# MULTIGRID ALGORITHMS AND COMPLEXITY RESULTS FOR DISCRETE-TIME STOCHASTIC CONTROL AND RELATED FIXED-POINT PROBLEMS

by

**Chee-Seng Chow**
Electrical Engineer, MIT, 1987
S.M., Electrical Engineering and Computer Science, MIT, 1985
S.B., Electrical Engineering, MIT, 1985
S.B., Computer Science, MIT, 1985
S.B., Mathematics, MIT, 1985
S.B., Physics, MIT, 1985

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF

**DOCTOR OF PHILOSOPHY**
**IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**

at the

**MASSACHUSETTS INSTITUTE OF TECHNOLOGY**
December 1989

Signature of Author ____________________
Departmen~~t of Electrical~~al Engineering and Computer Science
December 5, 1989

Certified by ____________ ____________________
John N. Tsitsiklis
Thesis Supervisor

Accepted by ____________________
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

# MULTIGRID ALGORITHMS AND COMPLEXITY RESULTS FOR DISCRETE-TIME STOCHASTIC CONTROL AND RELATED FIXED-POINT PROBLEMS

by

**Chee-Seng Chow**

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF
DOCTOR OF PHILOSOPHY
IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
DECEMBER 1989

## Abstract

We study the computational aspects of discrete-time stochastic control and related fixed-point problems. We first consider the approximate solution of a discrete-time stochastic control model: stationary, infinite-horizon, discounted-cost Markov Decision Processes that satisfy certain Lipschitz continuity assumptions. We study the computational complexity of the problem as a function of the accuracy and the discount parameters. We also study the effects of ergodicity conditions on the problem's discretization and complexity. We analyze the complexity of a traditional single-grid and a one-way multigrid algorithm for the problem. We show that the multigrid algorithm improves upon the single-grid algorithm. We show the optimality of the multigrid algorithm by obtaining information-based lower bounds on the problem. We also study the computational complexity of average-cost Markov Decision Processes.

We then develop an abstract framework for studying multigrid algorithms for related fixed-point problems. We consider a general fixed-point problem that can be discretized at various levels of accuracy. We assume that at each grid-level, we have an iterative algorithm for computing an approximate solution to the discretized problem. We consider three grid-level dependent parameters: (i) the rate of convergence of the algorithm, (ii) the cost of an iteration of the algorithm, and (iii) the accuracy of the discretized solution. We analyze the complexity of a single-grid and a one-way multigrid algorithm for some special cases that include discrete-time stochastic control, simulated annealing, and boundary value problems. We prove the optimality of the one-way multigrid algorithm by constructing an algorithmic-based lower bound. We also prove the optimality of one-way algorithms in general, under certain monotonicity conditions on the grid-level dependent parameters.

Thesis Supervisor: Dr. John N. Tsitsiklis
Title: Associate Professor of Electrical Engineering

**Acknowledgments**

I thank my thesis supervisor, Professor John Tsitsiklis, for his invaluable guidance and support. I am deeply impressed by his patience, his kindness, and his generousity. I greatly respect him as a teacher, as an advisor, and admire him as a person, as a friend. I consider myself very fortunate to have been supervised by him.

I thank my thesis readers, Professors Dimitri Bertsekas and Pierre Humblet, for their help and encouragement. Professor Bertsekas's pioneering work in dynamic programming has been a major influence on this thesis. Professor Humblet has frequently helped me on problems with the computing facilities.

Special thanks to Professor Michael Athans, my graduate counseler for more than three years, and who has been very helpful and supportive throughout my graduate years at MIT. He always gives good advice.

Numerous professors at MIT have had a major influence on my education and development. I am very glad to have learned and received help from them. I now thank them: Professors Fernando Corbato, Munther Dahleh, Alvin Drake, Robert Gallager, and Gerald Sussman.

Thanks to Dr. Craig Douglas at IBM Research, Yorktown Heights, for introducing me to the term "one-way multigrid algorithm". I have also gained valuable insights on the subject after a discussion with him.

I thank all my colleagues and officemates at LIDS for the many good discussions we have had. I thank the administrative staff members at LIDS who are always very friendly and very eager to help.

I thank my parents and my siblings for their patience and support, my wife for her love and encouragement.

# Contents

# List of Figures

# Chapter 1

# Introduction and Summary

We begin by giving a non-technical perspective of our research objectives. We then give a more detailed description of our work and survey some relevant literature. Next, we give an outline of the report. Finally, we list our contributions.

## 1.1 Introduction and Motivation

In this report, we study the computational aspects of discrete-time stochastic control and related fixed-point problems. This study is motivated by the desire to understand the close connection between control and computation.

The purpose of control is to design controllers to take care of certain control problems so that we do not have to be concerned about them. The use of such automatic or feedback controllers goes far back in man's history [see, for example, Mayr (1970) for a historical account on the development of mechanical feedback controllers]. The intent of control has been exercised probably as far back as the emergence of man as a social animal—with group leaders assigning "control tasks" to subordinates, so that they can pursue other interests while keeping things in order.

However, the advent of computer has changed the situation. Instead of using mechanical controllers with their mechanical failures or human controllers with their human failures, we now can use computers, which are logical, precise, and can carry out commands quickly and faithfully. Just look at any piece of twentieth century human engineering. From cars to space shuttles, to the space stations of the future, there are thousands (and thousands) of controllers in them, each finely tuned for some specific control tasks. The current trend is that logical controllers are quickly replacing mechanical controllers, just as mechanical controllers have replaced human controllers.

This trend points to an increasingly close connection between control and computation as two fields of study. However in this report we will look at only a very narrow aspect of this connection. We begin by considering a well-known, well-studied discrete-time stochastic control model: Markov Decision Processes. We address the question on the amount of computational effort needed to solve the problem to a certain accuracy as a function of the accuracy parameter. This is a natural question to ask for the following reasons: (i) Intuitively it is clear that the more accurately we want to solve the problem, the more computational effort is required. It is certainly nice to have a formal quantitative characterization of this intuition. (ii) A more important reason is that since computers are used to solve control problems, it is natural to want to know how well they perform. Can they be programmed to work more efficiently? Are there any inherent limitations on their performance?

The last two questions lead us into the domain of algorithms and lower bounds, which is the heart of the issue and is also the central theme of this report.

## 1.2 Problem Definition

We now discuss our work in more detail. We first study the approximate solution of a discrete-time stochastic model: stationary, infinite-horizon, discounted-cost Markov Decision Processes. We assume that the problem has bounded state and control spaces and satisfies certain Lipschitz continuity assumptions.

We study the complexity of computing an $\epsilon$-approximation of the optimal cost function. We analyze the complexity as a function of the accuracy parameter $\epsilon$ and the discount parameter $\alpha$ (in the limit as $\epsilon \downarrow 0$ and $\alpha \uparrow 1$). We show that this problem can be viewed as computing an approximation to Bellman's (fixed-point) equation.

We introduce a traditional single-grid and a one-way multigrid successive approximation algorithm. (The one-way multigrid algorithm proceeds from coarse to fine grid.) We analyze the complexity of the algorithms. We show that if the dimensions of the state and the control spaces are $n$ and $m$, respectively, then the complexity of the single-grid algorithm is

$$O\left(\frac{\log\frac{1}{(1-\alpha)\epsilon}}{1-\alpha}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{(2n+m)}\right),$$

and the complexity of the multigrid algorithm is

$$O\left(\frac{1}{1-\alpha}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{(2n+m)}\right).$$

Therefore, multigrid is an improvement of the single-grid algorithm.

We also study the effects of ergodicity conditions on the problem's discretization and complexity. We show that ergodicity improves the accuracy of discretization. Moreover, under an ergodicity condition, the single-grid and the multigrid algorithms have complexity

$$O\left(\log\frac{1}{(1-\alpha)\epsilon}\left[\frac{1}{(1-\alpha)\epsilon}\right]^{(2n+m)}\right) \quad \text{and} \quad O\left(\left[\frac{1}{(1-\alpha)\epsilon}\right]^{(2n+m)}\right),$$

respectively.

We show the optimality of the one-way multigrid algorithm by proving lower bounds. Using an adversary-type argument, we establish information-based lower bounds on the problem. We show that the information-based lower bounds are $\Omega\left([(1-\alpha)^2\epsilon]^{-(2n+m)}\right)$, in general, and $\Omega\left([(1-\alpha)\epsilon]^{-(2n+m)}\right)$, with ergodicity. Therefore, the one-way multigrid algorithm is, in general, within a factor of $O(1/(1-\alpha))$ from optimal; with ergodicity, it is optimal.

We also study the complexity of computing an $\epsilon$-optimal policy. We show that this problem is, in a certain sense, as hard as computing an $\epsilon$-optimal cost function. In particular, the earlier upper and lower bounds apply to this problem as well.

We also discuss other models of discrete-time stochastic control (average-cost and deterministic problems) and a multigrid algorithm based on policy iteration.

In the second part of the report, we develop an abstract framework for studying multigrid algorithms for related fixed-point problem. We consider the approximate solution of a general fixed-point equation

$$Ax = x$$

that can be discretized at various levels of accuracy. We assume that the equation has a unique fixed-point, denoted by $x^*$, and can be discretized into a family of fixed-point equations

$$A_h x = x, \qquad h \in (0,1],$$

where $A_h$ is a contraction mapping [on a metric space $(\mathsf{M},\mathsf{d})$] with fixed point $x_h^*$. (Here, a smaller $h$ implies a more accurate discretization.)

An algorithm, in the framework, computes an $\epsilon$-approximation of $x^*$ using only the family of contraction mappings $\{A_h\}$. We consider the following three grid-level dependent parameters: (i) the contraction factor of $A_h$, (ii) the cost of an iteration of $A_h$, and (iii) the distance $\mathsf{d}(x^*, x_h^*)$. We analyze the complexity of a single-grid and a one-way multigrid algorithm for some special cases that include discrete-time stochastic control, simulated annealing, and boundary value problems. We prove the optimality of the one-way multigrid algorithm by constructing an algorithmic-based lower bound. We also prove the optimality of one-way multigrid algorithms in general, under certain monotonicity conditions on the grid-level dependent parameters.

## 1.3 Literature Survey

The subject matter in this report is related to numerous areas of research. We now give a general survey of some of them; more specific references are given at the end of the relevant sections.

### 1.3.1 Computational Aspects

Our goal to understand the computational aspects of control problems is part of an ongoing research effort [see, for example, Papadimitriou and Tsitsiklis (1986, 1987)]. Papadimitriou and Tsitsiklis (1987) study the complexity of (finite-state) Markov Decision Problems. They show that these problems can be grouped into different complexity classes. The difference with their work is that theirs is inherently discrete, whereas ours is continuous.

Papadimitriou and Tsitsiklis (1986) also study certain decentralized decision making and control problems. They are able to relate the complexity of a continuous problem to that of a discrete version, thereby showing the intractability of certain continuous problems.

The continuous model of computation we use in this report has been studied and used by other researchers [see, for example, Papadimitriou and Tsitsiklis (1986), Nemirovsky and Yudin (1983), and Traub, et al (1988)]. This model of computation is often used in Numerical Analysis when only the number of arithmetic operations is counted, and roundoff error is ignored. For other works using this model of computation, see Traub, et al (1988).

### 1.3.2 Multigrid Methods

In this report we will look at two different algorithms: a traditional single-grid algorithm and a multigrid algorithm. Multigrid algorithms or, more generally, multigrid methods have a long history. They are the outgrowth of certain iterative methods used for solving elliptic partial differential equations and boundary value problems. They have been studied extensively by many researchers [see, for example, Hackbusch (1985) and Brandt (1986)]. Since then, the study of multigrid methods has become a field by itself. Its areas of applications include algebraic problems [Chatelin and Miranker (1982)], statistical physics [Goodman and Sokal (1986)], vision problems [Terzopoulos (1984)], and many others.

A simplified explanation why multiple grid levels should be used rather than using only one grid level is summarized by Brandt's golden rule: *Computation work should be proportional to real physical change in the system.* By using different grid levels, different levels of physical change of a system can be captured more effectively. In many problems for which multigrid ideas apply, the resultant algorithms have been shown empirically (and, in some limited cases, rigorously) to be optimal.

In the context of solving finite-state dynamic programming problems, multigrid ideas have been introduced and studied. For example, Schweitzer, et al (1985) introduce an aggregation-disaggregation method for accelerating the rate of convergence of a successive approximation algorithm. Bertsekas and Castanon (1986) go a step further and introduce an adaptive version, where the set of states are aggregated and disaggregated adaptively during the course of computation.

The multigrid algorithm we study in this report is different from all of the above multigrid (or multigrid-based) algorithms—though ours can be viewed as a special case—it is a *one-way* multigrid algorithm where the grid-levels are always chosen from coarse to fine. All the other multigrid algorithms mentioned above move up and down the grid-levels. An explanation for the difference is that we are interested in proving the *asymptotic optimality* of the algorithm, while other work is interested in improving the rate of convergence of the algorithm. (We will discuss improving the rate of convergence of the algorithm, but in a different context—ergodicity conditions.)

One-way multigrid algorithms are not new. However, our work appears to be the first systematic and rigorous study, particularly in the context of discrete-time stochastic control. One-way multigrid algorithms may be too simplistic for practical use. But they are easy to analyze, and in the context of discrete-time stochastic control, they can be shown to be optimal.

Multigrid algorithms have also been reported in the context of continuous-time stochastic control [see Hoppe (1986) and Akian, et al (1988)]. Again the algorithms move up and down the grid levels. A more detailed comparison with these algorithms is given in Section 3.2.3.

### 1.3.3 Discretization Procedures

The main reason for discretization is that one can approximate a problem with a large (for example, continuous) state space by a smaller (finite) state space problem, assuming that the original problem is sufficiently regular. Starting with the works by Fox (1971, 1973), by Bertsekas (1975), and by Whitt (1978,1979), a series of papers follows [for example, see Hinderer (1976), White (1980, 1982), Langen (1981), Haurie and L'Ecuyer (1986); see also Hernández-Lerma (1989) and the references therein]. The discretization procedures we use are most similar to Whitt (1978). However, we have adapted our procedures for multigrid analysis (whereas the available discretization procedures are mainly designed for single-grid applications). In a certain sense, our discretization is reminiscent of the non-stationary value-iteration approximation scheme used by Hernández-Lerma (1989).

### 1.3.4 General References

For standard results on finite-state dynamic programming, we use Bertsekas (1987); for measurability issues that arise in continuous-state dynamic programming, we use Bertsekas and Shreve (1978). Other references on dynamic programming as applied to Markov Decision Processes are Howard (1960), Puterman (1978), and Hartley, et al (1980).

A reference that has considerably influenced the presentation of Chapter 2 of this report is Hernández-Lerma (1989). The abstract framework for studying dynamic programming by Denardo (1967) has inspired our development of the abstract framework for studying multigrid algorithms in Chapter 5.

### 1.3.5 Notes

A preliminary version of the results in Chapter 3 has been reported in Chow and Tsitsiklis (1988). Section 3.3 is mainly adapted from Chow and Tsitsiklis (1989a); the rest of Chapter 3 is adapted from Chow and Tsitsiklis (1989b).

## 1.4 Report Outline

This report is divided into two parts that can be read independently. The first part, Chapters 2–4, studies the computational aspects of discrete-time stochastic control. The second part, Chapter 5, discusses an abstract framework for studying multigrid algorithms.

A more detailed outline is as follows. In Chapter 2 we introduce our discrete-time stochastic control model: stationary, infinite-horizon, discounted-cost Markov Decision Processes. We begin with a review of notation and some basic concepts. After giving a precise mathematical formulation of the model, we study the effects of certain ergodicity (or mixing) conditions. Next we discuss the discretization of the model. We derive discretization error bounds and show that certain ergodicity conditions improve the discretization bounds. Finally, we specialize the results to the case where the dynamics of the Markov process are described by a density. (This is the case we study in Chapters 3 and 4.)

In Chapter 3, which contains the main contribution of the first part of this report, we prove complexity results. We begin with an introduction to the model of computation, and we define various problem classes and their complexity. We introduce two versions of the successive approximation algorithm for the problem—a single-grid and a one-way multigrid algorithm—and analyze their computational complexity. We show that the (one-way) multigrid algorithm has a better complexity than its single-grid version. We also study the effects of certain ergodicity (or mixing) conditions on the complexity. Next, using an adversary-type argument, we establish the information-based lower bounds for these problems. As a result, we show that the discretization error bounds of Chapter 2 are optimal. We also show that the multigrid successive approximation algorithm, in general, has nearly optimal complexity; the complexity is optimal under certain ergodicity conditions. Finally, we analyze the problem of computing good control policies; we show that this problem is in a certain sense as hard as approximating the optimal cost function. We conclude the chapter with a discussion of some extensions.

Chapter 4 contains exploratory results and some candidate problems for future research. We begin by introducing another discrete-time stochastic control model; we show that this model is computationally harder than Markov Decision Processes; in particular, the discretization error bounds of Chapter 2 do not apply. We then discuss multigrid algorithms that are based on policy iteration (instead of successive approximation). We discuss some unresolved issues regarding analyzing the complexity of these algorithms. Next, we discuss average-cost problems. We show that these problems are, in general, computationally ill-posed. We also show that, under certain ergodicity conditions, the problems become well-posed and the algorithms of Chapter 3 can be used. Finally, we discuss some simple numerical results showing that the one-way multigrid algorithm performs better than its single-grid counterpart.

In Chapter 5, which is the second part of this report, we introduce a general framework for studying multigrid algorithms. This framework models the problem of computing an approximate solution of a general fixed-point problem that can be discretized at various levels of discretizations. We assume that at each grid (or discretization) level, we have an iterative algorithm for computing the solution of the discretized problem; furthermore, the algorithms have different convergence rates and iteration costs, and the discretized solutions have different discretization error.

In Section 5.1, we introduce a minimum computational cost problem to model the tradeoffs between these three parameters: rate of convergence, iteration cost, and discretization error.

In Section 5.2, we consider some special cases of the general framework and analyze the complexity of a single-grid algorithm (which uses only one grid-level) and a one-way multigrid algorithm (which uses multiple grid-levels, from coarse to fine). We show that the one-way multigrid algorithm has better complexity than the single-grid algorithm. We also prove a lower bound showing that the one-way multigrid algorithm is, in a certain sense, optimal.

In Section 5.3, we consider three areas of applications: the discrete-time stochastic control problem of the first part of this report, simulated annealing, and boundary value problems. Using the results of Section 5.2, we are able to analyze these problems with minimal effort. We also discuss how the Full Multigrid V-cycle algorithm [see, for example, Böhmer and Stetter (1984), Hackbusch (1985), or Briggs (1987)], which is widely used for the numerical solution of partial differential equations, can be viewed and analyzed as a one-way algorithm.

In Section 5.4, we discuss some more general results. We show, under general conditions, that one-way multigrid algorithms are in a certain sense optimal.

Finally, in Section 5.5 we discuss some alternative aspects of our general framework and suggest problems for future research.

In Chapter 6, we state our conclusions and suggest general areas for future research.

## 1.5 Contributions of This Report

We now state our contributions.

In Chapter 2, we study the effects of ergodicity conditions on the discretization of discounted-cost Markov Decision Processes. Prior research on ergodicity conditions has been mainly in the finite-state setting and in the context of average-cost problems. We show that certain ergodicity conditions improve the discretization of discounted-cost problems. We introduce a "novel" discretization procedure which, in our opinion, is more suitable for analyzing multigrid algorithms for discrete-time stochastic control.

In Chapter 3, we characterize the computational complexity of discounted-cost Markov Decision Processes. We show that computing an $\epsilon$-optimal policy is, in a certain sense, as hard as computing an $\epsilon$-optimal cost function. We show, in a detailed and rigorous analysis, that multigrid has better complexity than single-grid successive approximation. We establish information-based lower bounds on the problem. The lower bounds show the optimality of multigrid successive approximation and the optimality of the discretization procedures of Chapter 2. A novel feature of our complexity analysis is that we simultaneously consider the dependence on both the accuracy and the discount parameters

In Chapter 4, we show that deterministic problems, in general, do not have linear discretization error bounds. This implies that a certain popular discrete-time stochastic control model is computationally harder than Markov Decision Processes. We also show that, in general, average-cost problems are computationally ill-posed, and that under an ergodicity condition the problem is well-posed. We characterize the complexity of average-cost problems under the ergodicity condition and show the optimality of multigrid successive approximation.

