and formulated the corresponding algorithms for optimizing θ. Our study was limited to a small instance of the problem for which the gradient of the average reward can be computed exactly, and for which an idealized gradient algorithm can be implemented. For this case, the behavior of the simulation based algorithms was very similar to the idealized gradient algorithm. The method tends to make fast progress in the beginning obtaining a parameter vector for which the corresponding performance is close to the one obtained with the idealized gradient methods. However, convergence is slow, which is to be expected for an algorithm involving a stochastic noise term. Furthermore, we observe that versions of the algorithm which reduce the variance in the updates, but introduce a small bias, lead to better performance in practice.

Based on our theoretical and experimental studies presented in this thesis, we believe that the proposed methodology has the potential to address Markov reward and decision Processes which involve a large state space or for which an exact mathematical model is not available.

Future Research

We restricted our analysis to problems involving a finite state space and a finite set of control actions. However, it would be desirable to extend the theory to more general cases. In particular, Markov reward processes involving a continuous state space are of interest, as this would allow us to apply the methodology to semi-Markov Processes, where the time between transitions is not fixed, but is given in terms of a probability density function. Semi-Markov processes can be transformed into Markov processes by augmenting the state with the time that has passed since the occurrence of the previous transition. This modification however introduces a continous state space. Instead of using such a "brute force" approach, the method may also be extended to semi-Markov Processes by using the distribution of the time between transitions explicitely, resulting in a simpler analysis and more elegant results.

As pointed out in Chapter 5, the algorithm for updating the parameter vector to optimize the average reward belongs to the class considered in [Bor97]. To prove convergence, the author of [Bor97] used a time-scale separation mechanism, where one parameter gets updated on a much slower time scale. We showed that there are instances where this is not needed. It is unclear if our technique can be extended to a more general class of algorithms of the form given in [Bor97], or the technique relies on a special property inherent to the situation considered here.

We noted in Chapter 7 that methods which reduce the variance, but introduce a bias into the update direction, tend to perform better in practice. Motivated by our work on the call admission control problem, we proposed two methods with this property. However, we expect that several other approaches exist. To explore these possibilities, more case studies are needed that apply the methodology to large-scale problems.

Bibliography

[Ber95a] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. I and II.* Athena Scientific, Belmont, MA, 1995.

[Ber95b] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1995.

[Bor97] V.S. Borkar. Stochastic Approximation with Two Time Scales. *Systems and Control Letters*, 29:291–294, 1997.

[BSA83] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuron-like Elements that Can Solve Difficult Learning Control Problems. *IEEE Trans. on Systems, Man and Cybernetics*, 13:835–846, 1983.

[BT96] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA, 1996.

[BT97] D. P. Bertsekas and J. N. Tsitsiklis. Gradient Convergence in Gradient Methods. *Technical Report LIDS-P-2404, Lab. for Info. and Decision Systems, Massachusetts Institute of Technology, MA, submitted to the SIAM J. Optimization and Control*, 1997.

[CC97] X. R. Cao and H. F. Chen. Perturbation Realization, Potentials, and Sensitivity Analysis of Markov Processes. *IEEE Transactions on Automatic Control*, 42:1382–1393, 1997.

[CR94] E. K. P. Chong and P. J. Ramadage. Stochastic Optimization of Regenerative Systems Using Infinitesimal Perturbation Analysis. *IEEE Trans. on Automatic Control*, 39:1400–1410, 1994.

[CW98] X. R. Cao and Y. W. Wan. Algorithms for Sensitivity Analysis of Markov Systems through Potentials and Perturbation Realization. *IEEE Trans. on Control Systems Technology*, 6:482–494, 1998.

[Del96] B. Delyon. General Results on the Convergence of Stochastic Algorithms. *IEEE Trans. on Automatic Control*, 41:1245–1255, 1996.

[FGH81] G. J. Forschini, B. Gopinath, and J. F. Hayes. Optimal allocation of servers to two types of competing customers. *IEEE Trans. Commun.*, 29:1051–1055, 1981.

[FH94] M. C. Fu and J.-Q. Hu. Smoothed Perturbation Analysis Derivative Estimation for Markov Chains. *Operations Research Letters*, 15:241–251, 1994.

[FH97] M. Fu and J.-Q. Hu. *Conditional Monte Carlo: Gradient Estimation and Optimization Applications.* Kluwer Academic Publisher, Boston, MA, 1997.

[Gal95] R. G. Gallager. *Discrete Stochastic Processes.* Kluwer Academic Publishers, Boston/Dordrech/London, 1995.

[Gly86] P. W. Glynn. Stochastic Approximation for Monte Carlo Optimization. *Proceedings of the 1986 Winter Simulation Conference*, pages 285–289, 1986.

[Gly87] P. W. Glynn. Likelihood Ratio Gradient Estimation: An Overview. *Proceedings of the 1987 Winter Simulation Conference*, pages 366–375, 1987.

[HC91] Y. C. Ho and X. R. Cao. *Perturbation Analysis of Discrete Event Systems.* Kluwer Academic Publisher, Boston, MA, 1991.

[JSJ95] T. Jaakkola, S. P. Singh, and M. I. Jordan. Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems. *Advances in Neural Information Processing Systems*, 7:835–846, 1995.

[JV94] S. Jordan and P. P. Variya. Control of multiple service, multiple resource communication networks. *IEEE Trans. Commun.*, 42:2979–2988, 1994.

[Lin87] K. Lindenberger. Blocking for multi-slot heterogenuous traffic streams offered to a trunk group with reservation. *5th ITC Seminar: Traffic Engineering for ISDN Design and Planning, Lake Como, Italy*, 1987.

[Lju77] L. Ljung. Analysis of Recursive Stochastic Algorithms. *IEEE Trans. on Automatic Control*, 22:551–575, 1977.

[LZM89] K.-Q. Liao, Z. Ziong, and L. G. Mason. Dynamic link bandwidth allocation in an integrated services network. *ICC'89, Boston*, 1989.

[Mih98] O. Mihatsch. Personal communication. 1998.

[Miz98] E. Mizutani. A policy-only evaluation-free learning by actor neural networks. *Term paper, Course EE 298, University of California, Berkeley*, 1998.

[MMST98] P. Marbach, O. Mihatsch, M. Schulte, and J. N. Tsitsiklis. Reinforcement Learning for Call Admission Control and Routing in Integraged Service Networks. *Advances in Neural Information Processing Systems*, 10:922–928, 1998.

[MT97] P. Marbach and J. N. Tsitsiklis. A Neuro-Dynamic Programming Approach to Call Admission Control in Integrated Service Networks: The Single Link Case. *Technical Report LIDS-P-2394, Lab. for Info. and Decision Systems, Massachusetts Institute of Technology, MA*, 1997.

[Put94] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, New York, 1994.

[Rao96] K.V. Rao. Learning Algorithms for Markov Decision Processes. *Master Thesis, Dept. of Computer Science and Automation, Indian Institute of Science, Banglore*, 1996.

[Rob83] J. W. Roberts. Teletraffic models for the telecom 1 integrated services system. *Proc. ITC-10, Montreal*, 1983.

[Ros95] K. W. Ross. *Multiservice Loss Models for Broadband Telecommunication Networks.* Springer-Verlag, Berlin, Heidelberg, New York, 1995.

[RT89] K. W. Ross and D. Tsang. The stochastic knapsack problem. *IEEE Trans. Commun.*, 37:740–747, 1989.

[SB98] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

[Sut88] R. S. Sutton. Learning to Predict by the Methods of Temporal Differences. *Machine Learning*, 3:345–352, 1988.

[Tes92] G. J. Tesauro. Practical Issues in Temporal Difference Learning. *Machine Learning*, 8:257–277, 1992.

[TR96] J. N. Tsitsiklis and B. Van Roy. Feature-Based Methods for Large Scale Dynamic Programming. *Machine Learning*, 22:59–94, 1996.

[TR97] J. N. Tsitsiklis and B. Van Roy. Optimal Stopping of Markov Processes: Hilbert Space Theory, Approximation Algorithms, and an Application to Pricing High-Dimensional Financial Derivatives. *Lab. for Info. and Decision Systems Report LIDS-P-2389, Massachusetts Institute of Technology, Cambridge, MA, submitted to the IEEE Trans. on Automtic Control*, 1997.

[Tsi98] J.N. Tsitsiklis. Personal communication. 1998.

[Wat89] C. Watkins. *Learning from Delayed Rewards.* Ph.D. Thesis, Cambridge University, UK, 1989.

Appendix A

Preliminary Lemmas

In this appendix, we provide the proofs for Lemma 2, and Lemma 4. We will first consider Lemma 4 and show that the average reward $\lambda(\theta)$, the steady-state probability vector $\pi(\theta)$, the differential reward vector $v(\theta)$, and the mean-recurrence time $E_\theta[T]$ are bounded, twice differentiable and have bounded first and second derivatives, and $E_\theta[T^s]$ is bounded for every $s > 0$.

Proof of Lemma 4: By Assumption 5 (MRP Parameterization), the balance equations (3.7) and (3.8) can be written in the form $A(\theta)\pi(\theta) = a$, where the entries of $A(\theta)$ are bounded, twice differentiable, and have bounded first and second derivatives. Since the balance equations have a unique solution, the matrix $A(\theta)$ is invertible for every $\theta \in \Re^K$ and Cramer's rule yields

$$\pi(\theta) = \frac{C(\theta)}{\det(A(\theta))}, \tag{A.1}$$

where $C(\theta)$ is a vector whose entries are polynomial functions of the entries of $A(\theta)$. Using Assumption 5 (MRP Parameterization), $C(\theta)$ and $\det(A(\theta))$ are bounded, twice differentiable and have bounded first and second derivatives.

More generally, suppose that $P \in \overline{\mathcal{P}}$, i.e. P is the limit of the stochastic matrices $P(\theta_k)$ along some sequence (θ_k). The corresponding balance equations are again of the form $A(P)\pi = a$, where $A(P)$ is a matrix depending on P. Under Assumption 7 (MRP Recurrence), these balance equations have again a unique solution, which implies that $|\det(A(P))|$ is positive. Note that $|\det(A(P))|$ is a continuous function of P, and P lies in the set $\overline{\mathcal{P}}$, which is compact. It follows that $|\det(A(P))|$ is bounded below by a positive constant c. Since every $P(\theta)$ belongs to $\overline{\mathcal{P}}$, it follows that $|\det(A(\theta))| \geq c > 0$, for every θ. This fact, together with Eq.(A.1) implies that $\pi(\theta)$ is bounded, twice differentiable and has bounded first and second derivatives. The same property holds true for $\lambda(\theta)$ given by

$$\lambda(\theta) \sum_{i \in S} \pi_i(\theta) g_i(\theta),$$

as the functions $g_i(\theta)$, $i \in S$, are bounded, twice differentiable and have bounded first and second derivatives by Assumption 5 (MRP Parameterization). Furthermore, the differential reward vector $v(\theta)$ is the unique solution to the system of linear equations

$$v_i = g_i(\theta) - \lambda(\theta) + \sum_{j \in S} P_{ij}(\theta) v_j, \qquad i \in S \backslash \{i^*\},$$
$$v_{i^*} = 0,$$

and it follows by the same argument as the one for $\pi(\theta)$, that $v(\theta)$ is bounded, twice differentiable, and has bounded first and second derivatives.

Next, we consider the system of linear equations

$$t_i = 1 + \sum_{j \in S} P_{ij}(\theta) t_j, \qquad i \in S \backslash \{i^*\},$$
$$t_{i^*} = 0.$$

By the same argument as given above, there is an unique solution $t(\theta)$ for which the components $t_i(\theta)$ are bounded, twice differentiable have bounded first and second derivatives. Note that we have the following relations,

$$t_i(\theta) = E_\theta[T \mid i_0 = i], \qquad \text{for all } i \in S \backslash \{i^*\},$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that the state i_k is equal to i^*, and

$$E_\theta[T] = 1 + \sum_{j \in S} P_{ij}(\theta) t_j(\theta).$$

It follows that $E_\theta[T]$ is bounded, twice differentiable has bounded first and second derivatives, as the functions $P_{ij}(\theta)$ and $t_j(\theta)$ have that regularity property.

Finally, by Assumption 5 (MRP Parameterization) and 7 (MRP Recurrence), there exists a constant $\rho > 0$, such that for all parameters $\theta \in \Re^K$ and all states $i \in S$, we have

$$\sum_{n=1}^{N} P^n_{ii^*} \geq \rho,$$

where we use the fact that the set $\overline{\mathcal{P}}$ is compact, and the transition probabilities are continuous in θ. It follows that there exists a constant C such that, for all integers $n \geq 1$, we have

$$P(T \geq n \mid i_0 = i^*) \leq C \rho^{\frac{n}{N}},$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. This implies that for every integer $s > 0$, there exists a constant D_s, such that for all $\theta \in \Re^K$, we have

$$E_\theta[T^s \mid i_0 = i^*] \leq D_s.$$

□

We can use the result of this lemma to prove Lemma 2.

Proof of Lemma 2: Recall that Assumption 6 (MRP Termination) implies Assumption 7 (MRP Recurrence). Furthermore observe that under Assumption 6 the average reward is equal to 0, and that the reward-to-go vector $J(\theta)$ is equal to the differential reward vector $v(\theta)$. These observations, together with Lemma 4, imply hat $J(\theta)$ is bounded, twice differentiable, and has bounded first and second derivatives. The result that $\chi(\theta)$ is is bounded, twice differentiable, and has bounded first and second derivatives, follows then immediately from the definition of $\chi(\theta)$ given by Eq.(3.6). $\square$

Appendix B

Proofs for Chapter 4

B.1 Proof of Proposition 5

In this section, we prove convergence of the algorithm proposed in Section 4.3, which is given by the update rule

$$\theta_{m+1} \quad = \quad \theta_m + \gamma_m F_m(\theta_m),$$

where

$$F_m(\theta_m) = \sum_{n=t_m}^{t_{m+1}-1} \left(\tilde{J}_{i_n}(\theta_m) \frac{\nabla P_{i_{n-1}i_n}(\theta_m)}{P_{i_{n-1}i_n}(\theta_m)} + \nabla g_{i_n}(\theta_m) \right),$$

$$\tilde{J}_{i_n}(\theta) = \sum_{k=n}^{t_{m+1}-1} g_{i_k}(\theta), \qquad n = t_m + 1, \ldots, t_{m+1} - 1,$$

and $\tilde{J}_{i_n}(\theta) = 0$, for $n = t_m$. We rewrite the algorithm in the form

$$\theta_{m+1} = \theta_m + \gamma_m \nabla \chi(\theta_m) + \varepsilon_m, \tag{B.1}$$

where

$$\varepsilon_m = \gamma_m (F_m(\theta_m) - \nabla \chi(\theta_m))$$

Note that, by Proposition 4, we have

$$E[\varepsilon_m \mid \mathcal{F}_m] = 0, \tag{B.2}$$

where $\mathcal{F}_m = \{\theta_0, i_0, i_1, \ldots, i_{t_m}\}$ stands for the history of the algorithm up to and including time t_m. We first characterize the behavior of ε_m.