The main contributions of Chapter 5 are as follows: (i) We introduce a novel framework for studying multigrid algorithms for general fixed-point problems. (ii) We analyze the complexity of a single-grid algorithm and a one-way multigrid for some special cases that include discrete-time stochastic control, simulated annealing, and boundary value problems. (iii) We prove an new type of lower bounds—algorithmic-based lower bounds; thereby, proving the optimality of the one-way multigrid algorithm within the framework. (iv) We also establish the optimality of one-way multigrid algorithms in general.

The framework gives a unified view of many fixed-point problems: discrete-time stochastic control, simulated annealing, and boundary value problems. Moreover, it can model multigrid algorithms used in practice. More interestingly, the framework illustrates the use of the optimality principle of dynamic programming for solving minimum computational cost problems, further illustrating the close connection between control and computation.

# Chapter 2

# Markov Decision Processes and Their Discretizations

In a typical Markov Decision Process (MDP), we are given a controlled discrete-time system that evolves in a state space $S$ and we are interested in computing a fixed point $J^*$ of the dynamic programming operator $T$ (acting on a space of functions on the set $S$) defined by

$$(TJ)(x) = \inf_{u \in C} \left[ g(x,u) + \alpha \int_S q(dy \mid x,u) J(y) \right], \qquad \forall x \in S.$$

Here, $C$ is the control space, $g(x,u)$ is the cost incurred if the current state is $x$ and control $u$ is applied, $\alpha \in (0,1)$ is a discount factor, and $q(\cdot \mid x,u)$ is a stochastic kernel that specifies the probability measure of the next state, when the current state is $x$ and control $u$ is applied. Then, $J^*(x)$ is interpreted as the value of the expected discounted cost, starting from state $x$, and provided that the control actions are chosen optimally. Unfortunately, even if the problem data (the functions $g$ and $q$) are given in closed form, the equation $TJ^* = J^*$ does not usually admit a closed form solution and must be solved numerically. This can be accomplished by discretizing the continuous problem to obtain an MDP with finite state and control spaces. Then, the resultant discrete problem can be solved by means of several algorithms (for example, successive approximation, policy iteration, and linear programming). Furthermore, there are bounds available on how fine the discretization should be in order to achieve a desired accuracy.

In this chapter we give a precise formulation of an MDP. This MDP is the main model we will study in this and the next two chapters. We introduce the discretization procedures for the model and estimate the error resulting from discretization. We also study the effect of certain ergodicity or mixing conditions on the convergence of successive approximation and the accuracy of discretization.

We are mainly interested in the case where the dynamics are described by a probability density. To simplify the presentation, it is easier if we proceed from general to specific, by first formulating the model in terms of stochastic kernels. We describe the discretization procedures and the ergodicity conditions, and also study the effects of the ergodicity conditions in terms of stochastic kernels. Only in the end we specialize the results to probability densities.

This chapter is organized as follows. In Section 2.1, we introduce the notation and review some basic concepts. In Section 2.2, we give a precise formulation of an MDP; we introduce Bellman's equation and the optimal cost function and define the objective of computation. In Section 2.3, we study some ergodicity conditions and discuss their effects on the convergence of the successive approximation algorithm. In Section 2.4, we introduce some discretization procedures for the MDP. We also establish bounds on the discretization error and show that certain ergodicity conditions improve the discretization accuracy. In Section 2.5, we specialize the results to the case where the dynamics are described by a density (instead of a kernel), by providing bounds on the discretization error. Finally, in Section 2.6, we summarize the main ideas in the chapter and provide a list of assumptions and notation which will be used in the next chapter.

## 2.1 Notation and Basic Concepts

The purpose of this section is to introduce the notation and review some basic concepts. First, we introduce the basic notation. Second, we look at some examples of normed vector spaces and introduce more notation. Third, we consider operators on a Banach space and review a fixed point theorem. Fourth, we review some basic facts about signed measures. Fifth, we show how kernels can be viewed as operators. Finally, we give some references.

### 2.1.1 Basic Notation

The set of real numbers is denoted by $\mathbf{R}$. We assume that $\mathbf{R}$ is endowed with the usual topology. Let $x, y \in \mathbf{R}$. The minimum (respectively, maximum) of $x$ and $y$ is denoted by $x \wedge y$ (respectively, $x \vee y$). We use $\lceil x \rceil$ to denote the smallest integer greater than or equal to $x$; we use $\lfloor x \rfloor$ to denote the largest integer less than or equal to $x$.

For any positive integer $n$, we view $\mathbf{R}^n$ as a normed vector space by endowing it with the sup-norm $\|\cdot\|_\infty$. Moreover, if $x \in \mathbf{R}^n$ and $u \in \mathbf{R}^m$, for some positive integer $m$, then $(x,u) \in \mathbf{R}^{n+m}$ and $\|(x,u)\|_\infty = \|x\|_\infty \vee \|u\|_\infty$. We also endow $\mathbf{R}^n$ with the usual topology.

Let $\{f_\alpha\}_{\alpha \in I}$ be a collection of real-valued functions defined on a set $X$. Assuming that for all $x \in S$, $\sup_{\alpha \in I} |f_\alpha(x)|$ is finite, we use $\inf_{\alpha \in I} f_\alpha$ to denote the real-valued function defined by

$$\inf_{\alpha \in I} f_\alpha(x) = \inf_{\alpha \in I}\bigl\{f_\alpha(x)\bigr\}, \quad x \in X. \tag{2.1.1}$$

The notation $\sup_{\alpha \in I} f_\alpha$ is defined similarly.

For any topological space $X$, the *Borel sigma-algebra* of $X$ is denoted by $\mathcal{B}_X$; it is the smallest sigma-algebra which contains all of the open sets in $X$. A subset of $X$ is *Borel measurable* if it belongs to $\mathcal{B}_X$. Moreover, a mapping $\psi : X \mapsto Y$, for some topological space $Y$, is *Borel measurable* if $\psi^{-1}(B) \in \mathcal{B}_X$ for all $B \in \mathcal{B}_Y$; in particular, a real-valued function $f : X \mapsto \mathbf{R}$ is *Borel measurable* if $f^{-1}(B) \in \mathcal{B}_X$ for all Borel measurable $B \subset \mathbf{R}$. The space of all bounded Borel measurable (respectively, bounded continuous) functions on $X$ is denoted by $\mathcal{B}(X)$ [respectively, $\mathcal{C}(X)$]. Since we only consider Borel measurability, we will use the word "measurable" to mean "Borel measurable".

### 2.1.2 Normed Vector Spaces

Let $S$ be a measurable subset of $\mathbf{R}^n$. When comparing two functions $J, J' \in \mathcal{B}(S)$, we use the shorthand $J \leq J'$ to denote $J(x) \leq J'(x)$ for all $x \in S$; a similar convention applies to $J = J'$ and $J \geq J'$. Furthermore, any real scalar $c$ may also denote the constant function on $S$ of value c; in particular, $J + c$ denotes the function with value $J(x) + c$ at $x \in S$.

The function space $\mathcal{B}(S)$ becomes a normed vector space when endowed with a norm. One possible norm is the *sup-norm* $\|\cdot\|_\infty$, which is defined by

$$\|J\|_\infty = \sup_{x \in S} |J(x)|, \quad J \in \mathcal{B}(S). \tag{2.1.2}$$

This normed vector space is denoted by $(\mathcal{B}(S), \|\cdot\|_\infty)$.

We also consider the *span-norm* $\|\cdot\|_s$, which is defined by

$$\|J\|_s = \sup_{x \in S} J(x) - \inf_{x \in S} J(x), \quad J \in \mathcal{B}(S). \tag{2.1.3}$$

The span-norm is actually a *semi-norm;* namely, it satisfies all the requirements of being a norm, except that $\|J\|_s = 0$ does not imply $J = 0$. (In particular, it satisfies the triangle inequality.) The span-norm also satisfies

$$\|J\|_s = 2 \min_{c \in \mathbf{R}} \|J + c\|_\infty, \quad J \in \mathcal{B}(S), \tag{2.1.4}$$

where the minimum is attained by letting $c = -\left[\sup_x J(x) + \inf_x J(x)\right]/2$.

We make the span-norm a norm by introducing an equivalence relation between $J, J' \in \mathcal{B}(S)$ whenever $\|J - J'\|_s = 0$. We then partition $\mathcal{B}(S)$ into a collection of equivalence classes. Next, a representative is chosen from each equivalence class. Finally, we let $(\mathcal{B}(S), \|\cdot\|_s)$ be the normed vector space of all the representatives, and let $(\mathcal{C}(S), \|\cdot\|_s)$ be the subspace of all the continuous representatives.

### 2.1.3 Contraction Operators on a Banach Space

A complete normed vector space is called a *Banach space.* Let $(X, \|\cdot\|)$ be a Banach space and let $T$ be an operator mapping $X$ into itself. For any positive integer $k$, let $T^k$ denote the composition of $k$ copies of $T$; let $T^0$ denote the identity operator. The operator $T$ is called a *$k$-stage contraction operator* on $(X, \|\cdot\|)$ with *contraction factor $\alpha$,* if $\alpha \in [0, 1)$ and for all $x, x' \in X$ there hold: (i) $\|Tx - Tx'\| \leq \|x - x'\|$

and (ii) $\|T^k x - T^k x'\| \leq \alpha \|x - x'\|$. A 1-stage contraction operator is usually called a contraction operator.

The main reason for considering contraction operators on a Banach space is the following well-known proposition.

**Proposition 2.1.1** *(Banach's fixed point theorem) If $T$ is a $k$-stage contraction operator on a Banach space $(X, \|\cdot\|)$, then $T$ has a unique fixed point in $X$; namely, there exists a unique $x^* \in X$ such that $Tx^* = x^*$.*

**Proof** See Kolmogorov and Fomin (1970) or Denardo (1967). □

Most normed vector spaces we deal with are Banach spaces. It is well known that $(\mathcal{B}(S), \|\cdot\|_\infty)$ is a Banach space [for example, see Ash (1972)]. Similarly, $(\mathcal{C}(S), \|\cdot\|_\infty)$ is also a Banach space [for example, see Kolmogorov and Fomin (1970)]; hence, it is a closed subspace of $(\mathcal{B}(S), \|\cdot\|_\infty)$. It is easily shown [using Eq. (2.1.4)] that $(\mathcal{B}(S), \|\cdot\|_s)$ and $(\mathcal{C}(S), \|\cdot\|_s)$ also are Banach spaces [for example, see Theorem III.4.2, p. 73, of Conway (1985)].

Lastly, an operator $T : \mathcal{B}(S) \mapsto \mathcal{B}(S)$ is called a *monotone operator*, if for all $J, J' \in \mathcal{B}(S)$, $J \leq J'$ implies that $TJ \leq TJ'$. If $T$ is also a contraction operator, then it is called a *monotone contraction operator.*

### 2.1.4 Signed Measures

Let $S$ be a measurable subset of $\mathbf{R}^n$. Together with its Borel sigma-algebra $\mathcal{B}_S$, $S$ forms a *measurable space* $(S, \mathcal{B}_S)$. A *finite signed measure* on $(S, \mathcal{B}_S)$ is a real-valued function $\nu : \mathcal{B}_S \mapsto \mathbf{R}$ that satisfies the following conditions:

1. $\nu(\phi) = 0$;

2. for any sequence $\{A_i\}_{i=1}^{\infty}$ of disjoint measurable subsets of $S$, there hold

$$\nu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \nu(A_i) \quad \text{and} \quad \sum_{i=1}^{\infty} |\nu(A_i)| < \infty. \tag{2.1.5}$$

The finite signed measure is called a *finite measure* if $\nu(A) \geq 0$ for all $A \in \mathcal{B}_S$. A finite measure is called a *probability measure* (respectively, a *subprobability measure*) if $\nu(S) = 1$ [respectively, $\nu(S) \leq 1$]. Since we only consider finite signed measures and finite measures, we will not use the word "finite" but will simply call them "signed measures" and "measures" instead. (Signed measures are used when we consider generalizations of Markov Decisions Processes.)

## The Decomposition of a Signed Measure

Let $\nu$ be a signed measure on $(S, \mathcal{B}_S)$. We can "decompose" $\nu$ into its positive and negative parts as follows:

$$\nu^+(A) \stackrel{\text{def}}{=} \sup_{B \in \mathcal{B}_S} \nu(A \cap B); \tag{2.1.6}$$

$$\nu^-(A) \stackrel{\text{def}}{=} - \inf_{B \in \mathcal{B}_S} \nu(A \cap B). \tag{2.1.7}$$

Note that $\nu^+$ and $\nu^-$ are measures on $(S, \mathcal{B}_S)$. We can also define another measure on $(S, \mathcal{B}_S)$ by letting

$$|\nu|(A) \stackrel{\text{def}}{=} \nu^+(A) + \nu^-(A), \quad A \in \mathcal{B}_S. \tag{2.1.8}$$

A useful result on the decomposition of a signed measure is the following proposition.

**Proposition 2.1.2** *(Jordan-Hahn decomposition theorem) Let $\nu$ be a signed measure on a measurable space $(S, \mathcal{B}_S)$. There exist two disjoint measurable sets $S^+$ and $S^-$ whose union is $S$ such that*

$$\nu^+(A) = \nu(A \cap S^+) \quad \text{and} \quad \nu^-(A) = -\nu(A \cap S^-), \qquad \forall A \in \mathcal{B}(S). \tag{2.1.9}$$

*In particular, $|\nu|(S) = \nu(S^+) - \nu(S^-)$.*

**Proof** See Royden (1968) or Ash (1977). □

## A Normed Vector Space

Let $\nu_1$ and $\nu_2$ be signed measures on $(S, \mathcal{B}_S)$. We use $\nu_1 \wedge \nu_2$ to denote the signed measure on $(S, \mathcal{B}_S)$ defined by

$$(\nu_1 \wedge \nu_2)(A) = \nu_1(A) \wedge \nu_2(A), \quad \forall A \in \mathcal{B}_S. \tag{2.1.10}$$

We also use the shorthand $\nu_1 \geq \nu_2$ to denote $\nu_1(A) \geq \nu_2(A)$ for all $A \in \mathcal{B}_S$. The notation $\nu_1 \leq \nu_2$ and $\nu_1 = \nu_2$ are similarly defined.

For any scalars $c_1, c_2 \in \mathbf{R}$, we define a signed measure $c_1\nu_1 + c_2\nu_2$ by letting

$$(c_1\nu_1 + c_2\nu_2)(A) = c_1\nu_1(A) + c_2\nu_2(A), \quad \forall A \in \mathcal{B}_S. \tag{2.1.11}$$

It follows that the space of all signed measures on $(S, \mathcal{B}_S)$ forms a vector space (with 0 being the signed measure with value identically zero).

The (total) *variation norm* on a signed measure $\nu$ is given by

$$\|\nu\|_{\mathrm{v}} \stackrel{\text{def}}{=} |\nu|(S).$$

It is easily checked [for example, see Royden (1968)] that $\|\cdot\|_{\mathrm{v}}$ is indeed a norm on the vector space of signed measures; moreover, the resultant normed vector space is complete. Furthermore, the space of all probability measures and the space of all subprobability measures are also Banach spaces with respect to $\|\cdot\|_{\mathrm{v}}$.

We state a useful lemma about $\|\cdot\|_{\mathrm{v}}$.

**Lemma 2.1.1** *Let $\nu_1$ and $\nu_2$ be signed measures on a measurable space $(S, \mathcal{B}_S)$. Then*

$$\nu_1 \wedge \nu_2 = \frac{1}{2}(\nu_1 + \nu_2 - |\nu_1 - \nu_2|);$$

*in particular,* $\|\nu_1 - \nu_2\|_{\mathrm{v}} = \nu_1(S) + \nu_2(S) - 2(\nu_1 \wedge \nu_2)(S)$.

**Proof** See Royden (1968). $\square$

**Integrals and Densities**

Let $f$ be a measurable function on $\mathcal{B}(S)$. The integral with respect to the signed measure $\nu$,

$$\int_S f(x)\nu(dx) \overset{\text{def}}{=} \int_S f(x)\nu^+(dx) - \int_S f(x)\nu^-(dx).$$

It follows that

$$\|\nu\|_{\mathrm{v}} = \max_{f \in \mathcal{B}(S)} \left\{ \int_S f(x)\nu(dx) \mid \|f\|_\infty \leq 1 \right\}, \qquad (2.1.12)$$

where the maximum is attained by letting $f = 1$ on $S^+$ and $f = -1$ on $S^-$. [The sets $S^+$ and $S^-$ are the decomposition of $S$ according to the Jordan-Hahn decomposition theorem (Proposition 2.1.2).] It follows from Eq. (2.1.12) that

$$\left| \int_S f(x)\nu(dx) \right| \leq \|f\|_\infty \|\nu\|_{\mathrm{v}}, \quad \forall f \in \mathcal{B}(S). \qquad (2.1.13)$$

Let $g_1 \in \mathcal{B}(S)$. We define a signed measure $\nu_1$ on $(S, \mathcal{B}(S))$ by letting

$$\nu_1(A) = \int_A g_1(x)\,dx, \quad \forall A \in \mathcal{B}_S, \qquad (2.1.14)$$

where $\int_A \cdot\, dx$ denotes the Borel integral. [It is easily shown that $\nu_1$ is indeed a signed measure.] We say that the signed measure $\nu_1$ is specified by the density $g_1$, or simply $\nu_1$ is a density. Furthermore, if

$$\nu_2(A) = \int_A g_2(x)\,dx, \quad \forall A \in \mathcal{B}_S, \qquad (2.1.15)$$

for some $g_2 \in \mathcal{B}(S)$, then we can use the Jordan-Hahn decomposition theorem to show that

$$\|\nu_1 - \nu_2\|_{\mathrm{v}} = \int_S |g_1(x) - g_2(x)|\,dx. \qquad (2.1.16)$$

## 2.1.5 Kernels

Let $S_1$ and $S$ be measurable subsets of $\mathbf{R}^m$ and $\mathbf{R}^n$, respectively. A *kernel* $q$ on $S_1$ given $S$ is a family of signed measures $q(\cdot \mid x)$ [on $(S_1, \mathcal{B}_{S_1})$] parametrized by $x \in S$. The kernel is *measurable* (respectively, *continuous*), if for all $J \in \mathcal{B}(S)$ [respectively, $J \in \mathcal{C}(S)$] the function $\int_{S_1} J(y)q(dy \mid x)$ is measurable (respectively, continuous) in

$x$. Since we consider only measurable kernels, we will use the word "kernel" to mean "measurable kernel".

Let $q_1$ be a kernel on $S$ given $S$. We can view $q_1$ as a linear operator mapping $\mathcal{B}(S)$ into itself by letting

$$(q_1J)(x) = \int_S q_1(dy \mid x)J(y), \qquad J \in \mathcal{B}(S).$$

Let $q_2$ be another kernel on $S$ given $S$. The composition $q_1q_2$ is a kernel on $S$ given $S$ defined by

$$q_1q_2(A \mid x) = \int_S q_1(dy \mid x)q_2(A \mid y), \quad \forall A \in \mathcal{B}_S.$$

The kernel $q_1q_2$ is also a linear operator on $\mathcal{B}(S)$. Moreover, for any $J \in \mathcal{B}(S)$, we have

$$((q_1q_2)J)(x) = \int_S q_1q_2(dy \mid x)J(y), \quad x \in S.$$

Using Fubini's theorem, we have

$$(q_1q_2)J = (q_1(q_2J));$$

therefore, we can, without ambiguity, omit the parentheses when writing $q_1q_2J$. For any integer $t \geq 0$, we let $q_1^t$ denote the composition of $t$ copies of $q_1$, where $q_1^0$ denotes the identity operator.

Similarly to signed measures, kernels on $S$ given $S$ form a vector space. In particular, for any scalar $c_1, c_2 \in \mathbf{R}$, the kernel $c_1q_1 + c_2q_2$ on $S$ given $S$ is defined by

$$(c_1q_1 + c_2q_2)(A \mid x) = c_1q_1(A \mid x) + c_2q_2(A \mid x), \quad A \in \mathcal{B}_S, x \in S.$$

To obtain a normed vector space, we introduce the sup-norm for kernels,

$$\|q_1\|_\infty \stackrel{\text{def}}{=} \sup_{J \in \mathcal{B}(S)} \{\|q_1J\|_\infty \mid \|J\|_\infty \leq 1\}. \tag{2.1.17}$$

[The sup-norm is an induced norm.] We see that

$$\|q_1J\|_\infty \leq \|q_1\|_\infty\|J\|_\infty, \quad \forall J \in \mathcal{B}(S); \tag{2.1.18}$$

therefore,

$$\|q_1q_2\|_\infty \leq \|q_1\|_\infty\|q_2\|_\infty. \tag{2.1.19}$$

It is easily shown that

$$\|q_1\|_\infty = \sup_{x \in S} \|q(\cdot \mid x)\|_v. \tag{2.1.20}$$

Furthermore, the space of kernels with respect to $\|\cdot\|_\infty$ is a Banach space.