Lemma 10 *There exists a constant C such that for every integer $m \geq 0$*

$$E[\|\varepsilon_m\|^2 \mid \mathcal{F}_m] \leq C\gamma_m^2,$$

and the series $\sum_m \varepsilon_m$ converges with probability 1.

Proof: Recall that $g_{i_n}(\theta_n)$ is bounded by Assumption 5 (MRP Parameterization) . Thus, for $n = t_m, \ldots, t_{m+1}-1$, we have $|\tilde{J}_{i_n}(\theta)| \leq C(t_{m+1}-t_m)$, for some constant C. Furthermore, by Assumption 5 (MRP Parameterization) , the derivative $\nabla g_i(\theta)$, and the quotient $\frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)}$, are bounded. It follows that

$$\|F_m(\theta_m)\| \leq C(t_{m+1} - t_m)^2,$$

for some new constant C. Using Lemma 4 , which states that the expectation $E_{\bar{\pi},\theta}[(t_{m+1} - t_m)^2]$ is bounded, the conditional variance of $F_m(\theta_m)$, given $\mathcal{F}_m$, is bounded and the first statement follows. Finally, using Eq.(B.2) together with Assumption 8 (Step Size 1) , which states that the series $\sum_m \gamma_m^2$ converges, it follows that the sequence ϵ_m is a martingale with bounded second moment, and therefore converges with probability 1 [Gal95]. □

More generally, let Φ be the set of all functions on $\Re^K$ that are twice differentiable and have bounded first and second derivatives. Given a function $\phi \in \Phi$, we define

$$\varepsilon_m(\phi) = \phi(\theta_{m+1}) - \phi(\theta_m) - \gamma_m \nabla\phi(\theta_m) \cdot \nabla\chi(\theta_m).$$

We then have the following result.

Lemma 11 *If $\phi \in \Phi$, then the series $\sum_m \varepsilon_m(\phi)$ converges with probability 1.*

Proof: Consider a sample path of the sequence (θ_m). We use the Taylor expansion of $\phi(\theta)$ at θ_m, and obtain

$$\begin{aligned}
\varepsilon_m(\phi) &= \phi(\theta_{m+1}) - \phi(\theta_m) - \gamma_m \nabla\phi(\theta_m) \cdot \nabla\chi(\theta_m) \\
&\leq \nabla\phi(\theta_m) \cdot (\theta_{m+1} - \theta_m) + C\|\theta_{m+1} - \theta_m\|^2 - \gamma_m \nabla\phi(\theta_m) \cdot \nabla\chi(\theta_m) \\
&= \nabla\phi(\theta_m) \cdot \varepsilon_m + C\|\theta_{m+1} - \theta_m\|^2,
\end{aligned}$$

where C is a constant related to the bound on the second derivatives of ϕ. A symmetric argument also yields

$$\nabla\phi(\theta_m) \cdot \varepsilon_m - C\|\theta_{m+1} - \theta_m\|^2 \leq \varepsilon_m(\phi).$$

As ϕ is an element of Φ, the derivative $\nabla\phi$ is bounded. Therefore, the same martingale argument as in the proof of Lemma 10 shows that the series $\sum_m \nabla\phi(\theta_m) \cdot \varepsilon_m$ converges with probability 1. Note that $\|\theta_{m+1} - \theta_m\| = \|\gamma_m \nabla\chi(\theta_m) + \varepsilon_m\|$, which yields

$$\|\theta_{m+1} - \theta_m\|^2 \leq 2\gamma_m^2 \|\nabla\chi(\theta_m)\|^2 + 2\|\varepsilon_m\|^2.$$

The sequence $(\|\nabla\chi(\theta_m)\|)$ is bounded by Lemma 2 and γ_m^2 is summable by Assumption 8 (Step Size 1). Furthermore, it is an easy consequence of Lemma 10 that ε_m is also square summable. We conclude that $\|\theta_{m+1} - \theta_m\|^2$ is summable, and the result follows. □

We have shown that the algorithm proposed in Section 4.3 is of the form

$$\theta_{m+1} = \theta_m + \gamma_m \nabla\chi(\theta_m) + \epsilon_m,$$

where ϵ_m is a summable sequence. This is a gradient algorithm with a stochastic noise term ϵ_m, which is summable. The proof of Proposition 5 follows then from a standard convergence argument for such algorithms ([BT97]).

Proof of Proposition 5 : We first show that $\chi(\theta_m)$ converges. Suppose the contrary. Then, as $\chi(\theta)$ is bounded by Lemma 2, there exists some $\epsilon > 0$, such that for every integer n, there exists an integer n', $n < n'$, with $\chi(\theta_{n'}) - \chi(\theta_n) \leq -\epsilon$. As we have that

$$\begin{aligned}\chi(\theta_{n'}) - \chi(\theta_n) &= \sum_{m=n}^{n'-1} \gamma_m \|\nabla\chi(\theta_m)\|^2 + \sum_{m=n}^{n'-1} \varepsilon_m(\chi) \\ &\geq \sum_{m=n}^{n'-1} \varepsilon_m(\chi),\end{aligned}$$

it follows that for every integer n, there exists an integer n', $n < n'$, such that

$$\left|\sum_{m=n}^{n'-1} \varepsilon_m(\chi)\right| \geq \epsilon.$$

This leads to a contradiction as, by Lemma 2, $\chi(\theta)$ is an element of Φ, and therefore, by Proposition 11, the series $\sum \varepsilon_m(\chi)$ converges.

Next, we show that $\liminf_{m\to\infty} \|\nabla\chi(\theta_m)\| = 0$. Suppose the contrary. Then, there exists some $\epsilon > 0$, and some N, such that $\|\nabla\chi(\theta_m)\| > \epsilon$ for all $m > N$. We have

$$\begin{aligned}\chi(\theta_{m+1}) &= \chi(\theta_m) + \gamma_m\|\nabla\chi(\theta_m)\|^2 + \varepsilon_m(\chi) \\ &\geq \chi(\theta_m) + \gamma_m\epsilon^2 + \varepsilon_m(\chi).\end{aligned} \tag{B.3}$$

Since $\varepsilon_m(\chi)$ is summable, but $\sum_m \gamma_m = \infty$, we conclude that $\chi(\theta_m)$ converges to infinity, which is a contradiction.

Next we show that $\limsup_{m\to\infty} \|\nabla\chi(\theta_m)\| = 0$. Suppose the contrary. Then, there exists a $\epsilon > 0$ such that $\|\nabla\chi(\theta_n)\| > \epsilon$ for infinitely many indices n. For any such n, let n' be the first subsequent time that $\|\nabla\chi(\theta_{n'})\| < \epsilon/2$. Then we have

$$\begin{aligned}\frac{\epsilon}{2} &\leq \|\nabla\chi(\theta_n)\| - \|\nabla\chi(\theta_{n'})\| \\ &\leq \|\nabla\chi(\theta_n) - \nabla\chi(\theta_{n'})\| \\ &\leq C\|\theta_n - \theta_{n'}\| \\ &= C\left\|\sum_{m=n}^{n'-1} \gamma_m\nabla\chi(\theta_m) + \sum_{m=n}^{n'-1} \varepsilon_m\right\| \\ &\leq C\sum_{m=n}^{n'-1} \gamma_m\|\nabla\chi(\theta_m)\| + C\left\|\sum_{m=n}^{n'-1} \varepsilon_m\right\|,\end{aligned}$$

for some constant C, as $\nabla^2\chi(\theta)$ is bounded by Lemma 2. Furthermore, when n is large enough, the summability of the sequence ε_m yields $C\|\sum_{m=n}^{n'-1} \varepsilon_m\| \leq \epsilon/4$. Let B be a bound on $\nabla\chi(\theta_m)$, which exists by Lemma 2. It follows that $\sum_{m=n}^{n'-1} \gamma_m \geq \epsilon/4CB$. By an argument similar to the one that led to Eq.(B.3), it is then easily shown that there exists some $\beta > 0$ such that

$$\chi(\theta_{n'}) \geq \chi(\theta_n) + \beta,$$

which contradicts the convergence of the sequence $\chi(\theta_m)$. $\square$

B.2 Proof of Proposition 12 and 13

In this section, we analyze the two algorithms proposed in Section 4.6, which are given by the iteration rules

$$\theta_{m+1} = \theta_m + \gamma_m F_{S^*,m}(\theta_m), \tag{B.4}$$

where $F_{S^*,m}(\theta_m)$ is defined by Eq.(4.9) in Section 4.5, and by

$$\theta_{m+1} = \theta_m + \gamma_m F_{\alpha,m}(\theta_m), \tag{B.5}$$

respectively, where $F_{\alpha,m}(\theta_m)$ is defined by Eq.(4.10) in Section 4.5.

Let us first consider the iteration given by Eq. (B.4), which we rewrite as

$$\theta_{m+1} = \theta_m + \gamma_m(\nabla\chi(\theta_m) + \delta_{S^*}(\theta_m)) + \varepsilon_m,$$

where

$$\delta_{S^*}(\theta_m) = f_{\bar{\pi},S^*}(\theta_m) - \nabla\chi(\theta_m),$$
$$\varepsilon_m = \gamma_m(F_{S^*,m}(\theta_m) - f_{\bar{\pi},S^*}(\theta_m)),$$

and

$$f_{\bar{\pi},S^*}(\theta_m) = E_{\bar{\pi},\theta_m}[F_{S^*,m}(\theta_m)].$$

The function $\delta_{S^*}(\theta)$ is bounded by Proposition 9. By Proposition 8 we have

$$E[\varepsilon_m \mid \mathcal{F}] = 0.$$

Furthermore, by using the same argument as in the proof of Lemma 10 in Section B.1, we can show that the series $\sum_m \varepsilon_m$ converges with probability 1.

Similarly, we can rewrite the iteration given by Eq.(B.5), as

$$\theta_{m+1} = \theta_m + \gamma_m(\nabla\chi(\theta_m) + \delta_\alpha(\theta_m)) + \varepsilon_m,$$

where

$$\delta_\alpha(\theta_m) = f_{\bar{\pi},\alpha}(\theta_m) - \nabla\chi(\theta_m),$$
$$\varepsilon_m = \gamma_m(F_{\alpha,m}(\theta_m) - f_{\bar{\pi},\alpha}(\theta_m)),$$

and

$$f_{\bar{\pi},\alpha}(\theta_m) = E_{\bar{\pi},\theta_m}[F_{\alpha,m}(\theta_m)],$$

and where the function $\delta_\alpha(\theta)$ is bounded, and the series $\sum_m \varepsilon_m$ converges with probability 1.

This leads us to consider the following iteration rule

$$\theta_{m+1} = \theta_m + \gamma_m \left(\nabla\chi(\theta_m) + \delta(\theta_m)\right) + \varepsilon_m, \tag{B.6}$$

for which we assume that the function $\delta(\theta)$ is bounded and the terms ε_m are such that the series $\sum_m \varepsilon_m$ converges. To prove Proposition 12 and 13, it then suffices to prove the following result.

Proposition 34 *Let Assumption 5 (MRP Parameterization), 6 (MRP Termination) and 8 (Step Size 1) hold. Furthermore, let (θ_m) be the sequence of parameter vectors generated by the iteration given by Eq.(B.6). Then, with probability 1, we have*

$$\liminf_{m\to\infty} \|\nabla\chi(\theta_m)\| \leq D,$$

where D is such that $\|\delta(\theta)\| \leq D$, for all $\theta \in \Re^K$.

Proof: First we point out that for $\varepsilon_m(\chi)$, defined by

$$\varepsilon_m(\chi) = \chi(\theta_{m+1}) - \chi(\theta_m) - \gamma_m \nabla\chi(\theta_m) \cdot \left(\nabla\chi(\theta_m) + \delta(\theta_m)\right),$$

the series $\sum_m \varepsilon_m(\chi)$ converges. This can be shown by the same argument as in the proof of Lemma 11 in Section B.1, together with the fact that, by Lemma 2, $\chi(\theta)$ has bounded first and second derivatives.

Assume that the proposition is not true. Then there exists a constant $\epsilon > 0$ and an integer m_0 such that

$$\|\nabla\chi(\theta_m)\| \geq D + \epsilon, \qquad \text{for all } m \geq m_0.$$

Then, for $m \geq m_0$, we have

$$
\begin{aligned}
\chi(\theta_{m+1}) &= \chi(\theta_m) + \gamma_m \nabla\chi(\theta_m) \cdot \Big(\nabla\chi(\theta_m) + \delta(\theta_m)\Big) + \varepsilon_m(\chi) \\
&\geq \chi(\theta_m) + \gamma_m \Big(\|\nabla\chi(\theta_m)\|^2 - \|\nabla\chi(\theta_m)\| D\Big) + \varepsilon_m(\chi) \\
&\geq \chi(\theta_m) + \gamma_m \|\nabla\chi(\theta_m)\|(D + \epsilon - D) + \varepsilon_m(\chi) \\
&\geq \chi(\theta_m) + \gamma_m \|\nabla\chi(\theta_m)\|\epsilon + \varepsilon_m(\chi) \\
&\geq \chi(\theta_m) + \gamma_m D\epsilon + \varepsilon_m(\chi).
\end{aligned}
$$

Since the series $\sum_m \gamma_m$ diverges by Assumption 8 (Step Size 1), we conclude that $\chi(\theta_m)$ diverges. This contradicts Lemma 2, and the result follows. □

B.3 Proof of Proposition 14

In this section, we prove the convergence of the algorithm introduced in Section 4.7, which is described by the iteration

$$
\begin{aligned}
\theta_{k+1} &= \theta_k + \gamma_k \Big(\nabla g_{i_k}(\theta_k) + g_{i_k}(\theta_k) z_k\Big), \\
z_{k+1} &= \begin{cases} 0, & \text{if } i_{k+1} = i^* \\ z_k + \dfrac{\nabla P_{i_k i_{k+1}}(\theta_k)}{P_{i_k i_{k+1}}(\theta_k)}, & \text{otherwise.} \end{cases}
\end{aligned}
$$

Using the augmented state $x = (i, z)$, we rewrite the update equation for the parameter vector θ as

$$\theta_{k+1} = \theta_k + \gamma_k R(x_k, \theta_k),$$

where

$$R(x_k, \theta_k) = \nabla g_{i_k}(\theta_k) + g_{i_k}(\theta_k) z_k. \tag{B.7}$$

Consider the sequence of states $(i_0, i_1, \ldots)$ visited during the execution of the algorithm. As in Section 4.3, let t_m be the mth time that the state i^* is visited in this sequence. We then have

$$
\begin{aligned}
\theta_{t_{m+1}} &= \theta_{t_m} + \sum_{k=t_m}^{t_{m+1}-1} \gamma_k R(x_k, \theta_k) \\
&= \theta_{t_m} + \tilde{\gamma}_m \frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]} + \varepsilon_m,
\end{aligned} \tag{B.8}
$$

where $\tilde{\gamma}_m$ and ε_m are given by

$$\tilde{\gamma}_m = \sum_{k=t_m}^{t_{m+1}-1} \gamma_k, \tag{B.9}$$

$$\varepsilon_m = \sum_{k=t_m}^{t_{m+1}-1} \gamma_k \left(R(x_k, \theta_k) - \frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]} \right). \tag{B.10}$$

The algorithm given by Eq.(B.8) can be interpreted as a scaled gradient algorithm with a stochastic noise term. We use the following steps to prove Proposition 14.

1. Show that the series $\sum_m \epsilon_m$ converges.
2. Show that the sequence $(\chi(\theta_m))$ converges, and that we have
$$\lim_{m\to\infty} \|\nabla\chi(\theta_m)\| = 0.$$
3. Show that the whole sequence $(\chi(\theta_k))$ converges, and that we have
$$\lim_{k\to\infty} \|\nabla\chi(\theta_k)\| = 0.$$

B.3.1 Preliminary Results

In this subsection, we derive a few lemmas, which will become useful for the convergence analysis in the following subsections.

Lemma 12 *Let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold. Furthermore, let $(i_0, i_1, i_2, ...)$ be the sequence of states visited in the algorithm given above, which is described by*

$$\theta_{k+1} = \theta_k + \gamma_k R(x_k, \theta_k),$$

and let t_m be the mth time we visit the state i^* in that sequence. Then there exists a constant C and an integer M_0, such that, for all m and k, $M_0 \leq m \leq k$, we have

$$\begin{aligned}
\|R(x_k, \theta_k)\| &\leq C(t_{m+1} - t_m), \qquad k \in \{t_m, \ldots, t_{m+1} - 1\}, \\
\|\theta_k - \theta_{t_m}\| &\leq C\gamma_{t_m}(t_{m+1} - t_m)^2, \qquad k \in \{t_m, \ldots, t_{m+1} - 1\}, \\
\|R(x_k, \theta_k) - R(x_k, \theta_{t_m})\| &\leq C\gamma_{t_m}(t_{m+1} - t_m)^3, \qquad k \in \{t_m, \ldots, t_{m+1} - 1\}.
\end{aligned}$$

Proof: As, by Assumption 5 (MRP Parameterization), the quotient $\frac{\nabla P_{ij}(\theta)}{P_{ij}(\theta)}$ is bounded, there exists a constant $C_z \geq 1$, such that

$$\|z_k\| \leq C_z(t_{m+1} - t_m).$$

Furthermore, by Assumption 5 (MRP Parameterization), the functions $\nabla g_i(\theta)$ and $g_i(\theta)$ are bounded. Since $(t_{m+1} - t_m) \geq 1$, there exists a constant C_R, such that

$$\begin{aligned}
\|R(x_k, \theta_k)\| &= \|\nabla g_{i_k}(\theta_k) + g_{i_k}(\theta_k) z_k\| \\
&\leq \|\nabla g_{i_k}(\theta_k)\| + |g_{i_k}(\theta_k)| \|z_k\| \\
&\leq \|\nabla g_{i_k}(\theta_k)\| + |g_{i_k}(\theta_k)| C_z(t_{m+1} - t_m) \\
&\leq C_R(t_{m+1} - t_m),
\end{aligned}$$

which establishes the first inequality in the lemma.

Using this result, we have

$$\begin{aligned}
\|\theta_k - \theta_{t_m}\| &= \|\sum_{n=t_m}^{k-1} \gamma_n R(x_n, \theta_n)\| \\
&\leq \sum_{n=t_m}^{k-1} \|\gamma_n R(x_n, \theta_n)\| \\
&\leq C_R(t_{m+1} - t_m) \sum_{n=t_m}^{k-1} \gamma_{t_m} \\
&= C_R \gamma_{t_m} (t_{m+1} - t_m)^2.
\end{aligned}$$

Finally, by Assumption 5 (MRP Parameterization), there exists a constant L, such that, for all $\theta, \theta' \in \Re^K$, we have

$$\|\nabla g_i(\theta) - \nabla g_i(\theta')\| \leq L\|\theta - \theta'\|, \qquad |g_i(\theta) - g_i(\theta')| \leq L\|\theta - \theta'\|.$$

It follows that

$$\begin{aligned}
\|R(x_k, \theta_k) - R(x_k, \theta_{t_m})\| &\leq \|\nabla g_{i_k}(\theta_k) - \nabla g_{i_k}(\theta_{t_m})\| + |\nabla g_{i_k}(\theta_k) - \nabla g_{i_k}(\theta_{t_m})|\|z_k\| \\
&\leq L\|\theta_k - \theta_{t_m}\| + L\|\theta_k - \theta_{t_m}\| C_z(t_{m+1} - t_m) \\
&\leq L\|\theta_k - \theta_{t_m}\|(1 + C_z(t_{m+1} - t_m)) \\
&\leq 2LC_R C_z \gamma_{t_m} (t_{m+1} - t_m)^3,
\end{aligned}$$

since we chose C_z to be larger than or equal to 1. □

Lemma 13 *Let Assumptions 5 (MRP Parameterization) and 10 (MRP Termination) hold. Also, let $(i_0, i_1, i_2, ...)$ be the sequence of states visited in the algorithm described above, and let t_m be the mth time we visit the state i^* in that sequence. Then for every integer $s > 0$ there exists a constant D_s such that*

$$E[(t_{m+1} - t_m)^s] \leq D_s.$$

Proof: Let N be the number of states in the state space S. By Assumption 5 (MRP Parameterization), and 10 (MRP Termination) there exists a constant $\rho > 0$, such that, for every state $i \in S$ and every collection $\{P_{\bar{\pi},1}, ..., P_{\bar{\pi},N}\}$ of N matrices from the set $\bar{\mathcal{P}}_{\bar{\pi}}$, we have

$$Q_{ii^*} > \rho,$$

where

$$Q = P_{\bar{\pi},1} \ldots \cdot P_{\bar{\pi},N}.$$

To see this call that the set $\overline{\mathcal{P}}$ is compact, and that the transition probabilities $P_{ij}(\theta)$ are continuous in θ by Assumption 5 (MRP Parameterization). This implies that there exists a constant C such that for all integers $n \geq 1$ we have

$$P((t_m - t_{m+1}) \geq n) \leq C\rho^{\frac{n}{N}}, \qquad n = 1, 2, ...,$$

and the lemma follows. $\square$

Lemma 14 *Let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold. Let $(i_0, i_1, i_2, ...)$ be the sequence of states visited in the algorithm described above, and let t_m be the mth time we visit the state i^* in that sequence. Then the following is true,*

(a) For any positive integer s, we have

$$E\left[\sum_{m=1}^{\infty} \gamma_{t_m}^2 (t_{m+1} - t_m)^s\right] < \infty.$$

(b) We have

$$\sum_{m=1}^{\infty} \tilde{\gamma}_m = \infty, \qquad \sum_{m=1}^{\infty} \tilde{\gamma}_m^2 < \infty,$$

with probability 1.

Proof: (a) From Lemma 13, and because γ_{t_m} is $\mathcal{F}_m$-measurable, we have

$$E[\gamma_{t_m}^2 (t_{m+1} - t_m)^s] = E\left[\gamma_{t_m}^2 E[(t_{m+1} - t_m)^s \mid \mathcal{F}_m]\right] \leq E[\gamma_{t_m}^2] D_s.$$

Hence,

$$\sum_{m=1}^{\infty} E[\gamma_{t_m}^2 (t_{m+1} - t_m)^s] \leq D_s \sum_{k=1}^{\infty} \gamma_k^2 < \infty,$$

and the result follows.

(b) By Assumption 8 (Step Size 1), we have

$$\sum_{m=1}^{\infty} \tilde{\gamma}_m = \sum_{k=1}^{\infty} \gamma_k = \infty.$$

Furthermore, by Assumption 9 (Step Size 2), the sequence γ_k is non increasing, and we have

$$\tilde{\gamma}_m^2 \leq \gamma_{t_m}^2 (t_{m+1} - t_m)^2.$$

Using part (a) of the lemma, we obtain that $\sum_{m=1}^{\infty} \tilde{\gamma}_m^2$ has finite expectation and is therefore finite with probability 1. $\square$

B.3.2 Summability of ε_k

This subsection is devoted to the proof that the series $\sum_m \varepsilon_m$ converges, where ε_m is defined by Eq.(B.10), i.e. we have

$$\varepsilon_m = \sum_{k=t_m}^{t_{m+1}-1} \gamma_k \left(R(x_k, \theta_k) - \frac{\nabla \chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]} \right).$$

We introduce some more notation. The evolution of the augmented state $x_k = (i_k, z_k)$ is affected by the fact that θ_k changes at each time step. Given the time t_m at which i^* is visited for the mth time, we define a "frozen" augmented state $x_k^F = (i_k^F, z_k^F)$ which evolves the same way as x_k except that θ_k is held fixed at θ_{t_m} until the next visit at i^*. More precisely, we let $x_{t_m}^F = x_{t_m}$, and , for $k \geq t_m + 1$, we let i_k^F evolve as a time-homogeneous Markov chain with transition probabilities $p_{ij}(\theta_{t_m})$. We also let $t_{m+1}^F = \min\{k > t_m \mid i_k^F = i^*\}$ be the first time after t_m that i_k^F is equal to i^*, and define z_{k+1}^F by the recursion

$$z_{k+1}^F = z_k^F + \frac{\nabla P_{i_k^F i_{k+1}^F}(\theta_{t_m})}{P_{i_k^F i_{k+1}^F}(\theta_{t_m})},$$

where $z_{t_m}^F = 0$. We have the following lemma.

Lemma 15 *Let Assumptions 5 (MRP Parameterization) and 6 (MRP Termination) hold, and let $(x_{t_m}^F, ..., x_{t_{m+1}^F}^F)$ be defined as above. Then there exists a constant C, such that, for $k \in \{t_m, \ldots, t_{m+1} - 1\}$, we have*

$$\|R(x_k^F, \theta_{t_m})\| = \|\nabla g_{i_k^F}(\theta_{t_m}) + g_{i_k^F}(\theta_{t_m}) z_k^F)\| \leq C(t_{m+1}^F - t_m).$$

This lemma can be proven using a similar argument as the one given in the proof of Lemma 12.

Using the definition of x_k^F, and t_{m+1}^F, we rewrite the noise term ε_m as

$$\begin{aligned} \varepsilon_m &= \sum_{k=t_m}^{t_{m+1}-1} \gamma_k \left(R(x_k, \theta_k) - \frac{\nabla \chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]} \right) \\ &= \varepsilon_m^{(1)} + \varepsilon_m^{(2)} + \varepsilon_m^{(3)} + \varepsilon_m^{(4)} + \varepsilon_m^{(5)}, \end{aligned}$$

where

$$\begin{aligned} \varepsilon_m^{(1)} &= \sum_{k=t_m}^{t_{m+1}-1} (\gamma_{t_m} - \gamma_k) \frac{\nabla \chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]}, \\ \varepsilon_m^{(2)} &= \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}^F-1} \left[R(x_k^F, \theta_{t_m}) - \frac{\nabla \chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]} \right], \\ \varepsilon_m^{(3)} &= \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}-1} \left[R(x_k, \theta_{t_m}) - \frac{\nabla \chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]} \right] \end{aligned}$$

$$
\begin{aligned}
& -\gamma_{t_m} \sum_{k=t_m}^{t_{m+1}^F-1} \left[R(x_k^F, \theta_{t_m}) - \frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]} \right], \\
\varepsilon_m^{(4)} &= \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}-1} \left[R(x_k, \theta_k) - R(x_k, \theta_{t_m}) \right], \\
\varepsilon_m^{(5)} &= \sum_{k=t_m}^{t_{m+1}-1} \left(\gamma_k - \gamma_{t_m} \right) R(x_k, \theta_k).
\end{aligned}
$$

We will show that each one of the series $\sum_m \varepsilon_m^{(n)}$, $n = 1, \ldots, 5$, converges with probability 1.

Lemma 16 *The series $\sum_m \varepsilon_m^{(1)}$ converges with probability 1.*

Proof: Let B be a bound on $\|\nabla\chi(\theta_{t_m})\|$, which exists by Lemma 2. Since we have that $E_{\bar{\pi},\theta}[T] \geq 1$ for every $\theta \in \Re^K$, it follows that $\left\| \frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]} \right\| \leq B$. Using Assumption 9 (Step Size 2), which states that there exists a positive integer p and a positive scalar A such that, for all positive integers n and t, we have

$$
\sum_{k=n}^{n+t} (\gamma_n - \gamma_k) \leq A t^p \gamma_n^2.
$$

Using this inequality, we obtain

$$
\begin{aligned}
\|\varepsilon_m^{(1)}\| &\leq \sum_{k=t_m}^{t_{m+1}-1} (\gamma_{t_m} - \gamma_k) \left\| \frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]} \right\| \\
&\leq B \sum_{k=t_m}^{t_{m+1}-1} (\gamma_{t_m} - \gamma_k) \\
&\leq BA\gamma_{t_m}^2 (t_{m+1} - t_m)^p.
\end{aligned}
$$

Therefore, by Lemma 14(a) the infinite sum $\sum_m \|\varepsilon_m^{(1)}\|$ has finite expectation, and is therefore finite with probability 1. $\square$

Lemma 17 *The series $\sum_m \varepsilon_m^{(2)}$ converges with probability 1.*

Proof: When the parameters θ is frozen to its value at time t_m, the total update $\sum_{k=t_m}^{t_{m+1}^F-1} R(x_k^F, \theta_{t_m})$ coincides with the update $F_m(\theta_m)$ of the algorithm studied in Section B.1. Using the discussion in the beginning of that section, we have

$$
E_{\bar{\pi},\theta_{t_m}} \left[\sum_{k=t_m}^{t_{m+1}^F-1} R(x_k^F, \theta_{t_m}) \mid \mathcal{F}_m \right] = E_{\bar{\pi},\theta_{t_m}} [F_m(\theta_m) \mid \mathcal{F}_m] = \nabla\chi(\theta_{t_m}).
$$

Furthermore, observe that

$$E_{\bar{\pi},\theta_{t_m}}\left[\sum_{k=t_m}^{t^F_{m+1}-1}\frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]} \mid \mathcal{F}_m\right]=\frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]}E_{\bar{\pi},\theta_{t_m}}[T]=\nabla\chi(\theta_{t_m}).$$