Finally, a kernel is *stochastic* (respectively, *substochastic*) if $q(\cdot \mid x)$ is a probability (respectively, a subprobability) measure on $S$, for all $x \in S$. For stochastic (respectively, substochastic) kernels $q_1$ and $q_2$, it is easily checked that $\|q_1\|_\infty = 1$ (respectively, $\|q_1\|_\infty \leq 1$) and $q_1q_2$ is also a stochastic (respectively, substochastic) kernel. The space of all stochastic kernels and the space of all substochastic kernels are Banach spaces, with respect to the induced sup-norm.

### 2.1.6 Notes

For a more thorough treatment of normed vector spaces see Kolmogorov and Fomin (1970), which also discusses Banach spaces and contraction operators. A standard reference on monotone and contraction operators as applied to dynamic programming is Denardo (1967).

The most useful reference to our discussion on signed measures is Section 11.5 of Royden (1968). In fact our discussion on signed measures (Section 2.1.4) is mainly an adaptation of that section. Another relevant reference on signed measures is Ash (1972).

Our statement of the Jordan-Hahn decomposition theorem (Proposition 2.1.2) is adapted from Hernández-Lerma (1989). This book's appendices contain most of the material in this section. Moreover, this book has significantly influenced the presentation of this chapter.

For an in-depth discussion of stochastic kernels, see Section 7.4.3 of Bertsekas and Shreve (1978). Our definitions of measurable and continuous kernels can be shown to be equivalent to the (standard) definitions in that book. The main reason we view kernels as operators is that this simplifies some of the proofs later on.

## 2.2 Specification of a Markov Decision Process

In this section we give a precise definition of Markov Decision Processes (MDP's), introduce Bellman's equation, and define the objective of computation. Measurability issues pertaining to the formulation of the model are also discussed. There are no new results in this section, only the problem formulation is new.

First, we introduce the model, state the assumptions, and define the optimal cost function. Second, we introduce the dynamic programming operator and show that the optimal cost function is the unique solution to Bellman's equation. We also introduce a measurable selection theorem. Third, we give a precise statement of our objective. Finally, we make some observations and discuss some references.

### 2.2.1 The Model

An MDP has a state space $S$ on which a controlled stochastic process evolves, a control space $C$ from which control actions will be chosen, a stochastic (transition) kernel $q$ on $S$ given $S \times C$ that describes the dynamics of the process, and a cost function $g : S \times C \mapsto \mathbf{R}$. The stochastic kernel $q(\cdot \mid x, u)$ is to be interpreted as the probability measure of the next state, when the current state is $x$ and the control $u$ is applied.

We incorporate state-dependent constraints in our formulation. For each $x \in S$, we are given a nonempty set $U(x) \subset C$ of admissible controls. We are primarily interested in *discounted-cost* MDP's, where a discount factor $\alpha \in (0, 1)$ is also given. For discounted-cost MDP's, future costs are discounted; in particular, if at some stage $t$, the state is $x$ and control $u$ is applied, then the incurred cost is $\alpha^t g(x,u)$. The problem is to control the process so as to minimize the long-term accumulated cost.

## Assumptions

We assume that $S$ and $C$ are bounded, measurable subsets of some Euclidean space; without loss of generality, we can make the further assumption that $S \subset [0,1]^n$ and $C = [0,1]^m$. Let

$$\Gamma \overset{\text{def}}{=} \Big\{ (x,u) \;\Big|\; x \in S \text{ and } u \in U(x) \Big\}. \tag{2.2.1}$$

We assume that $\Gamma$ is the intersection of a closed subset of $\mathbf{R}^n \times \mathbf{R}^m$ with the set $S \times C$; that is, $\Gamma$ is closed with respect to the induced topology on $S \times C$. It follows that $U(x)$ is a compact subset of $C$, for all $x \in S$.

Furthermore, we assume that there exists a constant $K \geq 1$ such that:

**A.1** For all $x, x' \in S$ and $u, u' \in C$, $|g(x,u)| \leq K$ and $|g(x,u) - g(x',u')| \leq K\|(x,u) - (x',u')\|_\infty$.

**A.2** For all $x, x' \in S$ and $u, u' \in C$, $\|q(\cdot \mid x,u) - q(\cdot \mid x',u')\|_{\mathrm{v}} \leq K\|(x,u) - (x',u')\|_\infty$.

**A.3** For any $x, x' \in S$ and any $u' \in U(x')$, there exists some $u \in U(x)$ such that $\|u - u'\|_\infty \leq K\|x - x'\|_\infty$.

**A.4** For all $x \in S$ and $u \in C$, $0 \leq q(\cdot \mid x,u)$ and $q(S \mid x,u) = 1$.

The first assumption states that $g$ is a bounded Lipschitz continuous function. The second assumption states that $q$ is Lipschitz continuous with respect to the variation norm—it ensures that $q$ is a continuous kernel [cf. Eq. (2.2.5)]. The third is a continuity condition on the point-to-set mapping $x \mapsto U(x)$. Finally, Assumption A.4 reflects the fact that $q$ is a stochastic kernel.

We also introduce the following generalizations of Assumption A.4.

**A.4′** For all $x \in S$ and $u \in C$, $0 \leq q(\cdot \mid x,u)$ and $q(S \mid x,u) \leq 1$.

**A.4″** For all $x \in S$ and $u \in C$, $\|q(\cdot \mid x,u)\|_{\mathrm{v}} \leq 1$.

Assumption A.4′ says that $q$ is a substochastic kernel; it models those MDP's in which the system has some nonzero probability of entering a zero-cost absorbing state. Assumption A.4″ says that $q$ is a kernel with variation norm bounded by 1; it models an even more general class of problems that do not necessarily correspond to MDP's.

Unless otherwise stated, Assumptions A.1–A.3 will always be in effect. And by default Assumption A.4 will also be in effect, unless A.4′ or A.4″ is in effect. From now on, $K$ will always denote the constant of these assumptions.

**The Optimal Cost Function**

Let

$$\Pi \stackrel{\text{def}}{=} \left\{ \mu : S \mapsto C \;\middle|\; \mu \text{ is measurable and } \mu(x) \in U(x), \forall x \in S \right\}. \qquad (2.2.2)$$

Let $\Pi^\infty$ be the set of all sequences $\pi = (\mu_0, \mu_1, \ldots)$ of elements of $\Pi$. Each element of $\Pi^\infty$ is called a *policy* and is interpreted as a prescription for choosing control actions as a function of time and of the current state. In particular, if the state at time $t$ is equal to some $x$ and policy $\pi$ is used, then control $\mu_t(x)$ is applied. Once a particular policy is fixed, we can construct a Markov process $\{ x_t^\pi \mid t = 0, 1, \ldots \}$ by letting $q(x_{t+1}^\pi \mid x_t^\pi, \mu_t(x_t^\pi))$ be the probability measure of $x_{t+1}^\pi$ conditioned on $x_t^\pi$. (We construct the Markov process only when Assumption A.4 or A.4′ is in effect.)

For any policy $\pi \in \Pi^\infty$, we define its cost $J_\pi(x)$, as a function of the initial state, by letting

$$J_\pi(x) \stackrel{\text{def}}{=} \mathrm{E}\left\{ \sum_{t=0}^{\infty} \alpha^t g(x_t^\pi, \mu_t(x_t^\pi)) \;\middle|\; x_0^\pi = x \right\}, \quad x \in S. \qquad (2.2.3)$$

(Here, $\mathrm{E}\{\cdot\}$ denotes the expectation with respect to the Markov process.) Accordingly, we call $\inf_{\pi \in \Pi^\infty} J_\pi$ the *optimal cost function* and $\pi^* \in \Pi^\infty$ an *optimal policy* if $J_{\pi^*} = \inf_{\pi \in \Pi^\infty} J_\pi$. We next show that the optimal cost function is the unique solution to Bellman's equation.

### 2.2.2 The Dynamic Programming Operator and Bellman's Equation

Before we show that the optimal cost function is the unique solution to Bellman's equation, we first introduce the dynamic programming operator, stationary policies, and a measurable selection theorem.

We define the *dynamic programming operator* $T : \mathcal{B}(S) \mapsto \mathcal{B}(S)$, by letting

$$TJ(x) \stackrel{\text{def}}{=} \min_{u \in U(x)} \left\{ g(x,u) + \alpha \int_S J(y) q(dy \mid x, u) \right\}, \quad x \in S. \qquad (2.2.4)$$

We have used "min" in Eq. (2.2.4) because the minimum is always attained, which we will show when we prove the following lemma.

**Lemma 2.2.1** *(Under Assumption A.4″) $T$ maps $\mathcal{B}(S)$ into $\mathcal{C}(S)$. In particular, $T$ maps $\mathcal{C}(S)$ into itself.*

**Proof** Fix some $J \in \mathcal{B}(S)$ and define

$$H(x,u) = g(x,u) + \alpha \int_S J(y) q(dy \mid x, u), \quad x \in S, u \in C.$$

For any $x, x' \in S$ and $u, u' \in C$, we have, by Assumption A.1,

$$|g(x,u) - g(x',u')| \le K \|(x,u) - (x',u')\|_\infty$$

and, by Eq. (2.1.13) and Assumption A.2,

$$\left|\int_S J(y)(q(dy \mid x,u) - q(dy \mid x',u'))\right| \leq \|J\|_\infty \|q(\cdot \mid x,u) - q(\cdot \mid x',u')\|_{\mathrm{v}}$$
$$\leq K\|J\|_\infty \|(x,u) - (x',u')\|_\infty. \qquad (2.2.5)$$

Therefore,

$$|H(x,u) - H(x',u')| \leq K\left(1 + \alpha\|J\|_\infty\right)\|(x,u) - (x',u')\|_\infty, \quad \forall x,x' \in S, u,u' \in C.$$

In particular, $H(x,u)$ is a continuous function of $u$. Since $U(x)$ is compact, the minimum in Eq. (2.2.4) is attained by some $u \in U(x)$.

Fix some $x, x' \in S$, and let $v' \in U(x')$ be such that

$$H(x',v') = \min_{u \in U(x')} H(x',u).$$

According to Assumption A.3, there exists some $v \in U(x)$ such that $\|v - v'\|_\infty \leq K\|x - x'\|_\infty$. Thus,

$$\begin{aligned} TJ(x) - TJ(x') &= \min_{u \in U(x)} H(x,u) - \min_{u' \in U(x')} H(x',u') \\ &\leq |H(x,v) - H(x',v')| \\ &\leq K\left(1 + \alpha\|J\|_\infty\right)\|(x,v) - (x',v')\|_\infty \\ &\leq K\left(1 + \alpha\|J\|_\infty\right)K\|x - x'\|_\infty, \quad \forall x,x' \in S. \end{aligned}$$

By symmetry, the same bound applies to $TJ(x') - TJ(x)$ as well. Therefore,

$$|TJ(x) - TJ(x')| \leq K\left(1 + \alpha\|J\|_\infty\right)K\|x - x'\|_\infty, \quad \forall x,x' \in S.$$

This shows that $TJ$ is Lipschitz continuous; in particular, $TJ \in \mathcal{C}(S)$. □

The following well-known facts about the dynamic programming operator $T$ are easily shown.

**Proposition 2.2.1** *There hold*

**(a)** *under Assumption A.4″, $T$ is a contraction operator on $(\mathcal{B}(S), \|\cdot\|_\infty)$ with contraction factor $\alpha$;*

**(b)** *under Assumption A.4′, $T$ is also a monotone operator;*

**(c)** *under Assumption A.4, $T$ also satisfies*

$$T(J + c) = TJ + \alpha c, \quad \forall J \in \mathcal{B}(S), c \in \mathbf{R}; \qquad (2.2.6)$$
$$\|TJ - TJ'\|_{\mathrm{s}} \leq \alpha\|J - J'\|_{\mathrm{s}}, \quad \forall J, J' \in \mathcal{B}(S). \qquad (2.2.7)$$

**Proof** See Denardo (1967), Whitt (1978), or Bertsekas (1987). □

Since $T$ is a contraction operator on the Banach space $(\mathcal{B}(S), \|\cdot\|_\infty)$, the *dynamic programming equation*, also called *Bellman's equation*, $J = TJ$ has a unique solution in $\mathcal{B}(S)$. This solution is denoted by $J^*$.

It then follows from Lemma 2.2.1 that $J^*$ belongs to $\mathcal{C}(S)$. Furthermore, we will show that $J^*$ is actually the optimal cost function. Hence, the optimal cost function is a continuous function and could have been defined as the unique fixed point of the dynamic programming operator $T$.

## Stationary Policies and Their Associated Operators

For any $\mu \in \Pi$, a policy of the form $\pi = (\mu, \mu, \ldots)$ is called a *stationary policy.* When dealing with a stationary policy, we abuse notation and use $\mu$ to denote the policy and $J_\mu$ to denote its expected cost function (instead of using $\pi$ and $J_\pi$, respectively). For any $\mu \in \Pi$, we define the operator $T_\mu : \mathcal{B}(S) \mapsto \mathcal{B}(S)$, by letting

$$T_\mu J(x) \overset{\text{def}}{=} g\big(x, \mu(x)\big) + \alpha \int_S J(y) q\big(y \mid x, \mu(x)\big), \quad J \in \mathcal{B}(S), x \in S. \tag{2.2.8}$$

We check that

**Lemma 2.2.2** *(Under Assumption A.4″) $T_\mu$ maps $\mathcal{B}(S)$ into itself.*

**Proof** Fix some $J \in \mathcal{B}(S)$ and define $H(x,u)$ as in the proof of Lemma 2.2.1. Since $H(x,u)$ is a continuous function of $(x,u)$ and $\mu$ is measurable, it follows that $T_\mu J(x) = H(x, \mu(x))$ is a measurable function of $x$, as required. □

Similarly to $T$, we have the following proposition for $T_\mu$.

**Proposition 2.2.2** *For all $\mu \in \Pi$, Proposition 2.2.1 holds for $T_\mu$.*

**Proof** Let $\Pi = \{\mu\}$; the results follow. □

In particular, $T_\mu$ is a contraction operator on $(\mathcal{B}(S), \|\cdot\|_\infty)$ and must have a unique fixed point in $\mathcal{B}(S)$. It is then easily shown from Eq. (2.2.3) that the unique fixed point of $T_\mu$ is $J_\mu$. More importantly, we will show (in Theorem 2.2.1) that there exists an optimal stationary policy.

## A Measurable Selection Theorem

We now introduce an important theorem in continuous-state dynamic programming: a *(Borel) measurable selection theorem.* The following version of the theorem is a special case of Proposition 7.33 in Bertsekas and Shreve (1978).

**Proposition 2.2.3** *Let $X$ be a metric space, $Y$ be a compact metric space, and $D$ be a closed subset of $X \times Y$. Furthermore, let $D_x = \{y \in Y \mid (x,y) \in D\}$ and $D^* = \{x \in X \mid (x,y) \in D \text{ for some } y \in Y\}$. If $f : D \mapsto \mathbf{R}$ is a bounded continuous function, then there exists a measurable $\psi : D^* \mapsto Y$ such that for all $x \in D^*$, $\psi(x) \in D_x$ and $f(x, \psi(x)) = \min_{y \in D_x} f(x,y)$.*

**Proof** See Proposition 7.33, p. 153, in Bertsekas and Shreve (1978). □

An immediate consequence of this theorem is the following useful lemma:

**Lemma 2.2.3** *(Under Assumption A.4″) For any $J \in \mathcal{B}(S)$, there exists a $\mu \in \Pi$ such that $T_\mu J = TJ$; in particular,*

$$TJ = \min_{\mu \in \Pi} T_\mu J, \quad \forall J \in \mathcal{B}(S). \tag{2.2.9}$$

## The Optimal Cost Function and Bellman's Equation

We now show that $J^*$, the fixed point of the dynamic programming operator, is the optimal cost function and that there exists an optimal stationary policy $\mu^* \in \Pi$; namely, $J^* = J_{\mu^*}$. We are mainly interested in this result when $q$ is a stochastic kernel (Assumption A.4). But we will show the following theorem for substochastic kernels (Assumption A.4′).

**Theorem 2.2.1** *(Under Assumption A.4′) There holds $J^* = \inf_{\pi \in \Pi^\infty} J_\pi$; moreover there exists a $\mu^* \in \Pi$ such that $J^* = J_{\mu^*}$.*

**Proof** Under Assumption A.4′, the expectation in Eq. (2.2.3) is well-defined and is monotone. Therefore, we have for any $\pi = (\mu_0, \mu_1, \ldots) \in \Pi^\infty$,

$$\begin{aligned} T_{\mu_0} \cdots T_{\mu_{k-1}} J(x) &= \mathrm{E}\left\{ \sum_{t=0}^{k-1} \alpha^t g\bigl(x_t^\pi, \mu_t(x_t^\pi)\bigr) + \alpha^k J(x_k) \;\Big|\; x_0^\pi = x \right\} \\ &\le \mathrm{E}\left\{ \sum_{t=0}^{k-1} \alpha^t g\bigl(x_t^\pi, \mu_t(x_t^\pi)\bigr) \;\Big|\; x_0^\pi = x \right\} + \alpha^k \|J\|_\infty, \quad \forall x \in S. \end{aligned} \tag{2.2.10}$$

Since $|g(x,u)| \le K$ for all $(x,u) \in S \times C$, we also obtain

$$\begin{aligned} J_\pi(x) &= \mathrm{E}\left\{ \sum_{t=0}^{\infty} \alpha^t g\bigl(x_t^\pi, \mu_t(x_t^\pi)\bigr) \;\Big|\; x_0^\pi = x \right\} \\ &\ge \mathrm{E}\left\{ \sum_{t=0}^{k-1} \alpha^t g\bigl(x_t^\pi, \mu_t(x_t^\pi)\bigr) \;\Big|\; x_0^\pi = x \right\} - \frac{\alpha^k}{1-\alpha} K, \quad \forall x \in S. \end{aligned} \tag{2.2.11}$$

Combining Eqs. (2.2.10) and (2.2.11) gives

$$T_{\mu_0} \cdots T_{\mu_{k-1}} J \le J_\pi + \alpha^k \left( \|J\|_\infty + \frac{K}{1-\alpha} \right), \quad \forall J \in \mathcal{B}(S). \tag{2.2.12}$$

By definition, $TJ \le T_\mu J$ for all $\mu \in \Pi$; therefore, using the monotone property of $T$, we obtain

$$J^* \le T_{\mu_0} \cdots T_{\mu_{k-1}} J^*, \quad \forall k \in \{1, 2, \ldots\}. \tag{2.2.13}$$

Combining Eqs. (2.2.12) and (2.2.13) gives

$$J^* \le J_\pi + \alpha^k \left( \|J^*\|_\infty + \frac{K}{1-\alpha} \right), \quad \forall \pi \in \Pi,\ k \in \{1, 2 \ldots\}. \tag{2.2.14}$$

Since $\pi$ and $k$ are arbitrary, it follows that $J^* \le J_\pi$, for all $\pi \in \Pi$.

By Lemma 2.2.3 there exists a $\mu^* \in \Pi$ such that $T_{\mu^*} J^* = TJ^* = J^*$. But since $T_{\mu^*}$ is a contraction operator, we must have $J^* = J_{\mu^*}$. This shows that $J^*$ is the optimal cost function and there exists an optimal stationary policy, as required. □

With the above theorem, we can restrict attention to stationary policies and from now on, the word "policy" should be interpreted as "stationary policy".

### 2.2.3 Problem Statement

We are interested in the computation of $J^*$ and of a corresponding optimal policy $\mu^*$. This can be accomplished, in principle, by solving Bellman's equation $J = TJ$. However, since Bellman's equation is infinite-dimensional and nonlinear, we have to be content with computing approximations to $J^*$ and $\mu^*$.

Let $\epsilon > 0$. A function $J \in \mathcal{B}(S)$ that satisfies the inequality $\|J - J^*\|_\infty \leq \epsilon$ is called *an $\epsilon$-approximation* of $J^*$ or *an $\epsilon$-optimal cost function.* And a policy $\mu$ is called *an $\epsilon$-optimal policy,* if $\|J_\mu - J^*\|_\infty \leq \epsilon$. For now, we will mainly focus on the computation of an $\epsilon$-approximation of $J^*$.