Thus, the expectation $E[\varepsilon_m^{(2)} \mid \mathcal{F}_m]$ is equal to 0. Furthermore, let B be a bound on $\nabla\chi(\theta_{t_m})$, which exists by Lemma 2. Since we have that $E_{\bar{\pi},\theta}[T]\geq 1$, it follows that $\left\|\frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]}\right\|\leq B$. Therefore, by Lemma 12, we have

$$\begin{aligned}
\|\varepsilon_m^{(2)}\|^2 &\leq \gamma_{t_m}^2\left(\sum_{k=t_m}^{t^F_{m+1}-1}\left(\|R(x_k^F,\theta_{t_m})\|+\|\frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]}\|\right)\right)^2\\
&\leq \gamma_{t_m}^2\left(\sum_{k=t_m}^{t^F_{m+1}-1}(C(t_{m+1}-t_m)+B)\right)^2\\
&\leq \gamma_{t_m}^2(C+B)^2(t_{m+1}-t_m)^4.
\end{aligned}$$

Therefore, by Lemma 13, there exists a constant D_4, such that

$$E[\varepsilon_m^{(2)}\|^2 \mid \mathcal{F}_m]\leq\gamma_{t_m}^2(C+B)^2D_4.$$

Thus, $\sum_m\varepsilon_m^{(2)}$ is martingale with bounded variance and, therefore, converges [Gal95]. □

Lemma 18 *The series $\sum_m\varepsilon_m^{(3)}$ converges with probability 1.*

Proof: The proof is based on a coupling argument. Using Assumption 5 (MRP Parameterization), there exists a constant L, such that for all $\theta,\theta'\in\Re^K$, and all states $i,j\in S$, we have

$$|P_{ij}(\theta)-P_{ij}(\theta')|\leq L\|\theta-\theta'\|.$$

Furthermore, by Lemma 12, there exists a constant C, such that for all $m\geq 0$, and all $k=t_m,\dots,t_{m+1}-1$, we have

$$\|\theta_k-\theta_{t_m}\|\leq C\gamma_{t_m}(t_{m+1}-t_m)^2.$$

It follows that for, $k=t_m,\dots,t_{m+1}-1$, the two processes x_k and x_k^F can be defined on the same probability space, such that

$$\begin{aligned}
P(i^F_{k+1}\neq i_{k+1} \mid i^F_k=i_k) &\leq \sum_{i,j\in S}|P_{ij}(\theta_k)-P_{ij}(\theta_{t_m})|\\
&\leq B\|\theta_k-\theta_{t_m}\|\\
&\leq D\gamma_{t_m}(t_{m+1}-t_m)^2, \qquad \text{(B.11)}
\end{aligned}$$

for some constants B and D.

We define $\mathcal{E}_m$ to be the following event,

$$\mathcal{E}_m = \{x_k^F \neq x_k \text{ for some } k = t_m, \ldots, t_{m+1}\}.$$

Using Eq.(B.11), we obtain

$$P(\mathcal{E}_m \mid t_m, t_{m+1}) \leq D \sum_{k=t_m}^{t_{m+1}-1} \gamma_{t_m}(t_{m+1} - t_m)^2 = D\gamma_{t_m}(t_{m+1} - t_m)^3.$$

Note that if the event $\mathcal{E}_m$ does not occur, then $\varepsilon_m^{(3)} = 0$. Thus,

$$E[\|\varepsilon_m^{(3)}\| \mid t_m, t_{m+1}] = P(\mathcal{E}_m \mid t_m, t_{m+1}) E[\|\varepsilon_m^{(3)}\| \mid t_m, t_{m+1}, \mathcal{E}_m].$$

Let B be a bound on $\nabla\chi(\theta_{t_m})$, which exists by Lemma 2. Since we have that $E_{\bar{\pi},\theta}[T] \geq 1$, it follows that $\left\|\frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]}\right\| \leq B$. Therefore, using Lemma 12 and 15, we have

$$\begin{aligned}
\|\varepsilon_m^{(3)}\| \;&\leq\; \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}-1} \left[\|R(x_k, \theta_{t_m})\| + \|\frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]}\|\right] \\
&\quad + \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}^F-1} \left[\|R(x_k^F, \theta_{t_m})\| + \|\frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]}\|\right], \\
&\leq\; \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}-1} [C'(t_{m+1} - t_m) + B] + \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}^F-1} \left[C'(t_{m+1}^F - t_m) + B\right], \\
&\leq\; \gamma_{t_m} C((t_{m+1} - t_m)^2 + (t_{m+1}^F - t_m)^2),
\end{aligned}$$

for some constants C and C'. We conclude that

$$E\left[\|\varepsilon_m^{(3)}\| \mid t_m, t_{m+1}, \mathcal{E}_m\right] \leq \gamma_{t_m} C\Big((t_{m+1} - t_m)^2 + E[(t_{m+1}^F - t_m)^2 \mid t_m, t_{m+1}, \mathcal{E}_m]\Big).$$

By the same argument we used in Lemma 13, it is easily shown that there exists a constant D_2, such that we have

$$\begin{aligned}
E[(t_{m+1}^F - t_m)^2 \mid t_m, t_{m+1}, \mathcal{E}_m]) \;&\leq\; 2E[(t_{m+1}^F - t_{m+1})^2 \mid t_{m+1}, \mathcal{E}_m] + 2(t_{m+1} - t_m)^2 \\
&\leq\; D_2 + 2(t_{m+1} - t_m)^2.
\end{aligned}$$

As $(t_{m+1} - t_m) \geq 1$, this implies that

$$E\left[(t_{m+1}^F - t_m)^2 \mid t_m, t_{m+1}, \mathcal{E}_m\right] \leq C(t_{m+1} - t_m)^2,$$

for some new constant C. By combining these inequalities, we obtain

$$E\left[\|\varepsilon_m^{(3)}\| \mid t_m, t_{m+1}, \mathcal{E}_m\right] \leq C\gamma_{t_m}(t_{m+1} - t_m)^2,$$

and

$$E\left[\|\varepsilon_m^{(3)}\| \mid t_m, t_{m+1}\right] \leq DC\gamma_{t_m}^2(t_{m+1} - t_m)^5.$$

Using Lemma 14(a), which states that the expectation of the infinite $\sum_m \gamma_{t_m}^2(t_{m+1} - t_m)^5$ is bounded, we conclude that the infinite sum $\sum_m \|\varepsilon_m^{(3)}\|$ has finite expectation, and is therefore finite with probability 1. □

Lemma 19 *The series* $\sum_m \varepsilon_m^{(4)}$ *converges with probability 1.*

Proof: By Lemma 12, we have

$$\begin{aligned}
\|\varepsilon_m^{(4)}\| &\leq \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}-1} \Big\| R(x_k, \theta_k) - R(x_k, \theta_{t_m}) \Big\|, \\
&\leq \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}-1} C\gamma_{t_m}(t_{m+1} - t_m)^3 \\
&= C\gamma_{t_m}^2 (t_{m+1} - t_m)^4,
\end{aligned}$$

for some constant C.

Using Lemma 14(a), we conclude that the infinite sum $\sum_m \|\varepsilon_m^{(4)}\|$ has finite expectation, and is therefore finite with probability 1. □

Lemma 20 *The series* $\sum_m \varepsilon_m^{(5)}$ *converges with probability 1.*

Proof: Using Assumption 9 (Step Size 2) and Lemma 12, we have

$$\begin{aligned}
\|\varepsilon_m^{(5)}\| &\leq \sum_{k=t_m}^{t_{m+1}-1} (\gamma_k - \gamma_{t_m}) \, \|R(x_k, \theta_k)\| \\
&\leq C(t_{m+1} - t_m) \sum_{k=t_m}^{t_{m+1}-1} (\gamma_{t_m} - \gamma_k) \\
&\leq AC\gamma_{t_m}^2 (t_{m+1} - t_m)^{(p+1)},
\end{aligned}$$

for some constants A and C, and for some positive integer p. Using Lemma 14(a), the infinite sum $\sum_m \|\varepsilon_m^{(5)}\|$ has finite expectation, and is therefore finite with probability 1. □

B.3.3 Convergence of $\chi(\theta_k)$ and $\nabla\chi(\theta_k)$

In the previous section, we analyzed the error term ε_m in the algorithm defined by

$$\begin{aligned}
\theta_{t_{m+1}} &= \theta_{t_m} + \sum_{k=t_m}^{t_{m+1}-1} \gamma_k R(x_k, \theta_k) \\
&= \theta_{t_m} + \tilde{\gamma}_m \frac{\nabla\chi(\theta_{t_m})}{E_{\bar{\pi},\theta_{t_m}}[T]} + \varepsilon_m. \qquad \text{(B.12)}
\end{aligned}$$

In particular, we showed that the series $\sum_m \varepsilon_m$ converges with probability 1. We make the following, additional observation regarding the iteration given by (B.12).

Lemma 21 *Let Assumptions 5 (MRP Parameterization), 6 (MRP Termination) and Assumption 8 (Step Size 1) hold, then, with probability 1, we have*

$$\sum_{m=0}^{\infty} \frac{\tilde{\gamma}_m}{E_{\bar{\pi},\theta_{t_m}}[T]} = \infty,$$

and

$$\sum_{m=0}^{\infty} \left(\frac{\tilde{\gamma}_m}{E_{\bar{\pi},\theta_{t_m}}[T]} \right)^2 < \infty.$$

Proof: By Lemma 4, we have that

$$T_{\max} = \sup_{\theta \in \Re^K} E_{\bar{\pi},\theta}[T] < \infty,$$

where we used the fact that the set $\overline{\mathcal{P}}$ is compact. Since $1 \leq E_{\bar{\pi},\theta_{t_m}}[T] \leq T_{\max}$, Lemma 14 implies that, with probability 1, we have

$$\sum_{m=0}^{\infty} \frac{\tilde{\gamma}_m}{E_{\bar{\pi},\theta_{t_m}}[T]} \geq T_{\max} \sum_{m=0}^{\infty} \tilde{\gamma}_m = \infty,$$

and

$$\sum_{m=0}^{\infty} \left(\frac{\tilde{\gamma}_m}{E_{\bar{\pi},\theta_{t_m}}[T]} \right)^2 \leq \sum_{m=0}^{\infty} \tilde{\gamma}_m^2 < \infty.$$

□

This implies that we can interpret the iteration (B.12) as a gradient algorithm with step sizes $(\frac{\tilde{\gamma}_m}{E_{\bar{\pi},\theta_{t_m}}[T]})$ that satisfy Assumption 8 (Step Size 1), with probability 1, and with a stochastic error term (ε_m) which is summable. Therefore, the iteration given by Eq.(B.12) is same form as the one given in Section 4.3, expect that now the step size sequence $\frac{\tilde{\gamma}_m}{E_{\bar{\pi},\theta_{t_m}}[T]}$ is stochastic, rather than deterministic. Nevertheless, it can be easily checked that, by the same argument as the one given to prove Proposition 5 in Section B.1, the sequence $\chi(\theta_{t_m})$ converges and we have

$$\lim_{m \to \infty} \|\nabla \chi(\theta_{t_m})\| = 0.$$

Having established this result, we are now ready to prove Proposition 14, e.g. to show that that the sequence $(\chi(\theta_k))$ converges and that we have

$$\lim_{k \to \infty} \|\nabla \chi(\theta_k)\| = 0.$$

Note that, in order to do that, it suffices to show that for every scalar $\epsilon > 0$, there exists an integer m_0, such that for all integers $m \geq m_0$, and all integers $k \in \{t_m, ..., t_{m+1}\}$, we have

$$|\chi(\theta_k) - \chi(\theta_{t_m})| \leq \epsilon, \quad \text{and} \quad \|\nabla \chi(\theta_k) - \nabla \chi(\theta_{t_m})\| \leq \epsilon.$$

As by Proposition 2, there exists a constant L such that for all $\theta, \theta' \in \Re^K$ we have

$$|\chi(\theta) - \chi(\theta')| \leq L\|\theta - \theta'\|, \quad \text{and} \quad \|\nabla\chi(\theta') - \nabla\chi(\theta')\| \leq L\|\theta - \theta'\|,$$

this is equivalent to showing that, for every $\epsilon > 0$, there exists with probability 1 an integer m_0, such that, for all $m \geq m_0$ and for all all integers $k \in \{t_m, ..., t_{m+1}\}$, we have

$$\|\theta_k - \theta_{t_m}\| \leq \epsilon.$$

This result is established by the following lemma.

Lemma 22 *Let φ_m be defined by*

$$\varphi_m = \sum_{k=t_m}^{t_{m+1}-1} \gamma_k \|R(x_k, \theta_k)\|$$

Then, with probability 1, we have

$$\lim_{m\to\infty} \varphi_m = 0$$

Proof: Using Lemma 12, it follows that there exists a constant C, such that

$$\begin{aligned}
\varphi_m &\leq \sum_{k=t_m}^{t_{m+1}-1} \gamma_k \|R(x_k, \theta_k)\| \\
&\leq \sum_{k=t_m}^{t_{m+1}-1} \gamma_k C(t_{m+1} - t_m) \\
&\leq \sum_{k=t_m}^{t_{m+1}-1} \gamma_{t_m} C(t_{m+1} - t_m) \\
&\leq \gamma_{t_m} C(t_{m+1} - t_m)^2.
\end{aligned}$$

Using Lemma 14(a), which states that the expectation of the infinite $\sum_m \gamma_{t_m}^2 (t_{m+1} - t_m)^4$ is bounded, we conclude that the infinite sum $\sum_m \varphi_m^2$ has finite expectation. This implies that, with probability 1, φ_m converges to 0. □

Appendix C

Proofs for Chapter 5

C.1 Proof of Proposition 20

In this appendix, we prove convergence of the algorithm

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

where

$$F_m(\theta_m, \tilde{\lambda}_m) = \sum_{n=t_m}^{t_{m+1}-1} \left(\tilde{v}_{i_n}(\theta_m, \tilde{\lambda}_m) \frac{\nabla P_{i_{n-1}i_n}(\theta_m)}{P_{i_{n-1}i_n}(\theta_m)} + \nabla g_{i_n}(\theta_m) \right),$$

$$\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{t_{m+1}-1} \left(g_{i_k}(\theta) - \tilde{\lambda} \right), \qquad n = t_m + 1, \ldots, t_{m+1} - 1,$$

and

$$\tilde{v}_{i_{t_m}}(\theta, \tilde{\lambda}) = 0.$$

For notational convenience, we define the augmented parameter vector $r_m = (\theta_m, \tilde{\lambda}_m)$, and write the update equations in the form

$$r_{m+1} = r_m + \gamma_m H_m(r_m),$$

where

$$H_m(r_m) = \begin{bmatrix} F_m(\theta_m, \tilde{\lambda}_m) \\ \eta \displaystyle\sum_{n=t_m}^{t_{m+1}-1} (g_{i_n}(\theta_m) - \tilde{\lambda}_m) \end{bmatrix}. \tag{C.1}$$

Let

$$\mathcal{F}_m = \{\theta_0, \tilde{\lambda}_0, i_0, i_1, \ldots, i_{t_m}\}$$

stand for the history of the algorithm up to and including time t_m. Using Proposition 2 and Eq.(5.14), we have

$$E[H_m(r_m) \mid \mathcal{F}_m] = h(r_m),$$

where

$$h(r) = \left[\begin{array}{c} E_\theta[T]\nabla\lambda(\theta) + G(\theta)(\lambda(\theta) - \tilde{\lambda}) \\ \eta E_\theta[T](\lambda(\theta) - \tilde{\lambda}) \end{array} \right],$$

and

$$G(\theta) = E_\theta \left[\sum_{n=t_m+1}^{t_{m+1}-1} (t_{m+1} - n) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} \right].$$

We then rewrite the algorithm in the form

$$r_{m+1} = r_m + \gamma_m h(r_m) + \varepsilon_m,$$

where

$$\varepsilon_m = \gamma_m (H_m(r_m) - h(r_m)).$$

Note that for ε_m we have

$$E[\varepsilon_m \mid \mathcal{F}_m] = 0.$$

The proof rests on the fact that ε_m is "small," in a sense to be made precise, which will then allow us to mimic the heuristic argument of Section 3.3.

C.1.1 Preliminaries

In this subsection, we establish a few useful bounds and characterize the behavior of ε_m.

Lemma 23

(a) The function $G(\theta)$ is well defined and bounded.

(b) The sequence $\tilde{\lambda}_m$ is bounded, with probability 1.

(c) The sequence $h(r_m)$ is bounded, with probability 1.

Proof:

(a) Note that

$$E_\theta \left[\sum_{n=t_m+1}^{t_{m+1}-1} \left\| (t_{m+1} - n) \frac{\nabla P_{i_{n-1}i_n}(\theta)}{P_{i_{n-1}i_n}(\theta)} \right\| \right] \leq C E_\theta[T^2],$$

where C is a bound on $\|\nabla P_{ij}(\theta)/P_{ij}(\theta)\|$, which exists by Assumption 5 (MRP Parameterization). The right-hand side is bounded by Lemma 4. It follows that the expectation defining $G(\theta)$, exists and is a bounded function of θ.

(b) Using Assumption 8 (Step Size 1), which states that the infinite sum $\sum_m \gamma_m^2$ is bounded, and Lemma 4, we obtain

$$E \left[\sum_{m=1}^{\infty} \gamma_m^2 (t_{m+1} - t_m)^2 \right] < \infty.$$

This implies that $\gamma_m(t_{m+1}-t_m)$ converges to zero, with probability 1. Note that

$$\tilde{\lambda}_{m+1} \leq (1-\gamma_m(t_{m+1}-t_m))\tilde{\lambda}_m + \gamma_m(t_{m+1}-t_m)C,$$

where C is a bound on $g_i(\theta)$ for all $i \in S$, which exists by Assumption 5 (MRP Parameterization). For large enough m, we have $\gamma_m(t_{m+1}-t_m) \leq 1$, and $\tilde{\lambda}_{m+1} \leq \max\{\tilde{\lambda}_m, C\}$, from which it follows that the sequence $\tilde{\lambda}_m$ is bounded above. By a similar argument, the sequence $\tilde{\lambda}_m$ is also bounded below.

(c) Consider the term

$$h(r)) = \begin{bmatrix} E_\theta[T]\nabla\lambda(\theta) + G(\theta)(\lambda(\theta)-\tilde{\lambda}) \\ \eta E_\theta[T](\lambda(\theta)-\tilde{\lambda}) \end{bmatrix}.$$

Lemma 4 and part (a) of this lemma show that $E_{\theta_m}[T]$ and $G(\theta_m)$ are bounded. Also, $\lambda(\theta_m)$ is bounded by Lemma 4. Furthermore, $\nabla\lambda(\theta_m)$ is bounded by Lemma 4. Using also part (b) of this lemma, the result follows. □

Lemma 24 *There exists a constant C (which is random but finite with probability 1) such that, for all integers $m \geq 1$, we have*

$$E[\|\varepsilon_m\|^2 \mid \mathcal{F}_m] \leq C\gamma_m^2,$$

and the series $\sum_m \varepsilon_m$ converges with probability 1.

Proof: Recall that $g_{i_m}(\theta_m)$ and $\tilde{\lambda}_m$ are bounded with probability 1 by Assumption 5 (MRP Parameterization) and by Lemma 23(d)). Thus, for $n = t_m, \ldots, t_{m+1}-1$, we have $|\tilde{v}_{i_n}(\theta,\tilde{\lambda})| \leq C(t_{m+1}-t_m)$, for some constant C. Using this bound in the definition of

$$F_m(\theta_m,\tilde{\lambda}_m) = \sum_{n=t_m}^{t_{m+1}-1}\left(\tilde{v}_{i_n}(\theta_m,\tilde{\lambda}_m)\frac{\nabla P_{i_{n-1}i_n}(\theta_m)}{P_{i_{n-1}i_n}(\theta_m)} + \nabla g_{i_n}(\theta_m)\right),$$

we see that for almost all sample paths, we have

$$\|F_m(\theta_m,\tilde{\lambda}_m)\| \leq C(t_{m+1}-t_m)^2,$$

for some new constant C. Using Lemma 4(b), which states that the expectation $E_{\theta_m}[(t_{m+1}-t_m)^2]$ is bounded, the conditional variance of $F_m(\theta_m,\tilde{\lambda}_m)$, given $\mathcal{F}_m$, is bounded. Similar arguments also apply to the last component of $H_m(r_m)$. Since $\varepsilon_m = \gamma_m(H_m(r_m) - E[H_m(r_m) \mid \mathcal{F}_m])$, the first statement follows.

Fix a positive integer c and consider the sequence

$$w_n^c = \sum_{m=1}^{\min\{M(c),n\}} \varepsilon_m,$$

where $M(c)$ is the first time m such that $E[\|\varepsilon_m\|^2 \mid \mathcal{F}_m] > c\gamma_m^2$. The sequence w_n^c is a martingale with bounded second moment, and therefore converges with probability 1. This is true for every positive integer c. For (almost) every sample path, there exists some c such that $M(c) = \infty$. After discarding a countable union of sets of measure zero (for each c, the set of sample paths for which w_n^c does not converge), it follows that for (almost) every sample path, $\sum_m \varepsilon_m$ converges. □

We observe the following consequences of Lemma 24. First, ε_m converges to zero with probability 1. Since γ_m also converges to zero and the sequence $h(r_m)$ is bounded, we conclude that

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

with probability 1.

C.1.2 Convergence of $\tilde{\lambda}_m$ and $\lambda(\theta_m)$

In this subsection, we prove that $\tilde{\lambda}_m$ and $\lambda(\theta_m)$ converge to a common limit. The flow of the proof is similar to the heuristic argument of Section 5.3.

We will be using a few different Lyapunov functions to analyze the behavior of the algorithm in different "regions." The lemma below involves a generic Lyapunov function ϕ and characterizes the changes in $\phi(r)$ caused by the updates

$$r_{m+1} = r_m + \gamma_m h(r_m) + \varepsilon_m.$$

Let the set $\mathcal{D}_c$ be defined by

$$\mathcal{D}_c = \{(\theta, \tilde{\lambda}) \in \Re^{K+1} \mid |\tilde{\lambda}| \leq c\}.$$

We are interested in Lyapunov functions ϕ that are twice differentiable and have, for every c, bounded first and second derivatives on $\mathcal{D}_c$ for every c. Let Φ be the set of all such Lyapunov functions. For any $\phi \in \Phi$, we define

$$\varepsilon_m(\phi) = \phi(r_{m+1}) - \phi(r_m) - \gamma_m \nabla\phi(r_m) \cdot h(r_m),$$

where for any two vectors a, b, we use $a \cdot b$ to denote their inner product.

Lemma 25 *If $\phi \in \Phi$, then the series $\sum_m \varepsilon_m(\phi)$ converges with probability 1.*

Proof: Consider a sample path of the random sequence $\{r_m\}$. Using part (c) of Lemma 23, and after discarding a set of zero probability, there exists some c such that $r_m \in \mathcal{D}_c$ for all m. We use the Taylor expansion of $\phi(r)$ at r_m, and obtain

$$\begin{aligned} \varepsilon_m(\phi) &= \phi(r_{m+1}) - \phi(r_m) - \gamma_m \nabla\phi(r_m) \cdot h(r_m) \\ &\leq \nabla\phi(r_m) \cdot (r_{m+1} - r_m) + M\|r_{m+1} - r_m\|^2 - \gamma_m \nabla\phi(r_m) \cdot h(r_m) \\ &= \nabla\phi(r_m) \cdot \varepsilon_m + M\|r_{m+1} - r_m\|^2, \end{aligned}$$

where M is a constant related to the bound on the second derivatives of $\phi(\cdot)$ on the set $\mathcal{D}_c$. A symmetric argument also yields

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

Using the boundedness of $\nabla\phi$ on the set $\mathcal{D}_c$, the same martingale argument as in the proof of Lemma 24 shows that the series $\sum_m \nabla\phi(r_m) \cdot \varepsilon_m$ converges with probability 1. Note that $\|r_{m+1} - r_m\| = \|\gamma_m h(r_m) + \varepsilon_m\|$, which yields

$$\|r_{m+1} - r_m\|^2 \leq 2\gamma_m^2\|h(r_m)\|^2 + 2\|\varepsilon_m\|^2.$$

The sequence $h(r_m)$ is bounded by Lemma 23 and γ_m^2 is summable by Assumption 8 (Step Size 1). Furthermore, it is an easy consequence of Lemma 24 that ε_m is also square summable. We conclude that $\|r_{m+1} - r_m\|$ is square summable, and the result follows. □

From now on, we will concentrate on a single sample path for which the sequences ε_m and $\varepsilon_m(\phi)$ (for the Lyapunov functions to be considered) are summable. Accordingly, we will be omitting the "with probability 1" qualification.

The next lemma shows that if the error $\tilde{\lambda}_m - \lambda(\theta_m)$ in estimating the average reward is positive but small, then it tends to decrease. The proof uses $\tilde{\lambda} - \lambda(\theta)$ as a Lyapunov function.

Lemma 26 *Let L be such that $\|G(\theta)\| \leq L$ for all θ, and let*

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

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

$$\nabla\phi(r) \cdot h(r) \leq 0.$$

Proof: The fact that $\phi \in \Phi$ is a consequence of Lemma 4. We now have

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

Using the inequality $|a \cdot b| \leq \|a\|^2 + \|b\|^2$, to bound the last term, and the fact $E_\theta[T] \geq 1$, we obtain

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

which is non positive as long as $0 \leq \tilde{\lambda} - \lambda(\theta) \leq \eta/L^2$. □

In the next two lemmas, we establish that if $|\tilde{\lambda}_m - \lambda(\theta_m)|$ remains small during a certain time interval, then $\tilde{\lambda}_m$ cannot decrease by much. We first introduce a Lyapunov function that captures the behavior of the algorithm when $\tilde{\lambda} \approx \lambda(\theta)$.

Lemma 27 *As in Lemma 26, let L be such that $\|G(\theta)\| \leq L$. 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

$$\nabla\phi(r) \cdot h(r) \geq 0.$$

Proof: The fact that $\phi \in \Phi$ is a consequence of Lemma 4. We have

$$\nabla_\theta \phi(\theta, \tilde{\lambda}) = \Big(1 - (2L^2/\eta)(\lambda(\theta) - \tilde{\lambda})\Big)\nabla\lambda(\theta),$$

and

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

Therefore, assuming that $|\lambda(\theta) - \tilde{\lambda}| \leq \eta/4L^2$, and using the Schwartz inequality, we obtain

$$\begin{aligned}
\nabla\phi(r) \cdot h(r) &= \Big(1 - (2L^2/\eta)(\lambda(\theta) - \tilde{\lambda})\Big)\Big(\|\nabla\lambda(\theta)\|^2 E_\theta[T] + (\lambda(\theta) - \tilde{\lambda})G(\theta)\cdot\nabla\lambda(\theta)\Big) \\
&\quad + 2L^2(\lambda(\theta) - \tilde{\lambda})^2 E_\theta[T] \\
&\geq \frac{1}{2}\|\nabla\lambda(\theta)\|^2 - \frac{3}{2}|\lambda(\theta) - \tilde{\lambda}| L \|\nabla\lambda(\theta)\| + 2L^2(\lambda(\theta) - \tilde{\lambda})^2 \\
&\geq 0.
\end{aligned}$$

□

Lemma 28 *Consider the same function ϕ as in Lemma 27, 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

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

Then,

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

Proof: Using Lemma 27, we have

$$\nabla\phi(r_m) \cdot h(r_m) \geq 0, \qquad m = n, \ldots, n'-1.$$

Therefore, for $m = n, \ldots, n'-1$, we have

$$\begin{aligned}
\phi(r_{m+1}) &= \phi(r_m) + \gamma_m \nabla\phi(r_m)\cdot h(r_m) + \varepsilon_m(\phi) \\
&\geq \phi(r_m) + \varepsilon_m(\phi),
\end{aligned}$$

and

$$\phi(r_{n'}) \geq \phi(r_n) + \sum_{m=n}^{n'-1} \varepsilon_m(\phi). \tag{C.2}$$

Note that $|\phi(r_n) - \tilde{\lambda}_n| \leq (L^2\alpha^2/\eta) + \alpha$, and $|\phi(r_{n'}) - \tilde{\lambda}_{n'}| \leq (L^2\alpha^2/\eta) + \alpha$. Using these inequalities in Eq.(C.2), we obtain the desired result. □

Lemma 29 *We have* $\liminf_{m\to\infty} |\lambda(\theta_m) - \tilde{\lambda}_m| = 0$.

Proof: Suppose that the result is not true, and we will derive a contradiction. Since $\lambda(\theta_{m+1}) - \lambda(\theta_m)$ and $\tilde{\lambda}_{m+1} - \tilde{\lambda}_m$ converge to zero, as a consequence of Lemma 24, there exists a scalar $\epsilon > 0$ and an integer n, such that either $\lambda(\theta_m) - \tilde{\lambda}_m > \epsilon$, or $\lambda(\theta_m) - \tilde{\lambda}_m < -\epsilon$, for all $m > n$. Without loss of generality, let us consider the first possibility.