We are interested in the *complexity* (that is, the computational requirements) of computing an $\epsilon$-optimal cost function as a function of $\epsilon$ and $\alpha$, in the limit as $\epsilon \downarrow 0$ and $\alpha \uparrow 1$. Furthermore, we are also interested in how this complexity depends on any ergodicity conditions the problem may satisfy. In the next section, we will look at some ergodicity conditions.

### 2.2.4 Notes

Most authors consider more general formulations of MDP's than ours. Our formulation is similar to that in Hernández-Lerma (1989); in fact, our model is a special case of a certain continuous model there. However, the model we have now is still too general for our purpose: namely, to analyze the computational complexity of MDP's. Additional assumptions will be introduced later.

A standard reference on MDP's and general continuous-state dynamic programming is Bertsekas and Shreve (1978). This book considers even more general models and discusses measurability issues in considerable detail.

To formally construct the Markov process $\{x_t^\pi \mid t = 0, 1, \ldots\}$ requires a certain technical lemma. This is handled by Proposition 7.28, p. 140, in Bertsekas and Shreve (1978) [or see Section 2.7 of Ash (1972)]. The proposition also gives a precise definition to the expectation in Eq. (2.2.3).

Note that when $q$ is a substochastic kernel, we first augment the state space to get an equivalent stochastic kernel. We then construct a Markov process using the new kernel.

Lemma 2.2.1 and Theorem 2.2.1 are well known results. For more general results see Bertsekas and Shreve (1978).

Markov Decision Processes with discrete state and control spaces are usually called Markov Decision Problems. Our model includes finite-state Markov Decision Problems as a special case.

There are several limitations to our model. One of them is that the cost function and the state and the control spaces must be bounded. At present we do not have a suitable framework for analyzing the complexity of problems with unbounded cost or with unbounded state and control spaces. Developing one is an area of future research.

## 2.3 The Effects of Ergodicity Conditions

In this section, we consider a special case where the dynamics are described by a stochastic kernel and, in addition, satisfy certain ergodicity (or mixing) conditions. These conditions lead to faster convergence in the successive approximation algorithm (to be introduced in Chapter 3). Moreover, they also result in a more accurate problem discretization. The results here are direct generalizations of the well-known, well-studied results in finite-state problems; the proofs here are also adapted from there. For the rest of the section, we assume that $q$ is a stochastic kernel (Assumption A.4).

An outline of the section is as follows. We begin by introducing more notation and prove a useful fact about stochastic kernels. We then discuss the ergodicity conditions and show how they relate to each other. Next, we show that certain ergodicity conditions result in an extra contraction factor in the dynamic programming operator $T$; this in turn improves the convergence of successive approximation. Finally, we show that certain ergodicity conditions allow us to bound $\|J^*\|_s$ independently of the discount factor $\alpha$; we will use this result in Section 2.4 to obtain better discretization error bounds.

### 2.3.1 Preliminaries

We now introduce more notation. For any policy $\mu \in \Pi$, let

$$g_\mu(x) \overset{\text{def}}{=} g(x, \mu(x)), \quad x \in S;$$
$$q_\mu(\cdot \mid x) \overset{\text{def}}{=} q(\cdot \mid x, \mu(x)), \quad x \in S.$$

It is clear that $g_\mu$ is a measurable function and $q_\mu$ is a measurable stochastic kernel.

We now prove a useful result for stochastic kernels.

**Lemma 2.3.1** *Let $q_0$ and $q_1$ be kernels on $S$ given $S$. If $q_0$ satisfies*

$$q_0(S \mid x) = 0, \quad \forall x \in S, \tag{2.3.1}$$

*and $q_1$ is stochastic and satisfies*

$$\|q_1(\cdot \mid x) - q_1(\cdot \mid x')\|_v \leq 2(1-\rho), \quad \forall x, x' \in S, \tag{2.3.2}$$

*then*

$$\|q_0 q_1\|_\infty \leq (1-\rho)\|q_0\|_\infty. \tag{2.3.3}$$

**Proof** Fix a $J \in \mathcal{B}(S)$ and let $c = -[\sup_x J(x) + \inf_x J(x)]/2$. Using Eq. (2.3.1), we have

$$\begin{aligned} \|q_0 q_1 J\|_\infty &= \|q_0(q_1 J + c)\|_\infty \\ &\leq \|q_0\|_\infty \|q_1 J + c\|_\infty \\ &= \|q_0\|_\infty \|q_1 J\|_s / 2. \end{aligned} \tag{2.3.4}$$

Fix $x, x' \in S$. Using Eq. (2.3.2), we obtain

$$\begin{aligned}|q_1 J(x) - q_1 J(x')| &= \left|\int_S (q_1(dy \mid x) - q_1(dy \mid x'))J(y)\right| \\ &\leq \|q_1(dy \mid x) - q_1(dy \mid x')\|_{\mathrm{v}} \|J\|_\infty \\ &\leq 2(1-\rho)\|J\|_\infty.\end{aligned}$$

Since $x, x'$ are arbitrary, we have

$$\|q_1 J\|_{\mathrm{s}} \leq 2(1-\rho)\|J\|_\infty.$$

Using this in Eq. (2.3.4), we obtain

$$\|q_0 q_1 J\|_\infty \leq (1-\rho)\|q_0\|_\infty \|J\|_\infty.$$

The result follows by taking the supremum over all $J \in \mathcal{B}(S)$ with $\|J\|_\infty \leq 1$. $\square$

### 2.3.2 Ergodicity Conditions

We now consider, in increasing generality, the following ergodicity conditions.

**E.1** There exist a scalar $\rho \in (0,1]$ and a measure $\nu$ with $\nu(S) \geq \rho$ such that

$$q(\cdot \mid x, u) \geq \nu(\cdot), \quad \forall (x,u) \in \Gamma.$$

**E.2** There exists a scalar $\rho \in (0,1]$ such that

$$\|q(\cdot \mid x,u) - q(\cdot \mid x', u')\|_{\mathrm{v}} \leq 2(1-\rho), \quad \forall (x,u), (x',u') \in \Gamma.$$

**E.3** There exist an integer $k \geq 1$ and a scalar $\rho \in (0,1]$ such that for any policies $\mu_0, \mu_1, \ldots \mu_{k-1}, \mu'_0, \mu'_1, \ldots, \mu'_{k-1} \in \Pi$, there holds

$$\|q_{\mu_0} q_{\mu_1} \cdots q_{\mu_{k-1}}(\cdot \mid x) - q_{\mu'_0} q_{\mu'_1} \cdots q_{\mu'_{k-1}}(\cdot \mid x')\|_{\mathrm{v}} \leq 2(1-\rho), \quad \forall x, x' \in S.$$

**E.3′** There exist an integer $k \geq 1$ and a scalar $\rho \in (0,1]$ such that for any policies $\mu_0, \mu_1, \ldots \mu_{k-1}, \mu'_0, \mu'_1, \ldots, \mu'_{k-1} \in \Pi$, there holds

$$\|q_{\mu_0} q_{\mu_1} \cdots q_{\mu_{k-1}}(\cdot \mid x) \wedge q_{\mu'_0} q_{\mu'_1} \cdots q_{\mu'_{k-1}}(\cdot \mid x')\|_{\mathrm{v}} \geq \rho, \quad \forall x, x' \in S.$$

**E.4** There exist an integer $k \geq 1$ and a scalar $\rho \in (0,1]$ such that for any policies $\mu_0, \mu_1, \ldots \mu_k \in \Pi$, there holds

$$\|q_{\mu_0} q_{\mu_1} \cdots q_{\mu_{k-1}}(\cdot \mid x) - q_{\mu_1} q_{\mu_2} \cdots q_{\mu_{k-1}}(\cdot \mid x')\|_{\mathrm{v}} \leq 2(1-\rho).$$

**E.5** There exists a constant $K_e < \infty$ such that for any policy $\mu \in \Pi$, there exists a probability measure $\nu_\mu^*$ satisfying

$$\sum_{t=0}^{\infty} \|q^t(\cdot \mid x) - \nu_\mu^*\|_{\mathrm{v}} \leq K_e, \quad \forall x \in S.$$

(Note that $K_e$ is independent of $\mu$ and $x$.)

**E.6** For any policy $\mu \in \Pi$, there exists a probability measure $\nu_\mu^*$ satisfying

$$\lim_{t \to \infty} \|q^t(\cdot \mid x) - \nu_\mu^*\|_{\mathrm{v}} = 0, \quad \forall x \in S.$$

(Or equivalently, $q_\mu$ is *ergodic*.).

The relation between the above ergodicity conditions are given below

**Theorem 2.3.1** *(Under Assumption A.4) There hold*

$$E.1 \Rightarrow E.2 \Rightarrow (E.3 \Leftrightarrow E.3') \Rightarrow E.4 \Rightarrow E.5 \Rightarrow E.6.$$

**Proof** We will prove the implications from left to right.

To prove E.1 $\Rightarrow$ E.2, we fix some $(x,u),(x',u') \in \Gamma$ and let $\nu_1(\cdot) = q(\cdot \mid x,u)$ and $\nu_2(\cdot) = q(\cdot \mid x',u')$. By the Jordan-Hahn decomposition theorem (Proposition 2.1.2), we can partition $S$ into disjoint $S^+$ and $S^-$ whose union is $S$, such that

$$\begin{aligned}
\|\nu_1 - \nu_2\|_{\mathrm{v}} &= \nu_1(S^+) - \nu_2(S^+) - \nu_1(S^-) + \nu_2(S^-) \\
&\leq [1 - \nu(S^-)] - \nu(S^+) - \nu(S^-) + [1 - \nu(S^+)] \\
&= 2[1 - \nu(S)] \\
&\leq 2(1 - \rho),
\end{aligned}$$

as required.

It is clear that E.2 $\Rightarrow$ E.3$'$. We now verify E.3 $\Leftrightarrow$ E.3$'$. The equivalence between the two is seen by letting

$$\begin{aligned}
\nu_1(\cdot) &= q_{\mu_0} q_{\mu_1} \cdots q_{\mu_{k-1}}(\cdot \mid x); \\
\nu_2(\cdot) &= q_{\mu_0'} q_{\mu_1'} \cdots q_{\mu_{k-1}'}(\cdot \mid x').
\end{aligned}$$

Using Lemma 2.1.1, we have

$$\begin{aligned}
\|\nu_1 - \nu_2\|_{\mathrm{v}} &= \nu_1(S) + \nu_2(S) - 2\|\nu_1 \wedge \nu_2\|_{\mathrm{v}} \\
&= 2(1 - \|\nu_1 \wedge \nu_2\|_{\mathrm{v}}),
\end{aligned}$$

and the result follows (with the same $\rho$ in both cases).

Since E.4 is a generalization of E.3, the implication (E.3 $\Leftrightarrow$ E.3$'$) $\Rightarrow$ E.4 follows immediately.

We now prove E.4 $\Rightarrow$ E.5. To show the existence of $\nu_\mu^*$, we first show that $q_\mu^t$ is a Cauchy sequence with respect to $\|\cdot\|_\infty$. By Assumption E.4, we have

$$\|q_\mu^k(\cdot \mid x) - q_\mu^k(\cdot \mid x')\|_{\mathrm{v}} \le 2(1-\rho), \quad \forall x, x' \in S.$$

Now using Lemma 2.3.1, we obtain

$$\begin{aligned}\|q_\mu^k - q_\mu^{k+t'}\|_\infty &= \|(q_\mu^0 - q_\mu^{t'})q_\mu^k\|_\infty \\ &\le (1-\rho)\|(q_\mu^0 - q_\mu^t)\|_\infty \\ &\le 2(1-\rho), \quad \forall t' \ge 0,\end{aligned}$$

where we have used the fact the $(q_\mu^0 - q^{t'})$ satisfies Eq. (2.3.1). More generally, we have

$$\|q_\mu^t - q_\mu^{t+t'}\|_\infty \le (1-\rho)^{\lfloor t/k \rfloor} 2, \quad \forall t, t' \ge 0.$$

Thus, it follows that $q^t$ is a Cauchy sequence.

The space of all stochastic kernels is a Banach space with respect to $\|\cdot\|_\infty$; therefore, there exists a stochastic kernel $q_\mu^*$ such that $\lim_{t\to\infty} \|q_\mu^t - q_\mu^*\|_\infty = 0$. We now show that $q_\mu^*(\cdot \mid x) = q_\mu^*(\cdot \mid x')$ for all $x, x' \in S$. Using the triangle inequality, we have

$$\begin{aligned}\|q_\mu^*(\cdot \mid x) - q_\mu^*(\cdot \mid x')\|_{\mathrm{v}} &\le \|q_\mu^*(\cdot \mid x) - q_\mu^t(\cdot \mid x)\|_{\mathrm{v}} + \|q_\mu^t(\cdot \mid x) - q_\mu^t(\cdot \mid x')\|_{\mathrm{v}} \\ &\quad + \|q_\mu^t(\cdot \mid x') - q_\mu^*(\cdot \mid x')\|_{\mathrm{v}} \\ &\le 2\|q_\mu^* - q_\mu^t\|_\infty + \|q_\mu^t(\cdot \mid x) - q_\mu^t(\cdot \mid x')\|_{\mathrm{v}} \\ &\le 2\|q_\mu^* - q_\mu^t\|_\infty + (1-\rho)^{\lfloor t/k \rfloor} 2, \quad \forall t \ge 0.\end{aligned}$$

where we have used Lemma 2.3.1 to obtain the last inequality. By letting $t$ go to infinity, we obtain $\|q_\mu^*(\cdot \mid x) - q_\mu^*(\cdot \mid x')\|_{\mathrm{v}} = 0$ which gives the desired result. Thus, we can let $\nu_\mu^* = q_\mu^*(\cdot \mid x)$.

Finally, we bound the constant $K_e$. We have

$$\begin{aligned}\|q_\mu^t - q_\mu^*\|_\infty &\le \|q_\mu^t - q_\mu^{t+t'}\|_\infty + \|q_\mu^{t+t'} - q_\mu^*\|_\infty \\ &\le (1-\rho)^{\lfloor t/k \rfloor} 2 + \|q_\mu^{t+t'} - q_\mu^*\|_\infty, \quad \forall t, t' \ge 0.\end{aligned}$$

Letting $t'$ go to infinity, we obtain

$$\|q_\mu^t - q_\mu^*\|_\infty \le (1-\rho)^{\lfloor t/k \rfloor} 2, \quad \forall t \ge 0.$$

Therefore, for any $x \in S$, $\mu \in \Pi$, we have

$$\begin{aligned}\sum_{t=0}^{\infty} \|q_\mu^t(\cdot \mid x) - \nu_\mu^*\|_{\mathrm{v}} &\le \sum_{t=0}^{\infty} \|q_\mu^t - q_\mu^*\|_\infty \\ &\le 2\sum_{t=0}^{\infty} (1-\rho)^{\lfloor t/k \rfloor} \\ &\le 2k/\rho.\end{aligned}$$

Therefore, $K_e = 2k/\rho$ is the required constant. Finally, it is clear that E.5 $\Rightarrow$ E.6. □

We call Assumption E.1 a *1-stage ergodicity condition,* Assumption E.3 a *k-stage ergodicity condition,* and $\rho$ the *ergodicity rate.*

Note that when $k = 1$, Assumptions E.3 and E.4 are equivalent to Assumption E.2, which is weaker than Assumption E.1.

In this section we study the relationship between the different ergodicity conditions and some of their consequences. In the subsequent chapters we will only use Assumptions E.1 and E.3.

### 2.3.3 Ergodicity and Span-Norm Contraction

We now show that Assumptions E3 and E.4 lead to an additional (span-norm) contraction factor, independent of $\alpha$, in the dynamic programming operator $T$.

**Theorem 2.3.2** *(Under Assumption A.4) For all integer $t \geq 0$, if the transition kernel $q$ satisfies*

**(a)** *Assumption E.3, then*

$$\|T^tJ - T^tJ'\|_s \leq \alpha^t(1-\rho)^{\lfloor t/k \rfloor}\|J - J'\|_s, \quad \forall J, J' \in \mathcal{B}(S).$$

**(b)** *Assumption E.4, then*

$$\|T^{t+1}J - T^tJ\|_s \leq \alpha^t(1-\rho)^{\lfloor t/k \rfloor}\|TJ - J\|_s, \quad \forall J \in \mathcal{B}(S).$$

**Proof** We first prove (a) when $t = k$, the general result follows immediately. Let $\mu_0, \mu_1, \dots \mu_{k-1}$ (respectively, $\mu'_0, \mu'_1, \dots \mu'_{k-1}$) be the policies that attain the minimum in $T^kJ$ (respectively, $T^kJ'$); that is,

$$T^kJ = T_{\mu_0}T_{\mu_1}\cdots T_{\mu_{k-1}}J;$$
$$T^kJ' = T_{\mu'_0}T_{\mu'_1}\cdots T_{\mu'_{k-1}}J'.$$

Thus,

$$\begin{aligned} T^kJ - T^kJ' &= \left\{g_{\mu_0} + \alpha q_{\mu_0}T^{k-1}J\right\} - \left\{g_{\mu'_0} + \alpha q_{\mu'_0}T^{k-1}J'\right\} \\ &\leq \left\{g_{\mu'_0} + \alpha q_{\mu'_0}T^{k-1}J\right\} - \left\{g_{\mu'_0} + \alpha q_{\mu'_0}T^{k-1}J'\right\} \\ &= \alpha\left\{q_{\mu'_0}(T^{k-1}J - T^{k-1}J')\right\}. \end{aligned}$$

Repeating the above procedure $k-1$ more times,

$$T^kJ - T^kJ' \leq \alpha^k\left\{q_{\mu'_0}\cdots q_{\mu'_{k-1}}(J - J')\right\}. \tag{2.3.5}$$

A symmetrical argument yields,

$$T^kJ - T^kJ' \geq \alpha^k\left\{q_{\mu_0}\cdots q_{\mu_{k-1}}(J - J')\right\}. \tag{2.3.6}$$

Fix some $x, x' \in S$. For brevity, let $H = T^kJ - T^kJ'$,

$$\nu' = q_{\mu'_0}\cdots q_{\mu'_{k-1}} \quad \text{and} \quad \nu = q_{\mu_0}\cdots q_{\mu_{k-1}}.$$

Using Eqs. (2.3.5), (2.3.6), and Assumptions E.3, we obtain

$$\begin{aligned} H(x') - H(x) &\leq \alpha^k \left[ \int_S \nu'(dy)(J - J')(y) - \int_S \nu(dy)(J - J')(y) \right] \\ &\leq \alpha^k \|\nu' - \nu\|_{\mathrm{v}} \|J - J'\|_\infty \\ &\leq \alpha^k (1 - \rho) 2 \|J - J'\|_\infty. \end{aligned}$$

Since $x$ and $x'$ are arbitrary, we have

$$\|T^k J - T^k J'\|_{\mathrm{s}} \leq \alpha^k (1 - \rho) 2 \|J - J'\|_\infty. \tag{2.3.7}$$

Replacing $J$ by $J + c$ in Eq. (2.3.7), where

$$c = -[\sup_x (J - J')(x) + \inf_x (J - J')(x)]/2,$$

and using Proposition 2.2.1(c), we obtain

$$\begin{aligned} \|T^k J - T^k J'\|_{\mathrm{s}} &= \|T^k (J + c) - T^k J'\|_{\mathrm{s}} \\ &\leq \alpha^k (1 - \rho) 2 \|J + c - J'\|_\infty \\ &= \alpha^k (1 - \rho) \|J - J'\|_{\mathrm{s}}, \end{aligned}$$

as required.

The proof for (b) is similar. Let the minimizing policies in $T^{k+1}J$ be attained by $\mu_0, \mu_1 \ldots, \mu_k$. We argue as before to obtain

$$T^{k+1}J - T^k J \leq q_{\mu_1} \cdots q_{\mu_k} (TJ - J)$$

and

$$T^{k+1}J - T^k J \geq q_{\mu_0} \cdots q_{\mu_{k-1}} (TJ - J).$$

Now following the same argument used in (a), we obtain,

$$\|T^{k+1}J - T^k J\|_{\mathrm{s}} \leq \alpha^k (1 - \rho) \|TJ - J\|_{\mathrm{s}}.$$

The proof for general values of $t$ follows immediately. □

Note that under Assumption E.3, the dynamic programming operator $T$ is a $k$-stage contraction operator on $(\mathcal{B}(S), \|\cdot\|_{\mathrm{s}})$ with contraction factor $(1-\rho)$, independent of $\alpha$. We will see in Chapter 3 that this results in a better convergence rate in the successive approximation algorithm. Actually Assumption E.4 is sufficient for this purpose; in fact, most of the results in Chapter 3 can be shown under Assumption E.4 instead of E.3. But we will not do this to keep the proofs simple.