Recall that the update equation for $\tilde{\lambda}$ is of the form

$$\tilde{\lambda}_{m+1} = \tilde{\lambda}_m + \eta\gamma_m E_{\theta_m}[T](\lambda(\theta_m) - \tilde{\lambda}_m) + \delta_m,$$

where δ_m is the last component of the vector ε_m, which is summable by Lemma 24. Furthermore we have $E_{\theta_m}[T] \geq 1$. Given that $\lambda(\theta_m) - \tilde{\lambda}_m$ stays above ϵ, the sequence $\eta\gamma_m(\lambda(\theta_m) - \tilde{\lambda}_m)$ sums to infinity. As δ_m is summable, we conclude that $\tilde{\lambda}_m$ converges to infinity, which contradicts the fact that it is bounded by Lemma 23. □

The next lemma shows that the condition $\lambda(\theta_m) \geq \tilde{\lambda}_m$ is satisfied, in the limit.

Lemma 30 *We have* $\liminf_{m\to\infty}(\lambda(\theta_m) - \tilde{\lambda}_m) \geq 0$.

Proof: Suppose the contrary. Then, there exists some $\epsilon > 0$ such that the inequality

$$\tilde{\lambda}_m - \lambda(\theta_m) > \epsilon$$

holds infinitely often. Let $\beta = \min\{\epsilon, \eta/L^2\}$, where L is the constant of Lemma 26. Using Lemma 29, we conclude that $\tilde{\lambda}_m - \lambda(\theta_m)$ crosses infinitely often from a value smaller than $\beta/3$ to a value larger than $2\beta/3$. In particular, there exist infinitely many pairs n, n', with $n' > n$, such that

$$0 < \tilde{\lambda}_n - \lambda(\theta_n) < \frac{1}{3}\beta, \qquad \tilde{\lambda}_{n'} - \lambda(\theta_{n'}) > \frac{2}{3}\beta,$$

and

$$\frac{1}{3}\beta \leq \tilde{\lambda}_m - \lambda(\theta_m) \leq \frac{2}{3}\beta, \qquad m = n+1, \ldots, n'-1.$$

We use the Lyapunov function

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

and note that

$$\phi(r_{n'}) \geq \phi(r_n) + \frac{\beta}{3}. \tag{C.3}$$

For $m = n, \ldots, n'-1$, we have $0 < \tilde{\lambda} - \lambda(\theta) < \beta \leq \eta/L^2$. Lemma 26 applies and shows that $\nabla\phi(r_m) \cdot h(r_m) \leq 0$. Therefore,

$$\phi(r_{n'}) = \phi(r_n) + \sum_{m=n}^{n'-1} \Big(\gamma_m \nabla\phi(r_m) \cdot h(r_m) + \varepsilon_m(\phi)\Big) \leq \phi(r_n) + \sum_{m=n}^{n'-1} \varepsilon_m(\phi).$$

By Lemma 25 the series $\sum_m \varepsilon_m(\phi)$ converges, which implies that $\sum_{m=n}^{n'-1} \varepsilon_m(\phi)$ becomes arbitrarily small. This contradicts Eq.(C.3) and completes the proof. □

We now continue with the central step in the proof, which is the proof that $\lim_{m\to\infty}(\lambda(\theta_m)-\tilde{\lambda}_m)=0$. Using Lemma 30, it suffices to show that we cannot have $\limsup_{m\to\infty}(\lambda(\theta_m)-\tilde{\lambda}_m)>0$. The main idea is the following. Whenever $\lambda(\theta_m)$ becomes significantly larger than $\tilde{\lambda}_m$, then $\tilde{\lambda}_m$ is bound to increase significantly. On the other hand, by Lemma 28, whenever $\lambda(\theta_m)$ is approximately equal to $\tilde{\lambda}_m$, then $\tilde{\lambda}_m$ cannot decrease by much. Since $\tilde{\lambda}_m$ is bounded, this will imply that $\lambda(\theta_m)$ can become significantly larger than $\tilde{\lambda}_m$ only a finite number of times.

Lemma 31 *We have* $\lim_{m\to\infty}(\lambda(\theta_m)-\tilde{\lambda}_m)=0$.

Proof: We will assume the contrary and derive a contradiction. By Lemma 30, we have $\liminf_{m\to\infty}(\lambda(\theta_m)-\tilde{\lambda}_m)\geq 0$. So if the desired result is not true, we must have $\limsup_{m\to\infty}(\lambda(\theta_m)-\tilde{\lambda}_m)>0$, which we will assume to be the case. In particular, there is some $A>0$ such that $\lambda(\theta_m)-\tilde{\lambda}_m>A$, infinitely often. Without loss of generality, we assume that $A\leq\eta/4L^2$, where L is the constant of Lemma 27. Let $\alpha>0$ be some small constant (with $\alpha<A/2$), to be specified later. Using Lemma 30, we have $\lambda(\theta_m)-\tilde{\lambda}_m>-\alpha$ for all large enough m. In addition, by Lemma 8, the condition $|\lambda(\theta_m)-\tilde{\lambda}_m|\leq\alpha$ holds infinitely often. Thus, the algorithm can be broken down into a sequence of cycles, where in the beginning and at the end of each cycle we have $|\lambda(\theta_m)-\tilde{\lambda}_m|\leq\alpha$, while the condition $\lambda(\theta_m)-\tilde{\lambda}_m>A$ holds at some intermediate time in the cycle.

We describe the stages of such a cycle more precisely. A typical cycle starts at some time N with $|\lambda(\theta_N)-\tilde{\lambda}_N|\leq\alpha$. Let n'' be the first time after time N that $\lambda(\theta_{n''})-\tilde{\lambda}_{n''}>A$. Let n' be the last time before n'' such that $\lambda(\theta_{n'})-\tilde{\lambda}_{n'}<A/2$. Let also n be the last time before n' such that $\lambda(\theta_n)-\tilde{\lambda}_n<\alpha$. Finally, let n''' be the first time after n'' such that $|\lambda(\theta_{n'''})-\tilde{\lambda}_{n'''}|<\alpha$. The time n''' is the end of the cycle and marks the beginning of a new cycle.

Recall that the changes in θ_m and $\tilde{\lambda}_m$ converge to zero. For this reason, by taking N to be large enough, we can assume that $\lambda(\theta_n)-\tilde{\lambda}_n\geq 0$. To summarize our construction, we have $N<n<n'<n''<n'''$, and

$$
\begin{aligned}
&|\lambda(\theta_N)-\tilde{\lambda}_N|<\alpha,\\
&0\leq\lambda(\theta_n)-\tilde{\lambda}_n<\alpha,\\
&|\lambda(\theta_m)-\tilde{\lambda}_m|\leq A, \qquad m=N,\ldots,n''-1,\\
&\lambda(\theta_{n'})-\tilde{\lambda}_{n'}<\frac{A}{2},\\
&\lambda(\theta_{n''})-\tilde{\lambda}_{n''}>A\\
&\alpha\leq\tilde{\lambda}(\theta_m)-\tilde{\lambda}_m\leq A, \qquad m=n+1,\ldots,n''-1,\\
&\frac{A}{2}\leq\tilde{\lambda}(\theta_m)-\tilde{\lambda}_m\leq A, \qquad m=n'+1,\ldots,n''-1,\\
&\alpha\leq\lambda(\theta_m)-\tilde{\lambda}_m, \qquad m=n'',\ldots,n'''-1.
\end{aligned}
$$

Our argument will use the Lyapunov functions

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

where L is as in Lemma 30, and

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

We have

$$\varepsilon_m(\phi) = \phi(r_{m+1}) - \phi(r_m) - \gamma_m \nabla\phi(r_m) \cdot h(r_m),$$

and we define $\varepsilon_m(\psi)$ by a similar formula. By Lemma 25, the series $\sum_m \varepsilon_m(\phi)$ and $\sum_m \varepsilon_m(\psi)$ converge. Also, let

$$\delta_m = \tilde{\lambda}_{m+1} - \tilde{\lambda}_m - \eta\gamma_m E_{\theta_m}[T](\lambda(\theta_m) - \tilde{\lambda}_m).$$

We observe that δ_m is the last component of ε_m and therefore, the series $\sum_m \delta_m$ converges and $\lim_{m\to\infty} \delta_m = 0$. Finally, let C be a constant such that $|\nabla\psi(r_m) \cdot h(r_m)| \leq C$, for all m, which exists because $\psi \in \Phi$ and because the sequences $h(r_m)$ and $\tilde{\lambda}_m$ are bounded by Lemma 23.

Using the above observations, we see that if the beginning time N of a cycle is chosen large enough, then for any k, k' such that $N \leq k \leq k'$, we have

$$\begin{aligned} \gamma_k C &\leq \frac{A}{32}, \\ \left|\sum_{m=k}^{k'} \varepsilon_m(\phi)\right| &\leq \frac{A^2}{96C}, \\ \left|\sum_{m=k}^{k'} \varepsilon_m(\psi)\right| &\leq \frac{A}{32}, \\ \left|\sum_{m=k}^{k'} \delta_m\right| &\leq \frac{A^2}{8C}. \end{aligned}$$

Finally, we assume that α has been chosen small enough so that

$$2(\alpha + (L^2\alpha^2/\eta)) \leq \frac{A^2}{96C}.$$

Using the fact that $\lambda(\theta_{n'+1}) - \tilde{\lambda}_{n'+1} \geq A/2$, we have

$$\lambda(\theta_{n'}) - \tilde{\lambda}_{n'} = \lambda(\theta_{n'+1}) - \tilde{\lambda}_{n'+1} + \gamma_{n'}\nabla\psi(r_{n'}) \cdot h(r_{n'}) + \varepsilon_{n'}(\psi) \geq \frac{A}{2} - \frac{A}{16}.$$

Furthermore, we have

$$\begin{aligned} \frac{A}{2} &\leq \Big((\lambda(\theta_{n''}) - \tilde{\lambda}_{n''}) - (\lambda(\theta_{n'}) - \tilde{\lambda}_{n'})\Big) \\ &= -\psi(r_{n''}) + \psi(r_{n'}) \\ &= -\sum_{m=n'}^{n''-1} \gamma_m \nabla\psi(r_m) \cdot h(r_m) - \sum_{m=n'}^{n''-1} \varepsilon_m(\psi) \\ &\leq \sum_{m=n'}^{n''-1} \gamma_m C + \frac{A}{32}, \end{aligned}$$

which implies that

$$\sum_{m=n'}^{n''-1} \gamma_m \geq \frac{A}{2C} - \frac{A}{32C}.$$

Then,

$$\begin{aligned}
\tilde{\lambda}_{n'''} &= \tilde{\lambda}_n + \sum_{m=n}^{n'''-1} \gamma_m E_{\theta_m}[T](\lambda(\theta_m) - \tilde{\lambda}_m) + \sum_{m=n}^{n'''-1} \delta_m \\
&\geq \tilde{\lambda}_n + \sum_{m=n'}^{n''-1} \gamma_m(\lambda(\theta_m) - \tilde{\lambda}_m) + \sum_{m=n}^{n'''-1} \delta_m \\
&\geq \tilde{\lambda}_n + \left(\frac{A}{2C} - \frac{A}{32C}\right)\left(\frac{A}{2} - \frac{A}{16}\right) - \frac{A^2}{8C} \\
&\geq \tilde{\lambda}_n + \frac{A^2}{24C}.
\end{aligned}$$

We have shown so far that $\tilde{\lambda}_m$ has a substantial increase between time n and n'''. We now show that $\tilde{\lambda}_m$ can only have a small decrease in the time between N and n. Indeed, by Lemma 7, we have

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

By combining these two properties, we obtain

$$\begin{aligned}
\tilde{\lambda}_{n'''} &\geq \tilde{\lambda}_N - 2(\alpha + (L\alpha^2/\eta)) - \frac{A^2}{96C} + \frac{A^2}{24C} \\
&\geq \tilde{\lambda}_N + \frac{A^2}{48C}.
\end{aligned}$$

We have shown that $\tilde{\lambda}_m$ increases by a positive amount during each cycle. Since $\tilde{\lambda}_m$ is bounded above, this proves that there can only be a finite number of cycles, and a contradiction has been obtained. □

Lemma 32 *The sequences $\tilde{\lambda}_m$ and $\lambda(\theta_m)$ converge.*

Proof: Consider the function $\phi(r) = \lambda(\theta) - (L^2/\eta)(\lambda(\theta) - \tilde{\lambda})^2$, and the same constant L as in Lemma 27. Let α be a scalar such that $0 < \alpha \leq 1/(4L^2/\eta)$. By the preceding lemma and by Lemma 25, there exists some N such that if $N \leq n \leq n'$, we have

$$|\lambda(\theta_n) - \tilde{\lambda}_n| \leq \alpha,$$

and

$$\left|\sum_{m=n}^{n'-1} \varepsilon_m(\phi)\right| \leq \alpha.$$

Using Lemma 27, we have

$$\phi(\theta_{n'}) \geq \phi(\theta_n) + \sum_{m=n}^{n'-1} \varepsilon_m(\phi) \geq \phi(\theta_n) - \alpha, \qquad N \leq n \leq n',$$

which we can rewrite as

$$\lambda(\theta_{n'}) - (L^2/\eta)(\lambda(\theta_{n'}) - \tilde{\lambda}_{n'})^2 \geq \lambda(\theta_n) - (L^2/\eta)(\lambda(\theta_n) - \tilde{\lambda}_n)^2 - \alpha,$$

which implies that

$$\lambda(\theta_{n'}) \geq \lambda(\theta_n) - (L^2\alpha^2/\eta) - \alpha, \qquad N \leq n \leq n'.$$

Therefore, we have

$$\liminf_{n'\to\infty} \lambda(\theta_{n'}) \geq \lambda(\theta_n) - (L^2\alpha^2/\eta) - \alpha, \qquad N \leq n,$$

which implies that

$$\liminf_{m\to\infty} \lambda(\theta_m) \geq \limsup_{m\to\infty} \lambda(\theta_m) - (L^2\alpha^2/\eta) - \alpha.$$

Since α can be chosen arbitrarily small, we have $\liminf_{m\to\infty} \lambda(\theta_m) \geq \limsup_{m\to\infty} \lambda(\theta_m)$, and since the sequence $\lambda(\theta_m)$ is bounded, we conclude that it converges. Using also Lemma 10, it follows that $\tilde{\lambda}_m$ converges as well. □

C.1.3 Convergence of $\nabla\lambda(\theta_m)$

In the preceding subsection, we have shown that $\lambda(\theta_m)$ and $\tilde{\lambda}_m$ converge to a common limit. It now remains to show that $\nabla\lambda(\theta_m)$ converges to zero.

Since $\lambda(\theta_{t_m}) - \tilde{\lambda}_{t_m}$ converges to zero, the algorithm is of the form

$$\theta_{m+1} = \theta_m + \gamma_m E_{\theta_m}[T](\nabla\lambda(\theta_m) + e_m) + \epsilon_m,$$

where e_m converges to zero and ϵ_m is a summable sequence. This is a gradient method with errors, similar to the methods considered in [Del96] and [BT97]. However, [Del96] assumes the boundedness of the sequence of iterates, and the results of [BT97] do not include the term e_m. Thus, while the situation is very similar to that considered in these references, a separate proof is needed.

Proof of Proposition 20: We will first show that $\liminf_{m\to\infty} \|\nabla\lambda(\theta_m)\| = 0$. Suppose the contrary. Then, there exists some $\epsilon > 0$ and some N such that $\|\nabla\lambda(\theta_m)\| > \epsilon$ for all $m > N$. In addition, by taking N large enough, we can also assume that $\|e_m\| \leq \epsilon/2$. Then, it is easily checked that

$$\nabla\lambda(\theta_m) \cdot (\nabla\lambda(\theta_m) + e_m) \geq \frac{\epsilon^2}{2}.$$

Let $\phi(r) = \lambda(\theta)$. Note that $\phi \in \Phi$. We have

$$\begin{aligned} \lambda(\theta_{m+1}) &= \lambda(\theta_m) + \gamma_m E_{\theta_m}[T] \nabla\lambda(\theta_m) \cdot (\nabla\lambda(\theta_m) + e_m) + \varepsilon_m(\phi) \\ &\geq \lambda(\theta_m) + \gamma_m \frac{\epsilon^2}{2} + \varepsilon_m(\phi). \end{aligned} \tag{C.4}$$

Since $\varepsilon_m(\phi)$ is summable by Lemma 25, but the series $\sum_m \gamma_m$ diverges by Assumption 8 (Step Size 1), we conclude that the sequence $\lambda(\theta_m)$ diverges, which is a contradiction to Lemma 4.

Next we show that $\limsup_{m\to\infty} \|\nabla\lambda(\theta_m)\| = 0$. Suppose the contrary. Then, there exists some $\epsilon > 0$ such that $\|\nabla\lambda(\theta_n)\| > \epsilon$ for infinitely many indices n. For any such n, let n' be the first subsequent time that $\|\nabla\lambda(\theta_{n'})\| < \epsilon/2$.

Then, we have

$$\begin{aligned}
\frac{\epsilon}{2} &\leq \|\nabla\lambda(\theta_n)\| - \|\nabla\lambda(\theta_{n'})\| \\
&\leq \|\nabla\lambda(\theta_n) - \nabla\lambda(\theta_{n'})\| \\
&\leq C\|r_n - r_{n'}\| \\
&= C\left\|\sum_{m=n}^{n'-1}\gamma_m h(r_m) + \sum_{m=n}^{n'-1}\varepsilon_m\right\| \\
&\leq C\sum_{m=n}^{n'-1}\gamma_m\|h(r_m)\| + C\left\|\sum_{m=n}^{n'-1}\varepsilon_m\right\|,
\end{aligned}$$

for some constant C, as $\nabla^2\lambda(\theta)$ is bounded by Lemma 4. Recall that the function $\|h(r)\|$ is bounded by some constant B by Lemma 23. Furthermore, when n is large enough, the summability of the sequence ε_m yields $C\|\sum_{m=n}^{n'-1}\varepsilon_m\| \leq \epsilon/4$. This implies that $\sum_{m=n}^{n'-1}\gamma_m \geq \epsilon/4CB$. By an argument similar to the one that led to Eq.(C.4), it is easily shown that there exists some $\beta > 0$ such that

$$\lambda(\theta_{n'}) \geq \lambda(\theta_n) + \beta,$$

which contradicts the convergence of the sequence $\lambda(\theta_m)$ which is ensured by Lemma 32. □

C.2 Proofs of Proposition 24-25 and 29-30

In this section, we analyze the two algorithms proposed in Section 5.4 which are given by the iteration rules

$$\theta_{m+1} = \theta_m + \gamma_m F_{S^*,m}(\theta_m, \tilde{\lambda}_m), \tag{C.5}$$

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

where $F_{S^*,m}(\theta_m, \tilde{\lambda}_m)$ is defined by Eq.(5.18) in Section 5.4.1, and by

$$\theta_{m+1} = \theta_m + \gamma_m F_{\alpha,m}(\theta_m, \tilde{\lambda}_m), \tag{C.7}$$

$$\tilde{\lambda}_{m+1} = \tilde{\lambda}_m + \eta\gamma_m \sum_{n=t_m}^{t_{m+1}-1}(g_{i_n}(\theta_m) - \tilde{\lambda}_m), \tag{C.8}$$

respectively, where $F_{\alpha,m}(\theta_m, \tilde{\lambda}_m)$ is defined by Eq.(5.19) in Section 5.4.2. Here, γ_m is a step size parameter and η is a positive scalar.

First, we consider the iteration given by Eqs.(C.5) and (C.6), which we rewrite as

$$\begin{aligned}
\theta_{m+1} &= \theta_m + \gamma_m\left(E_{\theta_m}[T]\nabla\lambda(\theta_m) + \sigma_{S^*}(\theta_m) + G_{S^*}(\theta_m)(\lambda(\theta_m) - \tilde{\lambda}_m)\right) + \varepsilon_{\theta,m}, \\
\tilde{\lambda}_{m+1} &= \tilde{\lambda}_m + \eta\gamma_m(\lambda(\theta_m) - \tilde{\lambda}_m) + \varepsilon_{\lambda,m},
\end{aligned}$$

where

$$\begin{aligned}
\varepsilon_{\theta,m} &= \gamma_m \left(F_{S^*,m}(\theta_m, \tilde{\lambda}_m) - E_{\theta_m}[T] E_{\theta_m}\left[F_{S^*,m}(\theta_m, \tilde{\lambda}_m)\right]\right), \\
\varepsilon_{\lambda,m} &= \eta\gamma_m \left(\sum_{n=t_m}^{t_{m+1}-1} (g_{i_n}(\theta_m) - \tilde{\lambda}_m) - (\lambda(\theta_m) - \tilde{\lambda}_m)\right),
\end{aligned}$$

and $G_{S^*}(\theta)$ and $\sigma_{S^*}(\theta)$ are defined in Proposition 22 and 23 in Section 5.4.1. Let us briefly comment on these terms In Section 5.4.1, we showed that $G_{S^*}(\theta)$ and $\sigma_{S^*}(\theta)$ are bounded as a function of θ. By adapting the argument given to prove Proposition 24 in Appendix C.1, it can be shown that series $\sum \varepsilon_{\theta,m}$, and $\sum \varepsilon_{\lambda,m}$, converge with probability 1.

Similarly, we can rewrite the iteration given by Eq.(C.7)-(C.8) in the form

$$\begin{aligned}
\theta_{m+1} &= \theta_m + \gamma_m \left(E_{\theta_m}[T]\nabla\lambda(\theta_m) + \sigma_\alpha(\theta_m) + G_\alpha(\theta_m)(\lambda(\theta_m) - \tilde{\lambda}_m)\right) + \varepsilon_{\theta,m}, \\
\tilde{\lambda}_{m+1} &= \tilde{\lambda}_m + \eta\gamma_m(\lambda(\theta_m) - \tilde{\lambda}_m) + \varepsilon_{\lambda,m},
\end{aligned}$$

where the functions $\sigma_\alpha(\theta)$, and $G_\alpha(\theta)$ are bounded, and where the series $\sum \varepsilon_{\theta,m}$, and $\sum \varepsilon_{\lambda,m}$, converge with probability 1.

This leads us to consider the following iteration rule

$$\theta_{m+1} = \theta_m + \gamma_m \left(E_{\theta_m}[T]\nabla\lambda(\theta_m) + \sigma(\theta_m) + \bar{G}(\theta_m)(\lambda(\theta_m) - \tilde{\lambda}_m)\right) + \varepsilon_{\theta,m}, \tag{C.9}$$

$$\tilde{\lambda}_{m+1} = \tilde{\lambda}_m + \eta\gamma_m(\lambda(\theta_m) - \tilde{\lambda}_m) + \varepsilon_{\lambda,m}, \tag{C.10}$$

for which we assume that the functions $\sigma(\theta)$ and $\bar{G}(\theta)$ are bounded, and the terms $\varepsilon_{\theta,m}$ and $\varepsilon_{\lambda,m}$ are such that the series $\sum_m \varepsilon_{\theta,m}$, and $\sum_m \varepsilon_{\theta,m}$, respectively, converge. To prove Propositions 24, 25, 29, and 30, it then suffices to prove the following results.

Proposition 35 *Let Assumptions 5 (MRP Parameterization), 7 (MRP Recurrence) and 8 (Step Size 1) hold, and let $T_{\min}$ and D be such that, for all $\theta \in \Re^K$, we have*

$$\begin{aligned}
&1 \le T_{\min} \le E_\theta[T], \\
&\|\sigma(\theta)\| \le D,
\end{aligned}$$

Furthermore, let (θ_m) be the sequence of parameter vectors generated by the algorithm given by Eqs.(C.9) - (C.10). Then, with probability 1, we have

$$\liminf_{m\to\infty} \|\nabla\lambda(\theta_m)\| \le \frac{D}{T_{min}}.$$

Proposition 36 *Let Assumptions 5 (MRP Parameterization), 7 (MRP Recurrence) and 8 (Step Size 1) hold, and let $T_{\min}$, L, and D be such that, for all $\theta \in \Re^K$, we have*

$$\begin{aligned}
&1 \le T_{\min} \le E_\theta[T], \\
&\|\bar{G}(\theta)\| \le L, \\
&\|\sigma(\theta)\| \le D < \frac{\eta T_{\min}}{4L}.
\end{aligned}$$

Furthermore, let (θ_m) be the sequence of parameter vectors generated by the algorithm given by Eqs.(C.9) - (C.10). Then, with probability 1, we have

$$\limsup_{m\to\infty}(\tilde{\lambda}_m - \lambda(\theta_m)) \le \frac{\eta T_{min}}{2L^2}\left[\left(1 - \frac{2LD}{\eta T_{min}}\right) - \sqrt{1 - \frac{4LD}{\eta T_{min}}}\right].$$

To prove Proposition 35 and 36, we take a similar approach as in Appendix C.1. In particular, we will use a few different Lyapunov functions to analysis the algorithm in different "regions".

As in Appendix C.1, we define the augmented parameter vector $r_m = (\theta_m, \tilde{\lambda}_m)$, and write the update equations in the form

$$r_{m+1} = r_m + \gamma_m(h(r_m) + \rho(r_m)) + \varepsilon_m,$$

where

$$\begin{aligned} h(r) &= = \begin{bmatrix} E_\theta[T]\nabla\lambda(\theta) + \bar{G}(\theta)(\lambda(\theta) - \tilde{\lambda}) \\ \eta E_\theta[T](\lambda(\theta) - \tilde{\lambda}) \end{bmatrix}, \\ \rho(r) &= \begin{bmatrix} \sigma(\theta) \\ 0 \end{bmatrix}, \\ \varepsilon_m &= \begin{bmatrix} \varepsilon_{\theta,m} \\ \varepsilon_{\lambda,m} \end{bmatrix}. \end{aligned}$$

Also, recall how we defined in Appendix C.1 the set $\mathcal{D}_c = \{(\theta, \tilde{\lambda}) \in \Re^{K+1} \mid |\tilde{\lambda}| \le c\}$ and the set Φ, which contains all functions ϕ that are twice differentiable and for which $\nabla\phi$ and $\nabla^2\phi$ are bounded on $\mathcal{D}_c$, for every $c \ge 0$. We can show, by the same argument as the one given to prove Lemma 25, that for every function $\phi \in \Phi$ we have, for

$$\varepsilon_m(\phi) = \phi(r_{m+1}) - \phi(r_m) - \gamma_m\nabla\phi(r_m)\cdot(h(r_m) + \rho(r_m)),$$

that the series $\sum_m \varepsilon_m(\phi)$ converges with probability 1.

Let us first prove Proposition 35. We start out by showing in the next Lemma that if $\|\nabla\lambda(\theta_m)\|$ is nonzero, and the two quantities $\|\sigma(\theta_m)\|$ and $|\lambda(\theta_m) - \tilde{\lambda}_m|$ are small enough, then the difference $(\lambda(\theta_m) - \tilde{\lambda}_m)$ tends to increase.

Lemma 33 *Let L be such that $\|\bar{G}(\theta)\| \le L$, for all θ, let*

$$B(\theta) = \frac{\|\nabla\lambda(\theta)\|\left(\|\nabla\lambda(\theta)\|T_{min} - \|\sigma(\theta)\|\right)}{\eta T_{max} + \|\nabla\lambda(\theta)\|L}$$

and let

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

We have $\phi \in \Phi$. Furthermore, if $\left(\|\nabla\lambda(\theta)\|T_{min} - \|\sigma(\theta)\|\right) > 0$, and $|\tilde{\lambda} - \lambda(\theta)| \le B(\theta)$, then

$$\nabla\phi(r)\cdot\left(h(r) + \rho(r)\right) \ge 0.$$

Proof: The fact that $\phi \in \Phi$ is a consequence of Lemma 4. We now have

$$\begin{aligned}\nabla\phi(r)\cdot(h(r)+\rho(r)) &= -\eta E_\theta[T](\lambda(\theta)-\tilde{\lambda}) + E_\theta[T]\|\nabla\lambda(\theta)\|^2 + \\ &\quad (\lambda(\theta)-\tilde{\lambda})\nabla\lambda(\theta)\cdot\bar{G}(\theta) + \nabla\lambda(\theta)\cdot\sigma(\theta) \\ &\geq -\eta T_{max}|\lambda(\theta)-\tilde{\lambda}| + T_{min}\|\nabla\lambda(\theta)\|^2 - \\ &\quad L|\lambda(\theta)-\tilde{\lambda}|\|\nabla\lambda(\theta)\| - \|\nabla\lambda(\theta)\|\|\sigma(\theta)\| \\ &\geq -|\lambda(\theta)-\tilde{\lambda}|\Big(\eta T_{max} + L\|\nabla\lambda(\theta)\|\Big) + \\ &\quad \|\nabla\lambda(\theta)\|\left(\|\nabla\lambda(\theta)\|T_{min} - \|\sigma\|\right),\end{aligned}$$

which is nonnegative as long as $(\|\nabla\lambda(\theta)\|T_{min} - \|\sigma(\theta)\|) > 0$, and $|\tilde{\lambda}-\lambda(\theta)| \leq B(\theta)$. □

Note that the argument given in the proof of Lemma 29 of Appendix C.1 remains valid here and we have the following result.

Lemma 34 *We have* $\liminf_{m\to\infty}|\lambda(\theta_m)-\tilde{\lambda}_m| = 0$.

We are now ready to prove Proposition 35.