We now show that under Assumption E.5, we can bound $\|J^*\|_{\mathrm{s}}$ by a constant independent of $\alpha$. We will see in the next section that this bound leads to a better discretization error.

**Theorem 2.3.3** *(Under Assumption A.4) If the transition kernel $q$ satisfies Assumption E.5, then $\|J^*\|_s \le 2KK_e$, independent of $\alpha$, where $K_e$ is the constant of E.5.*

**Proof** By the measurable selection theorem (Proposition 2.2.3), there exists an optimal policy $\mu^* \in \Pi$ so that

$$J^* = J_{\mu^*} = \sum_{t=0}^{\infty} \alpha^t q_{\mu^*}^t g_{\mu^*}.$$

For any $x, x' \in S$, we have

$$\begin{aligned}
|J^*(x) - J^*(x')| &\le \sum_{t=0}^{\infty} \alpha^t \left| \int_S q_{\mu^*}^t(dy \mid x) g_{\mu^*}(y) - \int_S q_{\mu^*}^t(dy \mid x') g_{\mu^*}(y) \right| \\
&\le \sum_{t=0}^{\infty} \|q_{\mu^*}^t(\cdot \mid x) - q_{\mu^*}^t(\cdot \mid x')\|_V K \\
&\le K \sum_{t=0}^{\infty} \|q_{\mu^*}^t(\cdot \mid x) - \nu_{\mu^*}^*\|_V + \|\nu_{\mu^*}^* - q_{\mu^*}^t(\cdot \mid x')\|_V \\
&\le 2KK_e.
\end{aligned}$$

Therefore, $\|J^*\|_s \le 2KK_e$, independent of $\alpha$. □

## 2.3.4 Notes

Ergodicity conditions are well studied for finite-state MDP's (usually in the context of average cost problems). Federgruen, et al, (1978) and Schweitzer (1988) study (finite-state) ergodicity conditions and their relation to span-norm contraction. The two papers also consider necessary and sufficient conditions for span-norm contraction. Assumption E.3′ is a continuous-state formulation of a condition studied in Federgruen, et al, (1978).

Assumption E.4 is a continuous-state formulation of a condition in Bertsekas (1987). He also discusses a finite-state version of Assumption E.6 and shows how this condition leads to span-norm contraction in successive approximation. (We have not been able to establish such a connection for continuous-state problems.)

Assumption E.3 is a $k$-stage version (a generalization) of a condition studied by Hernández-Lerma (1989). Hernández-Lerma studies continuous-state ergodicity conditions in the context of average cost problems and proves a special case of Theorem 2.3.1. Our proof of Theorem 2.3.1 is motivated by this result.

Ergodicity conditions are also studied in the context of Markov chains. See for example Seneta (1981). Assumption E.3′ is known as a scrambling-type condition in this context.

Our definition of ergodicity (in Assumption E.6) is taken from Nummelin (1984). This book also discusses other ergodicity conditions in the continuous-state context. Assumption E.5 is related to a uniformly ergodic condition discuss there.

Going from finite-state to continuous-state ergodicity conditions introduces technical difficulties. We have been able to avoid most of the difficulties by establishing the existence of a minimizing policy in $TJ$.

## 2.4 Discretization Procedures and Error Bounds

The computation of an $\epsilon$-approximation of $J^*$ is usually accomplished by discretizing the original problem and by constructing a new MDP that has finite state and control spaces. However, since we will be comparing functions corresponding to different discretization levels, it is both conceptually and notationally simpler for us to consider MDP's that involve simple functions on $S$ rather than functions on finite subsets of $S$. In this section, we construct such a discretization.

First, we describe the procedures for discretizing the MDP. Second, we show that ergodicity conditions in the original problem are "inherited" by the discretized problems (when the discretization is sufficiently fine). Third, we prove the main theorems of this section—bounds on the discretization error—that is, we estimate the inaccuracy in discretization as a function of the grid-spacing and of the discount parameter $\alpha$. We also show how ergodicity conditions reduce the discretization error. Finally, we compare with other discretization procedures in the literature, and give some references.

### 2.4.1 Discretization of an MDP

We now consider the discretization of an MDP. We first show how the state and control spaces are discretized. We then consider the discretization of the costs and the dynamics. Next, we look at the discretization of Bellman's equation. Finally, we consider the discretization of policies.

#### Discretization of the State and Control Spaces

Let $h \in (0,1]$ be a scalar that parametrizes the coarseness of discretization; we call $h$ the *grid-size* or the *grid-level.* We start by partitioning the unit interval $I = [0,1]$ into a collection $\mathcal{I}_h$ of subsets. In particular, $\mathcal{I}_h$ consists of the set $[0,h]$ together with all nonempty sets of the form $(ih,(i+1)h] \cap I$, $i = 1,2,\ldots$. We then partition the unit $n$-dimensional cube $[0,1]^n$ into a collection $\mathcal{I}_h^n$ of subsets defined by

$$\mathcal{I}_h^n \stackrel{\text{def}}{=} \left\{ I_1 \times \cdots \times I_n \;\middle|\; I_i \in \mathcal{I}_h \right\}.$$

We discretize the state space by partitioning it into a finite collection of subsets. Each set in this partition is the intersection of $S$ with an element of $\mathcal{I}_h^n$. More precisely, we let $\mathcal{S}_h$ be the set of all nonempty sets $\sigma$ of the form $\sigma = S \cap \iota$, $\iota \in \mathcal{I}_h^n$, and these sets form the desired partition. We choose a representative element from each $\sigma \in \mathcal{S}_h$ and we let $\tilde{S}_h$ be the set of all representatives. For any $x \in S$, we let $\sigma_x$ be the element of $\mathcal{S}_h$ to which $x$ belongs. We also use $\breve{\sigma}_x$ to denote the representative of the set $\sigma_x$.

The control space is discretized by letting $\tilde{C}_h$ be the set of all $(u_1,\ldots,u_m) \in C$ such that each $u_i$ is an integer multiple of $h$. The set of admissible discretized controls is defined by

$$\tilde{U}_h(x) = \left\{ \tilde{u} \in \tilde{C}_h \;\middle|\; \|u - \tilde{u}\|_\infty \leq \frac{h}{2} \text{ for some } u \in U(\breve{\sigma}_x) \right\}, \quad x \in S. \tag{2.4.1}$$

For any $x \in S$, the set $U(\check{\sigma}_x)$ is nonempty, by assumption. Furthermore, using the definition of $\tilde{C}_h$, we see that for any $u \in U(\check{\sigma}_x)$ there exists some $\tilde{u} \in \tilde{C}_h$ such that $\|u - \tilde{u}\|_\infty \leq h/2$. Thus, the set $\tilde{U}_h(x)$ is nonempty for each $x \in S$. It is also easy to see that

$$\tilde{U}_h(x) = \tilde{U}_h(x') = \tilde{U}_h(\check{\sigma}_x), \quad \forall x \in S, \; \forall x' \in \sigma_x. \tag{2.4.2}$$

Lastly, let

$$\tilde{\Gamma}_h = \left\{ (x, \tilde{u}) \mid x \in S \text{ and } \tilde{u} \in \tilde{U}_h(x) \right\}. \tag{2.4.3}$$

**Discretization of the Cost and the Dynamics**

Given some $h \in (0,1]$, we define the discretized cost function $\tilde{g}_h : S \times \tilde{C}_h \mapsto \mathbf{R}$ by letting

$$\tilde{g}_h(x, \tilde{u}) \overset{\text{def}}{=} g(\check{\sigma}_x, \tilde{u}). \tag{2.4.4}$$

It follows from Assumption A.1 that

$$|g(x, \tilde{u}) - \tilde{g}_h(x, \tilde{u})| \leq Kh, \quad \forall x \in S, \tilde{u} \in \tilde{C}_h.$$

We now consider the discretization of the dynamics. We are primarily interested in the case where $h$ is small. We can therefore assume that $h \leq h_a$, where $h_a \in (0,1]$ is a constant to be determined later. We will also assume that for all $h \in (0, h_a]$, we have a kernel $\tilde{q}_h$ on $S$ given $\tilde{S}_h \times \tilde{C}_h$ satisfying the following discretization conditions.

**D.1** If $q$ satisfies Assumption A.4, A.4$'$, or A.4$''$ then $\tilde{q}_h$ satisfies Assumption A.4, A.4$'$, or A.4$''$, respectively.

**D.2** There exists some constant $K_q \geq 1$ such that

$$\|q(\cdot \mid \tilde{x}, \tilde{u}) - \tilde{q}_h(\cdot \mid \tilde{x}, \tilde{u})\|_{\mathrm{v}} \leq K_q h, \quad \tilde{x} \in \tilde{S}_h, \tilde{u} \in \tilde{C}_h.$$

We will construct $\tilde{q}_h$ from $q$ later after we introduce more assumptions on $q$; we will also determine the values of $h_a$ and $K_q$ then (see Section 2.5).

Given the above $\tilde{q}_h$, we can extend $\tilde{q}_h(\cdot \mid x, \tilde{u})$ to all $(x, \tilde{u}) \in S \times \tilde{C}_h$ by letting

$$\tilde{q}_h(\cdot \mid x, \tilde{u}) \overset{\text{def}}{=} \tilde{q}_h(\cdot \mid \check{\sigma}_x, \tilde{u}), \quad x \in S, \tilde{u} \in \tilde{C}_h. \tag{2.4.5}$$

This extended kernel $\tilde{q}_h$ on $S$ given $S \times \tilde{C}_h$ is the discretized kernel. It follows from Assumptions A.2 and D.2 that

$$\|q(\cdot \mid x, \tilde{u}) - \tilde{q}_h(\cdot \mid x, \tilde{u})\|_{\mathrm{v}} \leq (K + K_q)h, \quad \forall h \in (0, h_a], x \in \tilde{S}_h, \tilde{u} \in \tilde{C}_h. \tag{2.4.6}$$

Finally, a discretized MDP is specified by $\left(S, \tilde{C}_h, \{\tilde{U}_h(x)\}, \tilde{g}_h, \tilde{q}_h, \alpha\right)$ with the additional information $S_h$ and $\tilde{S}_h$.

## The Discretized Bellman's Equation

We now consider the discretization of Bellman's equation. We have so far constructed a discretized MDP $\left(S, \tilde{C}_h, \{\tilde{U}_h(x)\}, \tilde{g}_h, \tilde{q}_h, \alpha\right)$. The dynamic programming operator $\tilde{T}_h : \mathcal{B}(S) \mapsto \mathcal{B}(S)$ corresponding to this problem is defined by

$$\tilde{T}_h J(x) \stackrel{\text{def}}{=} \min_{\tilde{u} \in \tilde{U}_h(x)} \left\{ \tilde{g}_h(x, \tilde{u}) + \alpha \int_S \tilde{q}_h(dy \mid x, \tilde{u}) J(y) \right\}, \quad J \in \mathcal{B}(S). \tag{2.4.7}$$

By Assumption D.1, $\tilde{T}_h$ has the same properties as $T$ (cf. Proposition 2.2.1), except that $\tilde{T}_h$ maps measurable functions into simple functions, and therefore the fixed point of $\tilde{T}_h$, denoted by $\tilde{J}_h^*$, is a simple function. To see this we introduce the following terminology.

Given the partition $\mathcal{S}_h$ of the state space $S$, we say that a function $f$ with domain $S$ is a *simple function on* $\mathcal{S}_h$ if $f$ is constant on each element of $\mathcal{S}_h$. That is, $f(x) = f(x')$ for every $\sigma \in \mathcal{S}_h$ and every $x, x' \in \sigma$.

For any fixed $\tilde{u} \in \tilde{C}_h$, the functions $\tilde{g}_h(\cdot, \tilde{u})$ and $\int_S \tilde{q}_h(dy \mid \cdot, \tilde{u}) J(y)$ are simple on $\mathcal{S}_h$. It follows from Eq. (2.4.7) that for any $J \in \mathcal{B}(S)$, $\tilde{T}_h J$ is a simple function on $\mathcal{S}_h$.

Since simple functions on $\mathcal{S}_h$ form a Banach space, the fixed point of $\tilde{T}_h$ must also be a simple function on $\mathcal{S}_h$; in particular, there exists a unique simple function on $\mathcal{S}_h$, denoted by $\tilde{J}_h^*$, that solves the discretized Bellman's equation $J = \tilde{T}_h J$.

It is clear that the discretized problem $(S, \tilde{C}_h, \{\tilde{U}_h(x)\}, \tilde{g}_h, \tilde{q}_h, \alpha)$ is equivalent to a finite-state MDP $(\tilde{S}_h, \tilde{C}_h, \{\tilde{U}_h(x)\}, \tilde{g}_h, \tilde{q}_h, \alpha)$. To this latter problem, we can associate an optimal cost function with values $\tilde{J}_h^*(\tilde{x})$ for $\tilde{x} \in \tilde{S}_h$. For our purpose, however, it is easier to work with the state space $S$, rather than $\tilde{S}_h$, because $\tilde{J}_h^*$ and $J^*$ are defined on the same set $S$ and can be directly compared.

## Discretization of Policies

For the discretized problem, the set of policies is defined as

$$\tilde{\Pi}_h \stackrel{\text{def}}{=} \left\{ \tilde{\mu}_h : S \mapsto \tilde{C}_h \;\middle|\; \tilde{\mu}_h \text{ is a simple function on } \mathcal{S}_h \text{ and } \tilde{\mu}_h(x) \in \tilde{U}_h(x),\ \forall x \in S \right\}. \tag{2.4.8}$$

As in the original MDP, for any policies $\tilde{\mu} \in \tilde{\Pi}_h$ we can associate an operator $\tilde{T}_{\tilde{\mu}} : \mathcal{B}(S) \mapsto \mathcal{B}(S)$ by letting

$$\tilde{T}_{\tilde{\mu}} J = \tilde{g}_{\tilde{\mu}} + \alpha \tilde{q}_{\tilde{\mu}} J, \quad J \in \mathcal{B}(S),$$

where

$$\tilde{g}_{\tilde{\mu}}(x) \stackrel{\text{def}}{=} \tilde{g}_h(x, \tilde{\mu}(x)), \quad \forall x \in S;$$

$$\tilde{q}_{\tilde{\mu}}(\cdot \mid x) \stackrel{\text{def}}{=} \tilde{q}_h(\cdot \mid x, \tilde{\mu}(x)), \quad \forall x \in S.$$

We argue as before that for any $J \in \mathcal{B}(S)$, $\tilde{T}_{\tilde{\mu}} J$ is a simple function on $\mathcal{S}_h$ and it follows that the fixed point of $\tilde{T}_{\tilde{\mu}}$, denoted by $\tilde{J}_{\tilde{\mu}}$, is a simple function on $\mathcal{S}_h$.

Moreover, it is clear (since $\tilde{\Pi}_h$ is a finite set) that for any $J \in \mathcal{B}(S)$ there exists a $\tilde{\mu} \in \tilde{\Pi}_h$ such that $\tilde{T}_h J = \tilde{T}_{\tilde{\mu}} J$; in particular,

$$\tilde{T}_h J = \min_{\tilde{\mu} \in \tilde{\Pi}_h} \tilde{T}_{\tilde{\mu}} J, \quad \forall J \in \mathcal{B}(S).$$

It follows that there exists a policy $\tilde{\mu}^* \in \tilde{\Pi}_h$ such that $\tilde{J}_h^* = \tilde{T}_{\tilde{\mu}^*} \tilde{J}_h^*$; therefore, $\tilde{J}_h^* = \tilde{J}_{\tilde{\mu}^*}$ and $\tilde{\mu}^*$ is an optimal policy.

Finally, we have the following useful lemma:

**Lemma 2.4.1** *For any $\tilde{\mu} \in \tilde{\Pi}_h$, there exists a $\mu \in \Pi$ such that $\|\tilde{\mu} - \mu\|_\infty \le (K + 1/2)h$.*

**Proof** Fix some $\tilde{\mu} \in \tilde{\Pi}_h$ and some $x_0 \in S$. The partition that contains $x_0$ is $\sigma_{x_0}$ and its representative is $\breve{\sigma}_{x_0}$. Let $\tilde{u}_0 = \tilde{\mu}(x_0) = \tilde{\mu}(\breve{\sigma}_{x_0})$, where the second equality holds because $\tilde{\mu}$ is constant on the set $\sigma_{x_0}$. By the definition of $\tilde{\Pi}_h$, there exists some $u_0 \in U(\breve{\sigma}_{x_0})$ such that $\|u_0 - \tilde{u}_0\|_\infty \le h/2$.

Let $G \overset{\text{def}}{=} \{u \in C \mid \|u - \tilde{u}_0\|_\infty \le (K + 1/2)h\}$. By Assumption A.3, $G \cap U(x)$ is nonempty for all $x \in \sigma_{x_0}$. Thus, for every $x \in \sigma_{x_0}$, we can choose some $\mu(x) \in U(x)$ such that $\|\mu(x) - \tilde{\mu}(x)\|_\infty = \|\mu(x) - \tilde{u}_0\|_\infty \le (K + 1/2)h$. By repeating this argument for each set in the partition of $S$ we obtain a function $\mu$ that satisfies the desired inequality. There is one final issue that has to be dealt with: according to the definition of $\Pi$, $\mu$ must be a measurable function. This can be accomplished by appealing to the measurable selection theorem (Proposition 2.2.3). □

### 2.4.2 Discretization and Ergodicity Conditions

We now show that an ergodicity condition in the continuous-problem is "inherited" by the discretized problem for a sufficiently fine discretization. Note that for the discretized transition kernel $\tilde{q}_h$, we use the same definitions of ergodicity as in Section 2.3, except that we replace $\Gamma$ by $\tilde{\Gamma}_h$ and $\Pi$ by $\tilde{\Pi}_h$ (in Assumptions E.1–E.6).

Before proving the main result, we prove the following lemma:

**Lemma 2.4.2** *(Under Assumption D.2) For any $h \in (0, h_a]$ and $\tilde{\mu} \in \tilde{\Pi}_h$, there exists a $\mu \in \Pi$ such that*

$$\|\tilde{q}_{\tilde{\mu}} - q_\mu\|_\infty \le K_q' h,$$

*where $K_q' = K + K(K + 1/2) + K_q$.*

**Proof** Fix some $h \in (0, h_a]$ and some $\tilde{\mu} \in \tilde{\Pi}_h$. By Lemma 2.4.1, there exists a $\mu$ such that

$$\|\tilde{\mu} - \mu\|_\infty \le (K + 1/2)h. \tag{2.4.9}$$

Fix some $x \in S$. Using the triangle inequality, we have

$$\begin{aligned}
\|\tilde{q}_{\tilde{\mu}}(\cdot \mid x) - q_\mu(\cdot \mid x)\|_{\text{v}} &= \|\tilde{q}_h(\cdot \mid x, \tilde{\mu}(x)) - q(\cdot \mid x, \mu(x))\|_{\text{v}} \\
&\le \|\tilde{q}_h(\cdot \mid x, \tilde{\mu}(x)) - q(\cdot \mid x, \tilde{\mu}(x))\|_{\text{v}} \\
&\qquad + \|q(\cdot \mid x, \tilde{\mu}(x)) - q(\cdot \mid x, \mu(x))\|_{\text{v}} \\
&\le (K + K_q)h + K(K + 1/2)h \\
&= K_q' h,
\end{aligned}$$

where the last inequality follows from Eq. (2.4.6), Assumptions A.2, and Eq. (2.4.9). Since $x$ is arbitrary, it follows that

$$\|\tilde{q}_{\tilde{\mu}} - q_{\mu}\|_{\infty} \leq K'_q h,$$

as required. □

For the main result in this subsection, we are only interested in the case where $q$ is a stochastic kernel. (Note that it follows from Assumption D.1 that $\tilde{q}_h$ is also a stochastic kernel.) The theorem is as follows.