Proof of Proposition 35: If the proposition is not true, then there exists a constant $\epsilon > 0$ and an integer N such that

$$\|\nabla\lambda(\theta_m)\| > \frac{D}{T_{min}} + \epsilon, \qquad \text{for all } m \geq N.$$

Let then B be a bound on $\|\nabla\lambda(\theta)\|$, which exists by Lemma 4. Furthermore, let $D' = D/T_{min} + \epsilon$, and let $\beta = D'\epsilon/(\eta T_{max} + BL)$, where L is a bound on $\|\bar{G}(\theta)\|$ used in the statement of the proposition.

In a first step we show that if the proposition is not true, then the quantity $|\lambda(\theta_m)-\tilde{\lambda}_m|$ is infinitely many times larger or close to β, i.e., if

$$\liminf\|\nabla\lambda(\theta_m)\| > D',$$

then we have

$$\limsup_{m\to\infty}|\lambda(\theta_m)-\tilde{\lambda}_m| \geq \beta.$$

To do that, we use the Lyapunov function

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

Suppose the contrary. Then there exists a constant α, with $0 < \alpha < \beta$, such that, for all m large enough, we have $|\lambda(\theta_m)-\tilde{\lambda}_m| < \alpha$. This implies that for m large, we have

$$\begin{aligned}\phi(r_{m+1}) &= \phi(r_m) + \gamma_m\nabla\phi(r_m)\cdot\Big(h(r_m)+\rho(r_m)\Big) + \varepsilon_m(\phi) \\ &= \phi(r_m) - \eta\gamma_m\Big(\lambda(\theta_m)-\tilde{\lambda}_m\Big)E_{\theta_m}[T] +\end{aligned}$$

$$
\begin{aligned}
& \gamma_m \|\nabla\lambda(\theta_m)\|^2 E_{\theta_m}[T] + \gamma_m\Big(\lambda(\theta_m) - \tilde{\lambda}_m\Big)\nabla\lambda(\theta_m) \cdot \bar{G}(\theta_m) + \\
& \gamma_m \nabla\lambda(\theta_m) \cdot \sigma(\theta_m) + \varepsilon_m(\phi) \\
\geq\ & \phi(r_m) - \eta\gamma_m T_{max}|\lambda(\theta_m) - \tilde{\lambda}_m| + \gamma_m\|\nabla\lambda(\theta_m)\|^2 T_{min} - \\
& \gamma_m L|\lambda(\theta_m) - \tilde{\lambda}_m|\|\nabla\lambda(\theta_m)\| - \gamma_m\|\nabla\lambda(\theta_m)\|\|\sigma(\theta_m)\| + \varepsilon_m(\phi) \\
\geq\ & \phi(r_m) + \gamma_m\Big(- \alpha(\eta T_{max} + LB) + D'\epsilon\Big) + \varepsilon_m(\phi).
\end{aligned}
$$

By our choice, we have $\alpha < \beta \leq \frac{D'\epsilon}{\eta T_{\max} + BL}$ and we obtain that

$$
\begin{aligned}
\phi(r_{m+1}) \ &\geq\ \phi(r_m) + \gamma_m\Big(- \alpha(\eta T_{max} + LB) + D'\epsilon\Big) + \varepsilon_m(\phi) \\
&\geq\ \phi(r_m) + \gamma_m\kappa + \varepsilon_m(\phi),
\end{aligned}
$$

for some constant $\kappa > 0$. As $\lim_{m\to\infty}\varepsilon_m(\phi) = 0$ and $\sum_{m=1}^{\infty}\gamma_m = \infty$, it follows that $\phi(r_m)$ diverges. This is a contradiction to Lemma 34, which states that

$$
\liminf_{m\to\infty} |\lambda(\theta_m) - \tilde{\lambda}_m| = 0.
$$

We have shown that if the proposition does not hold, then we have that

$$
\limsup_{m\to\infty} |\lambda(\theta_m) - \tilde{\lambda}_m| \geq \beta,
$$

where β is defined in the beginning of the proof. Using this result together with Lemma 34, which states that $\liminf_{m\to\infty}|\lambda(\theta_m) - \tilde{\lambda}_m| = 0$, it follows that when Proposition 35 is not true, then there are infinitely many pairs n, n', with $n' > n$, such that

$$
\phi(r_{n'}) - \phi(r_n) = \Big(\lambda(\theta_{n'}) - \tilde{\lambda}_{n'}\Big) - \Big(\lambda(\theta_n) - \tilde{\lambda}_n\Big) < -\frac{1}{3}\beta,
$$

and, for $m = n, \ldots, n'-1$,

$$
\begin{aligned}
\|\nabla\lambda(\theta_m)\| &> D', \\
\Big(\|\nabla\lambda(\theta_m)T_{\min} - \|\sigma(\theta_m)\|\Big) &\geq \epsilon > 0, \\
|\lambda(\theta_m) - \tilde{\lambda}_m| &< \beta \\
&\leq \frac{D'\epsilon}{\eta T_{\max} + BL} \\
&\leq \frac{\|\nabla\lambda(\theta_m)\|\Big(\|\nabla\lambda(\theta_m)\|T_{min} - \|\sigma(\theta_m)\|\Big)}{\eta T_{max} + \|\nabla\lambda(\theta_m)\|L},
\end{aligned}
$$

as $T_{\min} \geq 1$. The constants D', ϵ, and β are as defined in the beginning of the proof. In this case, Lemma 33 applies and shows that $\nabla\phi(r_m) \cdot \Big(h(r_m) + \rho(r_m)\Big) \geq 0$, for $m = n, \ldots, n'-1$. Therefore,

$$
\begin{aligned}
-\frac{1}{3}\beta &> \phi(r_{n'}) - \phi(r_n) \\
&= \sum_{m=n}^{n'-1} \Big(\gamma_m \nabla\phi(r_m) \cdot (h(r_m) + \rho(r_m)) + \varepsilon_m(\phi)\Big) \geq \sum_{m=n}^{n'-1} \varepsilon_m(\phi). \qquad \text{(C.11)}
\end{aligned}
$$

Since the series $\sum_m \varepsilon_m(\phi)$ converges, which implies that $\Big\|\sum_{m=n}^{n'-1} \varepsilon_m(\phi)\Big\|$ becomes arbitrarily small, we have a contradiction to Eq.(C.11). This completes the proof. □

It remains to prove Proposition 36. To do this, we will use the following functions in $x, y \in \Re$,

$$
\begin{aligned}
f(x,y) &= x^2 + x\left(\frac{\eta T_{min}}{L^2} - \frac{2y}{L}\right) + \frac{y^2}{L^2} \\
B_l(y) &= \frac{\eta T_{min}}{2L^2}\left[\left(1 - \frac{2Ly}{\eta T_{min}}\right) - \sqrt{1 - \frac{4Ly}{\eta T_{min}}}\right], \\
B_h(y) &= \frac{\eta T_{min}}{2L^2}\left[\left(1 - \frac{2Ly}{\eta T_{min}}\right) + \sqrt{1 - \frac{4Ly}{\eta T_{min}}}\right].
\end{aligned}
$$

Note that, for $0 \leq D \leq T_{min}/4L$, $B_l(D)$ is the bound on $(\tilde{\lambda}_m - \lambda(\theta_m))$ that we used in Proposition 36. It is straightforward to show that the following relations hold,

$$
\begin{aligned}
&f(x, \eta T_{min}/4L) = (x + \eta T_{min}/4L^2)^2, \\
&0 \leq B_l(y_0) \leq \eta T_{min}/4L^2 \leq B_h(y_0), \qquad \text{for } 0 \leq y_0 \leq \eta T_{min}/4L, \qquad \text{(C.12)} \\
&f(x, y_0) \leq 0, \qquad \text{for } (-x) \in [B_l(y_0), B_h(y_0)], \quad 0 \leq y_0 \leq \eta T_{min}/4L. \qquad \text{(C.13)}
\end{aligned}
$$

In addition, $B_l(y)$ is strictly increasing in y.

We are now ready to prove Proposition 36. We start out by proving the following lemma, which states that if $\|\sigma(\theta)\|$ is small and $\Big(\tilde{\lambda} - \lambda(\theta)\Big)$ lies in a certain interval, then the difference $\Big(\tilde{\lambda} - \lambda(\theta)\Big)$ decreases. Note the difference to Lemma 33. Here we assume that $\|\sigma(\theta)\|$ is bounded by a constant, while in Lemma 33 the bound on $\|\sigma(\theta)\|$ depends on $\|\lambda(\theta)\|$.

Lemma 35 *Let L be such that $\|\bar{G}(\theta)\| \le L$, for all θ, and let*

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

We have $\phi \in \Phi$. Furthermore, if $\|\sigma(\theta)\| \le \frac{\eta T_{min}}{4L}$, and if $B_l(\|\sigma(\theta)\|) \le \tilde{\lambda} - \lambda(\theta) \le B_h(\|\sigma(\theta)\|)$, then

$$\nabla\phi(r) \cdot \Big(h(r) + \rho(r)\Big) \le 0.$$

Proof: The fact that $\phi \in \Phi$ is a consequence of Lemma 4. We now have

$$\begin{aligned}
\nabla\phi(r) \cdot \Big(h(r) + \rho(r)\Big) &= -(\tilde{\lambda} - \lambda(\theta))E_\theta[T] - \|\nabla\lambda(\theta)\|^2 E_\theta[T] + \\
&\quad (\tilde{\lambda} - \lambda(\theta))\nabla\lambda(\theta) \cdot \bar{G}(\theta) + \nabla\lambda(\theta) \cdot \sigma(\theta) \\
&= -(\tilde{\lambda} - \lambda(\theta))E_\theta[T] - \|\nabla\lambda(\theta)\|^2 E_\theta[T] + \\
&\quad \nabla\lambda(\theta) \cdot \Big((\tilde{\lambda} - \lambda(\theta))\bar{G}(\theta) + \sigma(\theta)\Big)
\end{aligned}$$

Using the inequality $|a \cdot b| \le \|a\|^2 + \|b\|^2$ to bound the last term and the fact that $E_\theta[T] \ge 1$, we obtain

$$\begin{aligned}
\nabla\phi(r) \cdot \Big(h(r) + \rho(r)\Big) &\le -\eta(\tilde{\lambda} - \lambda(\theta))T_{min} + \Big[(\tilde{\lambda} - \lambda(\theta))L + \|\sigma(\theta)\|\Big]^2 \\
&\le L^2 \left[(\tilde{\lambda} - \lambda(\theta))^2 - (\tilde{\lambda} - \lambda(\theta))\left(\frac{\eta T_{min}}{L^2} - \frac{2\|\sigma(\theta)\|}{L}\right) + \frac{\|\sigma(\theta)\|^2}{L^2}\right]
\end{aligned}$$

Finally, if $\|\sigma(\theta)\|$ is less or equal to $\eta T_{min}/4L$ and $\tilde{\lambda} - \lambda(\theta)$ is bounded as follows,

$$B_l(\|\sigma(\theta)\|) \le \tilde{\lambda} - \lambda(\theta) \le B_h(\|\sigma(\theta)\|),$$

then, by Eq.(C.13), we have

$$(\tilde{\lambda} - \lambda(\theta))^2 - (\tilde{\lambda} - \lambda(\theta))\left(\frac{\eta T_{min}}{L^2} - \frac{2\|\sigma(\theta)\|}{L}\right) + \frac{\|\sigma(\theta)\|^2}{L^2} \le 0.$$

The lemma follows. □

Proof of Proposition 36: Suppose the contrary. Then there exists a $\epsilon > 0$ such that the inequality

$$\tilde{\lambda}_m - \lambda(\theta_m) > B_l(D) + \epsilon$$

holds for infinitely many m. We introduce some more definitions. Using the assumption that $D < \eta T_{min}/4L$, together with Eq.(C.12) and the fact that $B_l(y)$ is strictly increasing in y, it follows that

$$B_l(D) < \eta T_{min}/4L^2.$$

Let $\delta > 0$ be such that $D + \delta < \eta T_{min}/4L$, and

$$\Big(B_l(D + \delta) - B_l(D)\Big) < \min\Big\{\epsilon, \Big(\eta T_{min}/4L^2\Big) - B_l(D)\Big\},$$

which exists as $B_l(y)$ is continuous in y. Since $B_l(y)$ is strictly increasing in y, it follows that $B_l(D+\delta) > B_l(D)$. Finally, let $\beta = \left(\eta T_{min}/4L^2\right) - B_l(D+\delta)$.

By Lemma 34, we have $\liminf_{m\to\infty} |\lambda(\theta_m) - \tilde{\lambda}_m| = 0$, and we conclude that $\tilde{\lambda}_m - \lambda(\theta_m)$ crosses infinitely often from a value smaller than $B_l(D+\delta) + \beta/3$ to a value larger than $B_l(D+\delta) + 2\beta/3$. In particular, there exist infinitely many pairs n, n', with $n' > n$, such that

$$B_l(D+\delta) < \tilde{\lambda}_n - \lambda(\theta_n) < B_l(D+\delta) + \frac{1}{3}\beta, \qquad \tilde{\lambda}_{n'} - \lambda(\theta_{n'}) > B_l(D+\delta) + \frac{2}{3}\beta,$$

and, for $m = n+1, \ldots, n'-1$,

$$\begin{aligned} &B_l(D+\delta) + \frac{1}{3}\beta \leq \tilde{\lambda}_m - \lambda(\theta_m) \leq B_l(D+\delta) + \frac{2}{3}\beta, \\ &\|\sigma(\theta_m)\| < D+\delta. \end{aligned}$$

We use the Lyapunov function

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

and note that

$$\phi(r_{n'}) \geq \phi(r_n) + \frac{\beta}{3}. \tag{C.14}$$

By our choice, for $m = n, \ldots, n'-1$, we have that $\|\sigma(\theta_m)\| < D+\delta < T_{min}/4L$ and

$$B_l(\|\sigma(\theta_m)\|) < B_l(D+\delta) < \tilde{\lambda} - \lambda(\theta) \leq \eta T_{\min}/4L^2 \leq B_h(\|\sigma(\theta_m)\|).$$

Lemma 35 applies and shows that $\nabla\phi(r_m) \cdot \Big(h(r_m) + \rho(r_m)\Big) \leq 0$. Therefore,

$$\phi(r_{n'}) = \phi(r_n) + \sum_{m=n}^{n'-1} \Big(\gamma_m \nabla\phi(r_m) \cdot \Big(h(r_m) + \rho(r_m)\Big) + \varepsilon_m(\phi)\Big) \leq \phi(r_n) + \sum_{m=n}^{n'-1} \varepsilon_m(\phi).$$

Since the series $\sum_m \varepsilon_m(\phi)$ converges, we have that $\left\|\sum_{m=n}^{n'-1} \varepsilon_m(\phi)\right\|$ becomes arbitrarily small. This contradicts Eq.(C.14) and completes the proof. $\square$