**Theorem 2.4.1** *(Under Assumption A.4, D.1, and D.2.) If $q$ satisfies Assumption E.3 (respectively, E.4) with ergodicity rate $2\rho$ then there exists an $h_b \in (0, h_a]$ such that for all $h \in (0, h_b]$, $\tilde{q}_h$ satisfies Assumption E.3 (respectively, E.4) with ergodicity rate $\rho$.*

**Proof** We will prove the result for Assumption E.4, the proof for the other case is similar.

Fix an $h \in (0, h_a]$. Let $\tilde{\mu}_0, \tilde{\mu}_1, \ldots, \tilde{\mu}_k \in \tilde{\Pi}_h$. By Lemma 2.4.2, there exist policies $\mu_0, \mu_1, \ldots, \mu_{k-1}, \mu_k \in \Pi$ such that

$$\|\tilde{q}_{\tilde{\mu}_i} - q_{\mu_i}\|_{\infty} \leq K'_q h, \quad \forall i = 0, 1, \ldots, k. \tag{2.4.10}$$

We now prove by induction that

$$\|\tilde{q}_{\tilde{\mu}_0} \cdots \tilde{q}_{\tilde{\mu}_i} - q_{\mu_0} \cdots q_{\mu_i}\|_{\infty} \leq (i+1)K'_q h, \quad \forall i = 0, \ldots, k-1. \tag{2.4.11}$$

When $i = 0$, the result follows from Eq. (2.4.10). We now assume that (2.4.11) is true for $i$ and will show it for $i+1$. By the triangle inequality,

$$\begin{aligned}\|\tilde{q}_{\tilde{\mu}_0} \cdots \tilde{q}_{\tilde{\mu}_{i+1}} - q_{\mu_0} \cdots q_{\mu_{i+1}}\|_{\infty} &\leq \|(\tilde{q}_{\tilde{\mu}_0} \cdots \tilde{q}_{\tilde{\mu}_i} - q_{\mu_0} \cdots q_{\mu_i})\tilde{q}_{\tilde{\mu}_{i+1}}\|_{\infty} \\ &\quad + \|q_{\mu_0} \cdots q_{\mu_i}(\tilde{q}_{\tilde{\mu}_{i+1}} - q_{\mu_{i+1}})\|_{\infty} \\ &\leq iK'_q h + K'_q h \\ &\leq (i+1)K'_q h,\end{aligned}$$

as required.

Using Eq. (2.4.11) and the triangle inequality, we have

$$\begin{aligned}\|\tilde{q}_{\tilde{\mu}_0} \cdots \tilde{q}_{\tilde{\mu}_{k-1}} - \tilde{q}_{\tilde{\mu}_1} \cdots \tilde{q}_{\tilde{\mu}_k}\|_{\infty} &\leq \|\tilde{q}_{\tilde{\mu}_0} \cdots \tilde{q}_{\mu_{k-1}} - q_{\mu_0} \cdots q_{\mu_{k-1}}\|_{\infty} \\ &\quad + \|q_{\mu_0} \cdots q_{\mu_{k-1}} - q_{\mu_1} \cdots q_{\mu_k}\|_{\infty} \\ &\quad + \|q_{\mu_1} \cdots q_{\mu_k} - \tilde{q}_{\tilde{\mu}_1} \cdots \tilde{q}_{\tilde{\mu}_k}\|_{\infty} \\ &\leq 2(1 - 2\rho) + 2kK'_q h \\ &= 2[1 - (2\rho - kK'_q h)].\end{aligned}$$

By letting $h_b \overset{\text{def}}{=} h_a \wedge \rho/(kK'_q)$, we see that the discretized stochastic kernel $\tilde{q}_h$ satisfies Assumption E.4 with ergodicity rate $\rho$ for all $h \in (0, h_b]$, as required. □

### 2.4.3 Discretization Error Bounds

An outline of this subsection is as follows. We first prove some useful lemmas. We then prove for the general case (where Assumption A.4″ is in effect) a theorem which bounds the error between the optimal cost functions of the original MDP and the discretized MDP. Finally, we specialize the theorem when Assumption A.4 is in effect and show that the discretization error is reduced if the dynamics satisfy certain ergodicity conditions.

We now prove some useful lemmas.

**Lemma 2.4.3** *Let $T_1$ be a $k$-stage contraction operator, with contraction factor $\alpha$, on a Banach space $(X, \|\cdot\|)$. If $J_1 \in X$ is the fixed point of $T_1$, then*

$$\|J_1 - J\| \leq \frac{k}{1-\alpha}\|T_1 J - J\|, \quad \forall J \in X.$$

**Proof** Using the triangle inequality, we have

$$\begin{aligned}\|T_1^k J_1 - J\| &\leq \sum_{i=1}^{k} \|T_1^{k+1-i} J - T_1^{k-i} J\| \\ &\leq k\|T_1 J - J\|.\end{aligned}$$

Now using the triangle inequality and the contraction property of $T_1$, we have

$$\begin{aligned}\|J_1 - J\| &\leq \|T_1^k J_1 - T_1^k J\| + \|T_1^k J - J\| \\ &\leq \alpha\|J_1 - J\| + k\|T_1 J - J\|,\end{aligned}$$

from which the result follows. □

**Lemma 2.4.4** *(Under Assumption A.4″) There holds*

$$\|J^*\|_\infty \leq K/(1-\alpha).$$

**Proof** Let $J^0(x) = 0$ for all $x \in S$. It is clear from Assumption A.1 that $\|TJ^0\|_\infty \leq K$. Therefore,

$$\begin{aligned}\|J^*\|_\infty &= \|J^* - J^0\|_\infty \\ &\leq \|TJ^* - TJ^0\|_\infty + \|TJ^0 - J^0\|_\infty \\ &\leq \alpha\|J^*\|_\infty + K;\end{aligned}$$

the result follows. □

We now show a similar result under Assumption A.4 and an ergodicity condition.

**Lemma 2.4.5** *(Under Assumption A.4) If the dynamics described by $q$ satisfy a $k$-stage ergodicity condition with ergodicity rate $\rho$ (Assumption E.3), then*

$$\|J^*\|_s \leq 2kK/\rho.$$

**Proof** Let $J^0(x) = 0$ for all $x \in S$. It follows that $\|TJ^0\|_s \leq 2\|TJ^0\|_\infty \leq 2K$. Now,

$$\begin{aligned}\|T^k J^0\|_s &\leq \sum_{i=1}^{k} \|T^{k+1-i}J^0 - T^{k-i}J^0\|_s \\ &\leq \sum_{i=1}^{k} \|TJ^0 - J^0\|_s \\ &\leq 2kK.\end{aligned}$$

Therefore, by Theorem 2.3.2(a),

$$\begin{aligned}\|J^*\|_s &\leq \|T^k J^* - T^k J^0\|_s + \|T^k J^0 - J^0\|_s \\ &\leq (1-\rho)\|J^*\|_s + 2kK;\end{aligned}$$

the result follows. □

We now prove the main theorem for the general case.

**Theorem 2.4.2** *(Under Assumption A.4″, D.1, and D.2) There exist constants $K_1$ and $K_2$ (depending only on the constants $K$ and $K_q$) such that for all $h \in (0, h_a]$ and all $J \in \mathcal{B}(S)$,*

$$\|TJ - \tilde{T}_h J\|_\infty \leq (K_1 + \alpha K_2 \|J\|_\infty)h; \tag{2.4.12}$$

$$\|J^* - \tilde{J}_h^*\|_\infty \leq \frac{1}{1-\alpha}(K_1 + \alpha K_2 \|J^*\|_\infty)h. \tag{2.4.13}$$

*Furthermore,*

$$\|J^* - \tilde{J}_h^*\|_\infty \leq \frac{K'h}{(1-\alpha)^2}, \tag{2.4.14}$$

*where $K' = K_1 + KK_2$.*

**Proof** Fix some $J \in \mathcal{B}(S)$ and some $x \in S$. We define

$$\begin{aligned}H(u) &= g(x,u) + \alpha \int_S q(dy\,|x,u)J(y), \quad u \in C; \\ \tilde{H}_h(\tilde{u}) &= \tilde{g}_h(x,\tilde{u}) + \alpha \int_S \tilde{q}_h(dy\,|x,\tilde{u})J(y), \quad \tilde{u} \in \tilde{C}_h.\end{aligned}$$

Let $v \in U(x)$ be such that $H(v) = \min_{u \in U(x)} H(u)$. Using Assumption A.3, there exists some $v' \in U(\breve{\sigma}_x)$ such that $\|v - v'\|_\infty \leq K\|x - \breve{\sigma}_x\|_\infty \leq Kh$. Finally, choose some $\tilde{v} \in \tilde{U}_h(\breve{\sigma}_x) = \tilde{U}_h(x)$ such that $\|v' - \tilde{v}\|_\infty \leq h/2$. [This is possible because of the way that $\tilde{U}_h(\breve{\sigma}_x)$ is defined.] By construction,

$$\|v - \tilde{v}\|_\infty \leq \left(K + \frac{1}{2}\right)h. \tag{2.4.15}$$

We now have

$$
\begin{aligned}
\tilde{T}_h J(x) - TJ(x) &= \min_{\tilde{u}\in\tilde{U}_h(x)} \tilde{H}_h(\tilde{u}) - \min_{u\in U(x)} H(u) \\
&\le \left|\tilde{H}_h(\tilde{v}) - H(v)\right| \\
&\le \left|\tilde{H}_h(\tilde{v}) - H(\tilde{v})\right| + |H(\tilde{v}) - H(v)|. \qquad (2.4.16)
\end{aligned}
$$

We bound the two terms in the right-hand side of Eq. (2.4.16). For the first term,

$$
\begin{aligned}
\left|\tilde{H}_h(\tilde{v}) - H(\tilde{v})\right| &\le |\tilde{g}_h(x,\tilde{v}) - g(x,\tilde{v})| + \alpha\|\tilde{q}_h(\cdot \mid x,\tilde{v}) - q(\cdot \mid x,\tilde{v})\|_{\mathrm{v}} \cdot \|J\|_\infty \\
&\le K\|x - \tilde{\sigma}_x\|_\infty + \alpha K_q h\|J\|_\infty \\
&\le (K + \alpha K_q\|J\|_\infty)h, \qquad (2.4.17)
\end{aligned}
$$

where $K_q$ is the constant of Assumption D.2 (see Section 2.4.1). And for the second term,

$$
\begin{aligned}
|H(\tilde{v}) - H(v)| &\le |g(x,\tilde{v}) - g(x,v)| + \alpha\|q(\cdot\,|x,\tilde{v}) - q(\cdot\,|x,v)\|_{\mathrm{v}} \cdot \|J\|_\infty \\
&\le K\|\tilde{v} - v\|_\infty + \alpha K\|\tilde{v} - v\|_\infty \cdot \|J\|_\infty \\
&\le (K + \alpha K\|J\|_\infty)\bigl(K + \tfrac{1}{2}\bigr)h. \qquad (2.4.18)
\end{aligned}
$$

Using the bounds of (2.4.17) and (2.4.18) in Eq. (2.4.16), we obtain

$$
\tilde{T}_h J(x) - TJ(x) \le (K_1 + \alpha K_2\|J\|_\infty)h,
$$

where $K_1$ and $K_2$ are suitable constants. By a symmetrical argument, we obtain the same bound for $TJ(x) - \tilde{T}_h J(x)$; thus,

$$
\left|\tilde{T}_h J(x) - TJ(x)\right| \le (K_1 + \alpha K_2\|J\|_\infty)h.
$$

Taking the supremum over all $x \in S$, we obtain

$$
\|TJ - \tilde{T}_h J\|_\infty \le (K_1 + \alpha K_2\|J\|_\infty)h, \qquad (2.4.19)
$$

which is the desired result. We now use Lemma 2.4.3 (with $k = 1$, $T_1 = \tilde{T}_h$, $J_1 = \tilde{J}_h^*$, $J = J^*$) and Eq. (2.4.19) to obtain

$$
\|J^* - \tilde{J}_h^*\|_\infty \le \frac{1}{1-\alpha}\|\tilde{T}_h J^* - TJ^*\|_\infty \le \frac{1}{1-\alpha}(K_1 + \alpha K_2\|J^*\|_\infty)h,
$$

which proves the first part of the theorem.

The second part follows trivially from Lemma 2.4.4, where we have $\|J^*\|_\infty \le K/(1-\alpha)$, and therefore

$$
\|J^* - \tilde{J}_h^*\|_\infty \le \frac{K'h}{(1-\alpha)^2},
$$

where $K' = K_1 + KK_2$.

□

We now specialize the result under Assumption A.4. We also show that if the dynamics satisfy a certain ergodicity condition, the discretization error bound can be improved.

**Theorem 2.4.3** *Let $K_1$ and $K_2$ be the constants of Theorem 2.4.1. For all $h \in (0, h_a]$ and all $J \in \mathcal{B}(S)$, there hold,*

$$\|TJ - \tilde{T}_h J\|_\infty \leq (K_1 + \frac{\alpha}{2} K_2 \|J\|_s)h; \tag{2.4.20}$$

$$\|J^* - \tilde{J}_h^*\|_\infty \leq \frac{1}{1-\alpha}(K_1 + \frac{\alpha}{2} K_2 \|J^*\|_s)h. \tag{2.4.21}$$

*Furthermore, if the dynamics satisfy Assumption E.3 (a k-stage ergodicity condition with ergodicity rate $\rho$) then*

$$\|J^* - \tilde{J}_h^*\|_\infty \leq \frac{K''h}{(1-\alpha)}, \tag{2.4.22}$$

*where $K'' = K_1 + kKK_2/\rho$*

**Proof** Let $c = -\frac{1}{2}[\sup_{x \in S} J(x) + \inf_{x \in S} J(x)]$. Since $T$ and $\tilde{T}_h$ satisfies Eq. (2.2.7), we have

$$\begin{aligned}\|TJ - \tilde{T}_h J\|_\infty &= \|T(J + c) - \tilde{T}_h(J + c)\|_\infty \\ &\leq (K_1 + \alpha K_2 \|J + c\|_\infty)h,\end{aligned}$$

where the last step uses Eq. (2.4.12). It is easily seen that $\|J + c\|_\infty = \|J\|_s/2$, and we obtain

$$\|TJ - \tilde{T}_h J\|_\infty \leq (K_1 + \frac{\alpha}{2} K_2 \|J\|_s)h.$$

Using Lemma 2.4.2 (as in the proof of Theorem 2.4.2), we obtain

$$\|J^* - \tilde{J}_h^*\|_\infty \leq \frac{1}{1-\alpha}(K_1 + \frac{\alpha}{2} K_2 \|J^*\|_s)h.$$

The second part follows from Lemma 2.4.5(b), where we have $\|J^*\|_s \leq 2kK/\rho$ and therefore

$$\|J^* - \tilde{J}_h^*\|_\infty \leq \frac{K''h}{(1-\alpha)},$$

where $K'' = K_1 + kKK_2/\rho$. □

Note that the bounds of Theorem 2.4.3 hold under Assumption E.4 or E.5, provided $K''$ is replaced by a suitable constant (independent of $\alpha$).

### 2.4.4 Notes

In an approximation scheme we use the same state and control spaces and only the cost function and the dynamics are approximated. In contrast, in a discretization scheme, both the state and control spaces are approximated by discrete (finite) sets; the cost function and the dynamics are also discretized appropriately. Our discretization scheme is somehow in between—we keep the same state space but discretize the control space, the cost function, and the dynamics.

Our discretization procedures are similar to the discretization procedures in Whitt (1978). Whitt considers an abstract framework for discretizing MDP's, and obtain similar bounds to Theorems 2.4.2 and 2.4.3. But a key difference is that our discretized problems are defined on the same state space $S$ (unlike Whitt) and all of the operators $\tilde{T}_h$ act on the same function space $\mathcal{B}(S)$. This greatly facilitates the grid–level changes in the multigrid algorithms to be introduced later. For example, in our framework, two iterations on different grids correspond to the application of an operator of the form $\tilde{T}_h\tilde{T}_{h'}$. In contrast, in the framework of Whitt, a grid-level change requires the application of certain interpolation and projection operators. Whitt does not study the effect of ergodicity conditions on the discretization error in his paper.

Our discretization scheme is similar to an approximation scheme in Hernández-Lerma (1989), except that we have discretized the control space. Our discretization bounds are similar to the bounds in Hernández-Lerma. Hernández-Lerma studies ergodicity conditions in the context of average-cost problems but does not study the effect on the discretization error bound for discounted-cost problems.

Note that a discretized policy may lie outside the set $\Pi$ of admissible policies for the original problem. This is why we require $g$ and $q$ to be Lipschitz continuous outside of $\Gamma$ (cf. Assumptions A.1–A.4).

## 2.5 Discretization of Densities

We now specialize the results of Sections 2.3 and 2.4 to the case where the transition kernel $q$ is a density. That is we assume that for each $(x,u) \in S \times C$, we are given a bounded, measurable function $P(\cdot \mid x,u) : S \mapsto \mathbf{R}$ such that

$$q(B \mid x,u) \overset{\text{def}}{=} \int_B P(y \mid x,u)\,dy, \qquad \forall B \in \mathcal{B}_S. \tag{2.5.1}$$

Note that for any $x, x' \in S$ and $u, u \in C$,

$$\|q(\cdot \mid x,u) - q(\cdot \mid x',u')\|_{\mathrm{V}} = \int_S |P(y \mid x,u) - P(y \mid x',u')|\,dy$$

and that $0 \le q(\cdot \mid x,u)$ is equivalent to $0 \le P(y \mid x,u)$ for all $y \in S$. So, when $q$ is a density the assumptions in the previous sections should be modified appropriately.

In this section we consider the discretization of $P$ when it satisfies additional Lipschitz continuity assumptions. We show how to construct a discretized transition density $\tilde{P}_h$ from $P$, so that the discretization conditions D.1 and D.2 are satisfied;

we also obtain explicit bounds on the constants $h_a$ and $K_q$ of these discretization conditions. (See Section 2.4.1.)

An outline of this section is as follows. First, we introduce the notion of "piecewise Lipschitz continuous functions" and impose additional assumptions on the dynamics. Second, we discuss the discretization of general densities (Assumption A.4″) and ensure that Assumptions D.1 and D.2 are satisfied. Third, we specialize the result to probability densities (Assumption A.4). Finally, we make some observations.

## 2.5.1 Piecewise Lipschitz Continuous Functions

We introduce the notion of piecewise Lipschitz continuous functions. We begin by introducing more notation. Let $X \subset \mathbf{R}^n$. The *interior* of $X$ is the union of all open subsets of $\mathbf{R}^n$ contained in $X$; it is denoted by $X^\circ$. The *closure* of $X$ is the intersection of all closed subsets of $\mathbf{R}^n$ containing $X$; it is denoted by $\overline{X}$. The *boundary* of $X$ is denoted and given by

$$\partial X \overset{\text{def}}{=} \left\{ x \in \overline{X} \;\middle|\; x \notin X^\circ \right\}.$$

Furthermore, for any $h > 0$,

$$\partial X^{(h)} \overset{\text{def}}{=} \left\{ x \in \mathbf{R}^n \;\middle|\; \|x - x'\|_\infty < h \text{ for some } x' \in \partial X \right\}.$$

Finally, let $\lambda_n(\cdot)$ denote the $n$-dimensional Borel measure; that is, $\lambda_n(B) = \int_B dy$. A function $f : S \mapsto \mathbf{R}$ is called a *piecewise Lipschitz continuous function on $S$ with Lipschitz constant $K_l$* if there exists a finite collection of measurable subsets $\{S_i\}_{i \in I}$ of $S$ such that $S = \bigcup_{i \in I} S_i$ and that

$$|f(y) - f(y')| \le K_l \|y - y'\|_\infty, \quad \forall y, y' \in S_i, i \in I; \tag{2.5.2}$$

$$\lambda_n \left( S \cap \bigcup_{i \in I} \partial S_i^{(h)} \right) \le K_l h, \quad \forall h \in (0, 1], \tag{2.5.3}$$

Equation (2.5.2) says that $f$ is Lipschitz continuous on each set $S_i$ in the partition. And Eq. (2.5.3) is a smoothness condition on the boundaries of the partition $\{S_i\}$; in particular, this assumption is satisfied when the boundaries $\partial S_i$ are given by smooth curves. This condition implies that $\lambda_n(\bigcup_{i \in I} \partial S_i) = 0$; but the latter is not sufficient for our purpose, because we need explicit bounds on $\lambda_n \left( S \cap \bigcup_{i \in I} \partial S_i^{(h)} \right)$ for our discretization estimates.

### Additional Assumptions

We now introduce the following additional assumption on the transition density $P$.

**A.5** For each $y, x \in S$ and $u \in C$, $|P(y \mid x, u)| \le K$ and $P(\cdot \mid x, u)$ is a piecewise Lipschitz continuous function on $S$ with Lipschitz constant $K$.

Assumption A.5 is a locally Lipschitz continuous assumption; it ensures that for each $(x,u)$, $P(\cdot \mid x,u)$ can be accurately approximated, when the discretization is sufficiently fine. [Note that the partitions on which $P(\cdot \mid x,u)$ is Lipschitz continuous may vary with $(x,u)$.] Since $P$ may have discontinuities, Assumption A.5 also bounds the magnitudes of these discontinuities (by $K$).

From now on, we will always assume that the dynamics are described by the density $P$ and that Assumption A.5 is always in effect.

### 2.5.2 Discretizing Piecewise Lipschitz Continuous Densities

We now consider the discretization of the transition density $P$ for the general case where Assumption A.4″ is in effect. Let $h_a = 1$ and fix some grid-size $h \in (0, h_a]$. We discretize the state and control spaces as in Section 2.4.1 and construct the discretized density $\tilde{P}_h$. We will check that $\tilde{P}_h$ satisfies the discretization conditions D.1 and D.2.

Fix some $(\tilde{x}, \tilde{u}) \in \tilde{S}_h \times \tilde{C}_h$. For conciseness, we write $P(\cdot)$ and $\tilde{P}_h(\cdot)$ instead of $P(\cdot \mid \tilde{x}, \tilde{u})$ and $\tilde{P}_h(\cdot \mid \tilde{x}, \tilde{u})$, respectively. Recall that for any $y \in S$, $\sigma_y$ denotes the partition of $S$ that contains $y$ and $\breve{\sigma}_y$ denotes the representative of $\sigma_y$. We let

$$\tilde{P}_h(y) \overset{\text{def}}{=} \begin{cases} P(\breve{\sigma}_y) & \text{if } \int_S |P(\breve{\sigma}_y)|\, dy \le 1; \\ P(\breve{\sigma}_y) \big/ \int_S |P(\breve{\sigma}_y)|\, dy & \text{otherwise.} \end{cases} \tag{2.5.4}$$

[Note that $\tilde{P}_h(\cdot)$ is defined from the values of $P(\cdot)$, sampled at the representatives.]

We check that $\tilde{P}_h$ satisfies the discretization conditions. It is clear from Eq. (2.5.4) that $\int_S \left|\tilde{P}_h(y)\right| dy \le 1$. Therefore, $\tilde{P}_h$ satisfies Assumption D.1. It remains to verify that $\tilde{P}_h$ satisfies Assumption D.2, which we will show in the following lemma.

**Lemma 2.5.1** *(Under Assumption A.4″) Let $h_a = 1$ and $K_q = 2(K + 2K^2)$. If $\tilde{P}_h$ is given by Eq. (2.5.4) then*

$$\int_S \left|P(y) - \tilde{P}_h(y)\right| dy \le K_q h, \qquad \forall h \in (0, h_a].$$

*Thus, $\tilde{P}_h$ satisfies Assumption D.2.*

**Proof** Fix some $h \in (0, h_a]$. Using Assumptions A.2 and A.5, we have

$$|P(y) - P(\breve{\sigma}_y)| \le K \|y - \breve{\sigma}_y\|_\infty \le Kh, \quad \forall y \in S, \tag{2.5.5}$$

except for a set of measure $Kh$, where $|P(y) - P(\breve{\sigma}_y)| \le 2K$. Therefore, we have the following useful bound

$$\begin{aligned} \int_S |P(y) - P(\breve{\sigma}_y)|\, dy &\le Kh \cdot \int_S dy + 2K \cdot Kh \\ &\le (K + 2K^2)h. \end{aligned} \tag{2.5.6}$$

We consider two cases: (i) $\int_S |P(\breve{\sigma}_y)|\,dy > 1$, and (ii) $\int_S |P(\breve{\sigma}_y)|\,dy \le 1$. For Case (i), we have

$$\begin{aligned}\int_S |P(\breve{\sigma}_y)|\,dy - 1 &\le \int_S |P(\breve{\sigma}_y)|\,dy - \int_S |P(y)|\,dy \\ &\le \int_S |P(\breve{\sigma}_y) - P(y)|\,dy \\ &\le (K + 2K^2)h, \end{aligned} \tag{2.5.7}$$

where the last inequality follows from Eq. (2.5.6). By the definition of $\tilde{P}_h$ [Eq. (2.5.4)], we have

$$\begin{aligned}\int_S \left|\tilde{P}_h(y) - P(y)\right| dy &= \int_S \frac{|P(\breve{\sigma}_z) - P(z)\cdot \int_S |P(\breve{\sigma}_y)|\,dy|}{\int_S |P(\breve{\sigma}_y)|\,dy}\,dz \\ &\le \int_S |P(\breve{\sigma}_y) - P(y)|\,dy + \int_S |P(y)|\,dy \cdot \left|\int_S |P(\breve{\sigma}_y)|\,dy - 1\right| \\ &\le \int_S |P(\breve{\sigma}_y) - P(y)|\,dy + \int_S |P(\breve{\sigma}_y)|\,dy - 1 \\ &\le 2(K + 2K^2)h, \end{aligned} \tag{2.5.8}$$

where the last inequality follows from Eq. (2.5.6) and (2.5.7). The result is within the desired bound.

We consider Case (ii), where $\int_S |P(\breve{\sigma}_y)|\,dy \le 1$. It follows from Eq. (2.5.4) that

$$\begin{aligned}\int_S \left|\tilde{P}_h(y) - P(y)\right| dy &= \int_S |P(\breve{\sigma}_y) - P(y)|\,dy \\ &\le (K + 2K^2)h, \end{aligned}$$

where the last inequality follows from Eq. (2.5.6). Again the result is within the desired bound.

□

Therefore, we have constructed $\tilde{P}_h$ from the samples of $P$ and shown that $\tilde{P}_h$ satisfies the discretization conditions D.1 and D.2. The same discretization procedure applies to the case when $P$ is a substochastic kernel (Assumption A.4′) and the resultant $\tilde{P}_h$ also satisfies D.1 and D.2.

### 2.5.3 Discretization of Probability Densities

We now consider the discretization of the transition density $P$ when $P$ is a probability density (Assumption A.4). We proceed as in the general case, except that we let $h_a = 1/(2K + 4K^2)$ instead of 1. We fix some grid-size $h \in (0, h_a]$ and some $(\tilde{x}, \tilde{u}) \in \tilde{S}_h \times \tilde{C}_h$. The discretized density is now given by

$$\tilde{P}_h(y) \overset{\text{def}}{=} P(\breve{\sigma}_y) \Big/ \int_S |P(\breve{\sigma}_y)|\,dy \tag{2.5.9}$$

We verify that $\tilde{P}_h$ is well-defined by checking that the denominator in Eq. (2.5.9) is non-zero. We note that Eq. (2.5.6) applies, and it follows that

$$\begin{aligned}\int_S P(\breve{\sigma}_y)\,dy &\geq \int_S P(y)\,dy - (K+2K^2)h \\ &\geq 1-(K+2K^2)h \\ &\geq 1/2 \end{aligned} \tag{2.5.10}$$

verifying that the denominator of Eq. (2.5.9) is indeed non-zero.

We check that $\tilde{P}_h$ satisfies the discretization conditions D.1 and D.2. Assumption D.1 requires $\tilde{P}_h(\cdot)$ to be a probability density on $S$; but that is clear from Eq. (2.5.9). It remains to verify that $\tilde{P}_h$ satisfies Assumption D.2. We show this in the following lemma.

**Lemma 2.5.2** *(Under Assumption A.4) Let* $h_a = 1/(2K+4K^2)$ *and* $K_q = 4(K+K^2)$. *If* $\tilde{P}_h$ *is given by Eq. (2.5.9) then*

$$\int_S \left|P(y)-\tilde{P}_h(y)\right| dy \leq K_q h, \qquad \forall h \in (0,h_a].$$

**Proof** We consider two cases: (i) $\int_S P(\breve{\sigma}_y)\,dy > 1$, and (ii) $\int_S P(\breve{\sigma}_y)\,dy \leq 1$.

For Case (i), where $\int_S P(\breve{\sigma}_y)\,dy > 1$, the discretized density $\tilde{P}_h$ is identical to the general case and therefore the bound of Lemma 2.5.1 applies. Hence, $K_q$ is within the desired bound.

It remains to consider Case (ii), where $\int_S P(\breve{\sigma}_y)\,dy \leq 1$. Using Eqs. (2.5.9) and (2.5.10), we have

$$\begin{aligned}\int_S \left|\tilde{P}_h(y)-P(y)\right| dy &= \int_S \frac{|P(\breve{\sigma}_z)-P(z)\cdot\int_S P(\breve{\sigma}_y)\,dy|}{\int_S P(\breve{\sigma}_y)\,dy}\,dz \\ &\leq \frac{\int_S |P(\breve{\sigma}_y)-P(y)|\,dy + \int_S P(y)\,dy \cdot |\int_S P(\breve{\sigma}_y)\,dy - 1|}{1/2} \\ &\leq 2\left[(K+2K^2)h+(K+2K^2)h\right] \\ &= 4(K+2K^2)h, \end{aligned}$$

where the last inequality follows from Eqs. (2.5.6) and (2.5.7) and the fact that $\int_S P(y)\,dy = 1$. □

Thus, when we discretize a probability density according to Eq. (2.5.9), we obtain a probability density that satisfies Assumptions D.1 and D.2.

## 2.5.4 Notes

The normalization in Eqs. (2.5.4) and (2.5.9) is needed to ensure that Assumption D.1 is satisfied, which in turn ensures that $\tilde{T}_h$ has the same contraction and monotone properties as $T$. The normalization is important in our analysis because we study the dependence on $\alpha$ as $\alpha \uparrow 1$.

The discretization results in this section are new. The discretization bounds can be extended to the case where the densities have "impulses" at fixed known locations and provided Assumption A.2 is satisfied.

## 2.6 Summary

In this chapter we have formulated a discounted-cost Markov Decision Process (MDP) and discussed some ergodicity conditions. We show that certain ergodicity conditions result in an extra span-norm contraction factor in the dynamic programming operator. We also study the discretization of the MDP and show that certain ergodicity conditions improve the discretization error bounds. Finally, in Section 2.5, we specialize our results to the case where the dynamics are described by a piecewise Lipschitz continuous density.

From now on, we are interested in the above results only when the dynamics are specified by a density; thus, an MDP is specified by $(S, C, \{U(x)\}, P, g, \alpha)$. For easy reference, we restate all the assumptions that we will use in the next chapter, in terms of the density $P$ (instead of the kernel $q$).

### Assumptions

**A.1** For all $x, x' \in S$ and $u, u' \in C$, $|g(x,u)| \leq K$ and $|g(x,u) - g(x',u')| \leq K\|(x,u) - (x',u')\|_\infty$.

**A.2** For all $x, x' \in S$ and $u, u' \in C$, $\int_S |P(y \mid x,u) - P(y \mid x',u')| \, dy \leq K\|(x,u) - (x',u')\|_\infty$.

**A.3** For any $x, x' \in S$ and any $u' \in U(x')$, there exists some $u \in U(x)$ such that $\|u - u'\|_\infty \leq K\|x - x'\|_\infty$.

**A.4** For all $y, x \in S$ and $u \in C$, $0 \leq P(y \mid x,u)$ and $\int_S P(y \mid x,u)\, dy = 1$.

**A.4$'$** For all $y, x \in S$ and $u \in C$, $0 \leq P(y \mid x,u)$ and $\int_S P(y \mid x,u)\, dy \leq 1$.

**A.4$''$** For all $x \in S$ and $u \in C$, $\int_S |P(y \mid x,u)|\, dy \leq 1$.

**A.5** For each $y, x \in S$ and $u \in C$, $|P(y \mid x,u)| \leq K$ and $P(\cdot \mid x,u)$ is a piecewise Lipschitz continuous function on $S$ with Lipschitz constant $K$.

Assumptions A.1–A.3 and A.5 will always be in effect; by default Assumption A.4 is also in effect, unless Assumption A.4$'$ or A.4$''$ is in effect. Moreover, $K$ will always denote the constant of these assumptions.

**Ergodicity Conditions**

Two ergodicity conditions we will use are:

**E.1** There exist a scalar $\rho \in (0,1]$ and a non-negative measurable function $r$ with $\int_S r(y)\,dy \geq \rho$ such that $P(y \mid x,u) \geq r(y), \quad \forall y \in S, (x,u) \in \Gamma$.

**E.3** There exist an integer $k \geq 1$ and a scalar $\rho \in (0,1]$ such that for any policies $\mu_0, \mu_1, \ldots \mu_{k-1}, \mu'_0, \mu'_1, \ldots, \mu'_{k-1} \in \Pi$, there holds

$$\int_S \left| P_{\mu_0} P_{\mu_1} \cdots P_{\mu_{k-1}}(y \mid x) - P_{\mu'_0} P_{\mu'_1} \cdots P_{\mu'_{k-1}}(y \mid x') \right| dy \leq 2(1-\rho), \quad \forall x, x' \in S.$$

Assumption E.1 is called a 1-stage ergodicity condition; Assumption E.3 is called a $k$-stage ergodicity condition. Moreover, $\rho$ will always denote the ergodicity rate of these assumptions, and $k$ will denote the constant of Assumption E.3.

**Discretization of the Dynamics**

We now summarize the discretization of the transition density $P$. Under Assumption A.4″ or A.4′, the constants of the discretization conditions D.1 and D.2 (see Section 2.4.1), $h_a$ and $K_q$, are given by $h_a = 1$ and $K_q = 2(K + 2K^2)$ (cf. Lemma 2.5.1). And for all $y, x \in S$, $\tilde{u} \in \tilde{C}_h$, and $h \in (0, h_a]$, the discretized density is given by

$$\tilde{P}_h(y \mid x, \tilde{u}) \overset{\text{def}}{=} \begin{cases} P(\check{\sigma}_y \mid \check{\sigma}_x, \tilde{u}) & \text{if } \int_S |P(\check{\sigma}_y \mid \check{\sigma}_x, \tilde{u})|\,dy \leq 1; \\ P(\check{\sigma}_y \mid \check{\sigma}_x, \tilde{u}) \big/ \int_S |P(\check{\sigma}_y \mid \check{\sigma}_x, \tilde{u})|\,dy & \text{otherwise.} \end{cases}$$

Under Assumption A.4, we have $h_a = 1/(2K + 4K^2)$ and $K_q = 4(K + K^2)$ (cf. Lemma 2.5.2). And for all $y, x \in S$, $\tilde{u} \in \tilde{C}_h$, and $h \in (0, h_a]$, the discretized density is given by

$$\tilde{P}_h(y \mid x, \tilde{u}) \overset{\text{def}}{=} P(\check{\sigma}_y \mid \check{\sigma}_x, \tilde{u}) \Big/ \int_S |P(\check{\sigma}_y \mid \check{\sigma}_x, \tilde{u})|\,dy.$$

**Other Constants**

Some other constants we will encounter are: (i) the constants $K_1$, $K_2$, and $K'$ of Theorem 2.4.2, all of which depend only on $K$, (ii) the constant $K''$ of Theorem 2.4.3, which depends on $K$, $k$, and $\rho$.

# Chapter 3

# The Complexity of Markov Decision Processes

In this chapter we analyze the computational complexity of various classes of Markov Decision Processes (MDP's). We study the complexity of computing the optimal cost function of an MDP. We introduce a multigrid version of the traditional single-grid algorithm and show that multigrid is better than single-grid. We also obtain lower bounds on complexity of the problem and show that the multigrid algorithm has optimal complexity with respect to the dependence on the accuracy parameter $\epsilon$ and nearly optimal in $\alpha$. This is in contrast to the single-grid algorithm which is non-optimal. We show that the discretization procedures in the preceding section are optimal. We also consider the problem of computing an $\epsilon$-optimal policy. We show that this problem is in a certain sense as hard as the problem of computing an $\epsilon$-optimal cost function.

We now give an outline of this chapter. In Section 3.1, we introduce our model of computation. We use a finite real number computer with an oracle. We also define various problem classes and make precise the notion of computational complexity. In Section 3.2, we analyze the computational complexity of a single-grid successive approximation algorithm and a multigrid version of the algorithm. We show that the multigrid version has better complexity than the single-grid version. We also study the effect of ergodicity conditions on the complexity of the problem. We compare our multigrid algorithm with other multigrid algorithms. In Section 3.3, we prove lower bounds. Using an adversary-type argument, we obtain the information-based lower bounds of the problem and as a result show that multigrid algorithm is within a factor of $O\left(\frac{1}{1-\alpha}\right)$ from the lower bound; it is optimal when the problem satisfies an ergodicity condition. In Section 3.4, we address the question of computing an $\epsilon$-optimal policy. We show that this problem is in a certain sense as hard as computing an $\epsilon$-optimal cost function. In Section 3.5, we consider some simple extensions and state our conclusions.

## 3.1 Preliminaries

In this section, we introduce our model of computation, review a well known approximation algorithm, and analyze the computational requirements of this algorithm. We first define the model of computation.

### 3.1.1 Model of Computation

Given that we are dealing with problems involving continuous variables, discrete models of computation such as Turing machines [Lewis and Papadimitriou (1981)] are not suitable. We shall use instead a continuous model in which arithmetic operations are performed on infinite precision real numbers [see Nemirovsky and Yudin (1983) and Traub, et al (1988) for related models].

Our model consists of three components:

1. A mechanism for reading the input.

   The input to the computation is provided by means of an "oracle" that works as follows:

   (i) To obtain information about $S$, a computer submits to the oracle "queries" consisting of an element $\iota \in \mathcal{I}_h^n$. If $\iota \cap S$ is empty then the oracle returns a special symbol to indicate this fact; otherwise, the oracle returns an element in $\iota \cap S$ and the volume $\lambda_n(\iota \cap S)$ of that set, where $\lambda_n(\cdot)$ stands for the Borel measure.

   (ii) To obtain information about $U(x)$, a computer submits to the oracle a pair $(h, x)$ and the oracle returns a list of the elements of the set $\tilde{U}_h(x)$.

   (iii) Finally, to obtain values of $g$ and $P$ at some specific points, the computer submits to the oracle a triple $(y, x, u)$, and the oracle returns the values of $P(y \mid x, u)$ and $g(x, u)$. We then say that the computer *samples* $(y, x, u)$.

2. The nature of the allowed computations.

   We consider a computing machine, or simply a "computer" that has the capability of performing comparisons and elementary arithmetic operations on infinite precision real numbers. Furthermore, the computer can use the results of earlier computations (or the answers to earlier queries) to decide what queries to submit to the oracle. The rules by which the computing machine decides at each step what to do next will be referred to as an "algorithm". [We are therefore dealing with an "adaptive" algorithm in the sense of Traub, et al (1988).]

3. A format for representing the output of the computation.

   In our case, the output of the computation is a function $J$ which is simple on $\mathcal{S}_h$, where the discretization parameter $h$ is to be decided by the computer itself. One possible format is the following. The computer first outputs the value of $h$, which implicitly specifies the partition $\mathcal{S}_h$ of $S$. It then outputs the pair $(\tilde{x}, J_h(\tilde{x}))$, for every $\tilde{x} \in \tilde{S}_h$.

There are some additional assumptions that have to be made in our particular context: The computer is provided the values of the dimensions $m$ and $n$ (of $C$ and $S$, respectively), the discount factor $\alpha$, the desired accuracy $\epsilon$, and the constant $K$ (of Assumptions A.1–A.5). Furthermore, if a $k$-stage ergodicity condition is assumed, the computer is also given the values of $k$ and of the ergodicity rate $\rho$.

The computational cost of an algorithm (also called its *complexity*) will be counted in a very simple manner: each query to the oracle costs one unit; similarly, each arithmetic operation or comparison costs one unit. [In a variation of this model, a query asking for the elements of a set $\tilde{U}_h(x)$ could have cost equal to the cardinality of the set returned by the oracle. Our complexity estimates, however, are not sensitive to minor variations of this type.]

### 3.1.2 Problem Classes

Let us fix the dimensions $m$, $n$, of $S$ and $C$, respectively, the constant $K$ of Assumptions A.1–A.5 (see Section 2.6), and the constants $k$ and $\rho$ involved in the ergodicity conditions (see Section 2.3). Once these parameters are fixed, let $\mathcal{P}_{\mathrm{std}}(\alpha)$ be the set of all MDP's with discount factor $\alpha$ and let $\mathcal{P}_{\mathrm{std}} = \cup_{\alpha \in (0,1)} \mathcal{P}_{\mathrm{std}}(\alpha)$. Let us consider an algorithm $\gamma$ that given any $\epsilon > 0$ and any MDP in $\mathcal{P}_{\mathrm{std}}$, returns an $\epsilon$-optimal cost function. We use $\mathrm{C}^{\gamma}_{\mathrm{std}}(\alpha, \epsilon)$ to denote the worst case running time of this algorithm for a particular value of $\epsilon$ and where the worst case is taken over all MDP's belonging to $\mathcal{P}_{\mathrm{std}}(\alpha)$. We then define the complexity $\mathrm{C}_{\mathrm{std}}(\alpha, \epsilon)$ of solving MDP's as the minimum of $\mathrm{C}^{\gamma}_{\mathrm{std}}(\alpha, \epsilon)$ over all algorithms $\gamma$ with the above mentioned properties.

There is a similar set of definitions for the more general class of MDP's where Assumption A.4′ (respectively, A.4″) is in effect. We use the subscript "sub" (respectively, "gen") instead of "std".

We also consider other problem classes where an additional ergodicity condition is assumed. We summarize below all the different problem classes we will encounter in this chapter.

(a) $\mathcal{P}_{\mathrm{std}}$—Assumptions A.1–A.5;

(b) $\mathcal{P}_{\mathrm{sub}}$—Assumptions A.1–A.3, A.4′, and A.5;

(c) $\mathcal{P}_{\mathrm{gen}}$—Assumptions A.1–A.3, A.4″, and A.5;

(d) $\mathcal{P}_{\mathrm{mix}}$—Assumptions A.1–A.5 and E.3 (a $k$-stage ergodicity condition with rate $2\rho$);

(e) $\mathcal{P}_{\mathrm{sub1}}$—Assumptions A.1–A.3, A.4′, A.5, and E.1 (a 1-stage ergodicity condition with rate $\rho$);

(f) $\mathcal{P}_{\mathrm{mix1}}$—Assumptions A.1–A.5 and E.1 (a 1-stage ergodicity condition with rate $\rho$).

### Order of Magnitude Notation

It is convenient to only consider order of magnitude estimates when arguing about algorithm or problem complexity. We thus introduce the following notation:

1. Let $f, g : (0,1] \mapsto [0,\infty)$ be functions of the grid-size $h$. We write $f = O(g)$ if there exist constants $c$ and $h_0 > 0$ such that $f(h) \le cg(h)$ for all $h \in (0, h_0]$. We also write $f = \Omega(g)$ if $g = O(f)$, and write $f = \Theta(g)$ if $f = O(g)$ and $f = \Omega(g)$.

2. Let $f, g : (0,1) \times (0,1] \mapsto [0,\infty)$ be functions of $\alpha$ and $\epsilon$. We write $f = O(g)$, if there exist constants $c$, $\epsilon_0 > 0$, and $\alpha_0 < 1$ such that $f(\epsilon,\alpha) \le cg(\epsilon,\alpha)$, for all $\epsilon \in (0,\epsilon_0]$ and $\alpha \in [\alpha_0, 1)$. We also write $f = \Omega(g)$ if $g = O(f)$, and write $f = \Theta(g)$ if $f = O(g)$ and $f = \Omega(g)$.

## 3.1.3 An Approximation Algorithm

To find an approximation of the optimal cost function to within some prespecified accuracy, a problem is first discretized using a suitable choice of grid-size. For the discretized problem, there are many algorithms for computing its optimal cost function $\tilde{J}_h^*$: for example, successive approximation, policy iteration, and linear programming [see Bertsekas (1987)]. In this chapter we will only discuss one of them—successive approximation. In the next chapter we will discuss policy iteration.

### Successive Approximation Algorithm

In this section, we introduce the successive approximation algorithm, review some known bounds on its speed of convergence, and discuss the effect of an ergodicity condition.

The successive approximation algorithm for a discretized problem proceeds as follows. We start with some function $J \in \mathcal{B}(S)$ which is simple on $\mathcal{S}_h$, and we compute $\tilde{T}_h^t J$ $(t = 1, 2, \ldots)$, where $\tilde{T}_h^t$ stands for the composition of $t$ replicas of $\tilde{T}_h$ and $\tilde{T}_h^0$ represents the identity operator. Since $\tilde{T}_h$ is a contraction operator (with contraction factor $\alpha$) and since $\tilde{J}_h^*$ is (by definition) a fixed point of $\tilde{T}_h$, we have

$$\|\tilde{J}_h^* - \tilde{T}_h^t J\|_\infty \le \alpha^t \|\tilde{J}_h^* - J\|_\infty. \tag{3.1.1}$$

In particular, $\tilde{T}_h^t J$ converges to $\tilde{J}_h^*$. A further consequence of the contraction property of $\tilde{T}_h$ is the following well-known error bound [Denardo (1967)]:

$$\|\tilde{J}_h^* - \tilde{T}_h^t J\|_\infty \le \frac{\alpha}{1-\alpha}\|\tilde{T}_h^t J - \tilde{T}_h^{t-1} J\|_\infty \le \frac{\alpha^t}{1-\alpha}\|\tilde{T}_h J - J\|_\infty. \tag{3.1.2}$$

In contrast to Eq. (3.1.1), the bounds of Eq. (3.1.2) can be computed with information available to the algorithm.

We note that the above algorithm applies to problems originated from $\mathcal{P}_{\text{gen}}$. And for problems in $\mathcal{P}_{\text{sd}}$, since $\tilde{T}_h$ is also a monotone operator and satisfies

$$T(J+c) = TJ + \alpha c, \quad \forall J \in \mathcal{B}(S), c \in \mathbf{R} \tag{3.1.3}$$

[cf. Proposition 2.2.1(c)], the convergence rate of the algorithm can be accelerated by using the following error bounds [see, for example, Bertsekas (1987)], that are valid for any $J \in \mathcal{B}(S)$:

$$\tilde{T}_h^{t+1}J + \frac{\alpha}{1-\alpha}\underline{c}_h^{t+1} \leq \tilde{J}_h^* \leq \tilde{T}_h^{t+1}J + \frac{\alpha}{1-\alpha}\overline{c}_h^{t+1} \tag{3.1.4}$$

where

$$\underline{c}_h^{t+1} = \min_{x \in S}\left\{(\tilde{T}_h^{t+1}J - \tilde{T}_h^t J)(x)\right\}; \tag{3.1.5}$$

$$\overline{c}_h^{t+1} = \max_{x \in S}\left\{(\tilde{T}_h^{t+1}J - \tilde{T}_h^t J)(x)\right\}. \tag{3.1.6}$$

(We have used "max" and "min" because $\tilde{T}_h^t J$ and $\tilde{T}_h^{t+1}J$ are simple functions.) The following is an approximation to $\tilde{J}_h^*$ that exploits the bounds of Eq. (3.1.4):

$$J^{t+1} = \tilde{T}_h^{t+1}J + \frac{\alpha}{2(1-\alpha)}\left[\underline{c}_h^{t+1} + \overline{c}_h^{t+1}\right]. \tag{3.1.7}$$

We subtract Eq. (3.1.4) from Eq. (3.1.7) to obtain

$$\|\tilde{J}_h^* - J^{t+1}\|_\infty \leq \frac{\alpha}{2(1-\alpha)}\|\tilde{T}_h^{t+1}J - \tilde{T}_h^t J\|_s \leq \frac{\alpha^{t+1}}{2(1-\alpha)}\|\tilde{T}_h J - J\|_s. \tag{3.1.8}$$

This bound is not much better than the bound of Eq. (3.1.2). However, for problems in $\mathcal{P}_{\text{mix}}$ where we assume that a $k$-stage ergodicity condition (Assumption E.3) holds, Theorem 2.3.2 yields

$$\|\tilde{T}_h^{tk+1}J - \tilde{T}_h^{tk}J\|_s \leq \alpha^{tk}(1-\rho)^t\|\tilde{T}_h J - J\|_s \leq (1-\rho)^t\|\tilde{T}_h J - J\|_s. \tag{3.1.9}$$

Combining with Eq. (3.1.8), we obtain

$$\|\tilde{J}_h^* - J^{tk+1}\|_\infty \leq \frac{(1-\rho)^t}{2(1-\alpha)}\|\tilde{T}_h J - J\|_s, \tag{3.1.10}$$

Thus, the distance of $J^t$ from $\tilde{J}_h^*$ contracts by a factor of at least $(1-\rho)$ every $k$ iterations. In particular, the convergence rate is independent of $\alpha$.

We next analyze the computational requirements (the complexity) of a typical iteration of the algorithm, using the model of computation defined previously.

## The Complexity of Evaluating $\tilde{T}_h J$

We estimate here the complexity of evaluating $\tilde{T}_h J$ according to the formula

$$\tilde{T}_h J(x) = \min_{\tilde{u} \in \tilde{U}(x)} \left\{ \tilde{g}_h(x, \tilde{u}) + \alpha \int_S J(y) \tilde{P}_h(y \mid x, \tilde{u}) dy \right\}, \tag{3.1.11}$$

for the case where $J$ is a simple function on $\mathcal{S}_h$. Since $\tilde{T}_h J$ also turns out to be a simple function on $\mathcal{S}_h$, we only need to determine the values of $\tilde{T}_h J$ for $\tilde{x} \in \tilde{S}_h$. Thus, $\tilde{T}_h J$ is determined by

$$\tilde{T}_h J(\tilde{x}) = \min_{\tilde{u} \in \tilde{U}(\tilde{x})} \left\{ \tilde{g}_h(\tilde{x}, \tilde{u}) + \alpha \sum_{\tilde{y} \in \tilde{S}_h} J(\tilde{y}) \tilde{P}_h(\tilde{y} \mid \tilde{x}, \tilde{u}) \lambda_n(\sigma_{\tilde{y}}) \right\}, \quad \tilde{x} \in \tilde{S}_h, \tag{3.1.12}$$

where $\lambda_n(\cdot)$ stands for the $n$-dimensional Borel measure.†

We make the following observations. Since $|\tilde{S}_h| = \mathrm{O}(h^{-n})$, and $|\tilde{U}_h(\tilde{x})| \le |\tilde{C}_h| = \mathrm{O}(h^{-m})$, there are $\mathrm{O}\left(h^{-(n+m)}\right)$ different pairs $(\tilde{x}, \tilde{u})$. Also, for any fixed $\tilde{x}$ and $\tilde{u}$, the right-hand side of Eq. (3.1.12) can be computed with $\mathrm{O}(h^{-n})$ operations, with most of the work needed for the summation. Thus, the total time spent in arithmetic operations and comparisons is $\mathrm{O}\left(h^{-(2n+m)}\right)$. Furthermore, $\mathrm{O}\left(h^{-(2n+m)}\right)$ oracle queries are sufficient for obtaining the required values of the functions $\tilde{g}_h$, $\tilde{P}_h$, and of the elements of the sets $\tilde{U}_h(\tilde{x})$. We have therefore proved the following:

**Lemma 3.1.1** *If $J$ is a simple function on $\mathcal{S}_h$, then the complexity of computing $\tilde{T}_h J$ is $\mathrm{O}(1/h^{2n+m})$.*

In our estimates, we have assumed that the minimization with respect to $\tilde{u}$ is carried out by exhaustive enumeration. In practice, the dependence on $u$ may have a special structure that can be exploited to reduce the computational requirements. Nevertheless, our analysis will be carried out for the general case where no special structure is assumed.

We now prove some useful lemmas.

**Lemma 3.1.2** *(Under Assumption A.4″) For every $J \in \mathcal{B}(S)$ and every $h \in (0, h_a]$, we have*

**(a)** $\|\tilde{T}_h J\|_\infty \le K + \alpha \|J\|_\infty$;

**(b)** *if* $\|J\|_\infty \le K/(1-\alpha)$ *then* $\|\tilde{T}_h J\|_\infty \le K/(1-\alpha)$;

**(c)** $\|\tilde{J}_h^*\|_\infty \le K/(1-\alpha)$.

†This formula should explain why we have assumed that the oracle can provide information on the volume of certain sets [see item 1(i) in Section 3.1.1]. If such volume information were not directly available, then it should be somehow estimated. Although this could be an important issue in practice, its theoretical aspects are somewhat tangential to the present discussion.

**Proof** We first prove (a). Fix an $x \in S$ and let $v \in \tilde{U}_h(x)$ attains the minimum in $\tilde{T}_h J(x)$; that is

$$\begin{aligned}\left|\tilde{T}_h J(x)\right| &= \left|\tilde{g}_h(x,v) + \alpha \int_S \tilde{q}_h(dy \mid x,u) J(y)\right| \\ &\le |\tilde{g}_h(x,v)| + \alpha \|J\|_\infty \|\tilde{q}_h(\cdot \mid x,u)\|_{\mathrm{v}} \\ &\le K + \alpha \|J\|_\infty.\end{aligned}$$

Since $x$ is arbitrary (a) follows, and now (b) follows immediately from (a).

Since the space of all $J \in \mathcal{B}(S)$ with $\|J\|_\infty \le K/(1-\alpha)$ forms a closed subset of $\mathcal{B}(S)$, it follows from (b) that $\tilde{T}_h$ is a contraction operator on this subspace; therefore, its fixed point $\tilde{J}_h^*$ lies in the subset, and (c) holds. □

An analogous result holds when Assumptions A.4 and E.3 are in effect:

**Lemma 3.1.3** *(Under Assumptions A.4) If $q$ satisfies a $k$-stage ergodicity condition (Assumption E.3) with ergodicity rate $2\rho$, then for any $J \in \mathcal{B}(S)$ and every $h \in (0, h_b]$, we have*

**(a)** *if $\|J\|_{\mathrm{s}} \le 2(k+1)K/\rho$, then*

$$\|\tilde{T}_h^{k\ell+1} J\|_{\mathrm{s}} \le 2(k+1)K/\rho, \quad \forall \ell = 1,2,\ldots. \tag{3.1.13}$$

**(b)** $\|\tilde{J}_h^*\|_{\mathrm{s}} \le 2kK/\rho$.

**Proof** Let $J^0(x) = 0$ for all $x \in S$. First, we note from Theorems 2.4.1 and 2.3.2 that $\tilde{T}_h$ is a $k$-stage contraction operator on $(\mathcal{B}(S), \|\cdot\|_{\mathrm{s}})$ with contraction factor $(1-\rho)$; from the proof of Lemma 2.4.5 that $\|\tilde{T}_h^k J^0\|_{\mathrm{s}} \le 2kK$; and from Lemma 3.1.2(a) that $\|\tilde{T}_h J\|_{\mathrm{s}} \le 2K + \|J\|_{\mathrm{s}}$ for all $J \in \mathcal{B}(S)$.

We now prove (a) by induction on $\ell$. For $\ell = 1$, we have

$$\begin{aligned}\|\tilde{T}_h^{k+1} J\|_{\mathrm{s}} &= \|\tilde{T}_h^{k+1} J - J^0\|_{\mathrm{s}} \\ &\le \|\tilde{T}_h^{k+1} J - \tilde{T}_h^k J^0\|_{\mathrm{s}} + \|\tilde{T}_h^k J^0 - J^0\|_{\mathrm{s}} \\ &\le (1-\rho)\|\tilde{T}_h J\|_{\mathrm{s}} + \|\tilde{T}_h^k J^0\|_{\mathrm{s}} \\ &\le (1-\rho)[2(k+1)K/\rho + 2K] + 2kK \\ &\le 2(k+1)K/\rho.\end{aligned}$$

Now assuming that Eq. (3.1.13) holds for $\ell$, we will prove it for $\ell + 1$.

$$\begin{aligned}\|\tilde{T}_h^{(\ell+1)k+1} J\|_{\mathrm{s}} &\le \|\tilde{T}_h^{(\ell+1)k+1} J - \tilde{T}_h^k J^0\|_{\mathrm{s}} + \|\tilde{T}_h^k J^0\|_{\mathrm{s}} \\ &\le (1-\rho)\|\tilde{T}_h^{\ell k+1} J\|_{\mathrm{s}} + 2kK \\ &\le (1-\rho)[2(k+1)K/\rho] + 2kK \\ &\le 2(k+1)K/\rho,\end{aligned}$$

as required. And (a) follows.

To prove (b), we have

$$\begin{aligned}\|\tilde{J}_h^*\|_{\mathrm{s}} &\le \|\tilde{T}_h^k \tilde{J}_h^* - \tilde{T}_h^k J^0\|_{\mathrm{s}} + \|\tilde{T}_h^k J^0 - J^0\|_{\mathrm{s}} \\ &\le (1-\rho)\|\tilde{T}_h^k \tilde{J}_h^*\|_{\mathrm{s}} + 2kK,\end{aligned}$$

as required. □

## 3.2 The Complexity of Successive Approximation

In this section, we use the model of computation of Section 3.1 to analyze the complexity of various successive approximation algorithms. We will consider separately (i) the general class of problems $\mathcal{P}_{\text{gen}}$ where Assumption A.4″ is in effect, and (ii) the special class of problems $\mathcal{P}_{\text{mix}}$ where Assumption A.4 is in effect and the problems are assumed to satisfy a $k$-stage ergodicity condition. We assume that the ergodicity rate is $2\rho$ so that for all discretization parameter $h \leq h_b$, the discretized problem satisfies a $k$-stage ergodicity condition with ergodicity rate $\rho$ (cf. Theorem 2.4.1). We first consider the traditional single-grid successive approximation algorithm.

### 3.2.1 Single-Grid Successive Approximation

The basic idea in single-grid successive approximation is that we choose a grid-size $h_f$ so that $\|J^* - \tilde{J}^*_{h_f}\|_\infty$ is small. We then keep applying the operator $\tilde{T}_{h_f}$ until a sufficiently accurate approximation of $\tilde{J}^*_{h_f}$ is obtained. We first consider the general case where Assumption A.4″ is in effect.

#### The General Case

Let $\epsilon$ be the desired accuracy. From the discretization error bound of Theorem 2.4.2, we have

$$\|J^* - \tilde{J}^*_{h_f}\|_\infty \leq \frac{K'}{(1-\alpha)^2} h_f. \tag{3.2.1}$$

Thus, if we let

$$h_f = \frac{(1-\alpha)^2\epsilon}{2K'}, \tag{3.2.2}$$

we obtain $\|J^* - \tilde{J}^*_{h_f}\|_\infty \leq \epsilon/2$. [Note that $h_f$ is chosen with the knowledge of $K'$, a constant which depends only on $K$ (cf. Theorem 2.4.2); therefore, $h_f$ can be determined from the values of $K$, $\alpha$, and $\epsilon$. Also note that Theorem 2.4.2 has the condition $h \leq h_a$. This of no concern because we are interested in the cases where $\epsilon \downarrow 0$ or $\alpha \uparrow 1$. In these cases, Eq. (3.2.2) shows that $h_f$ becomes arbitrarily small.]

With our choice of $h_f$, the complexity of evaluating $\tilde{T}_{h_f}J$, for some $J$ that is simple on $\mathcal{S}_h$, is $O\left(1/((1-\alpha)^2\epsilon)^{2n+m}\right)$ by Lemma 3.1.1.

Let $J^0(x) = 0$ for all $x \in S$, and apply $\tilde{T}_{h_f}$ on $J^0$ for $t$ times, where $t$ is the smallest integer satisfying

$$\frac{\alpha^t}{(1-\alpha)}\|\tilde{T}_h J^0\|_\infty \leq \frac{\epsilon}{2}.$$

Let $J^t = \tilde{T}^t_h J^0$. Then, Eq. (3.2.2) yields $\|\tilde{J}^*_{h_f} - \tilde{T}^t_h J^0\|_\infty \leq \epsilon/2$, and the triangle inequality shows that

$$\|J^* - J^t\|_\infty \leq \|J^* - \tilde{J}^*_{h_f}\|_\infty + \|\tilde{J}^*_{h_f} - J^t\|_\infty \leq \epsilon,$$

as desired.

We now bound the complexity of this algorithm. Since $\|\tilde{T}_h J^0\|_\infty \leq K$, it is seen that

$$t \leq \frac{\log\left[2K/((1-\alpha)\epsilon)\right]}{|\log \alpha|} + 1 = \mathrm{O}\left(\frac{\log \frac{1}{(1-\alpha)\epsilon}}{|\log \alpha|}\right).$$

Therefore, the complexity of the algorithm is

$$\mathrm{O}\left(\frac{\log \frac{1}{(1-\alpha)\epsilon}}{|\log \alpha|}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right).$$

## The Special Case

We now assume that Assumption A.4 is in effect and impose an ergodicity condition. Theorem 2.4.2 yields

$$\|J^* - \tilde{J}^*_{h_f}\|_\infty \leq \frac{K''}{(1-\alpha)} h_f.$$

We wish to have $\|J^* - \tilde{J}^*_{h_f}\|_\infty \leq \epsilon/2$ and this can be accomplished by letting

$$h_f = \frac{(1-\alpha)\epsilon}{2K''}.$$

Accordingly, the complexity of each iteration is $\mathrm{O}\left(1/((1-\alpha)\epsilon)^{2n+m}\right)$.

Let again $J^0(x) = 0$ for all $x \in S$, and apply $\tilde{T}_{h_f}$ on $J^0$ for $lk+1$ times. Let

$$J^{\ell k+1} = \tilde{T}_h^{lk+1} J + \frac{\alpha}{2(1-\alpha)}\left[\underline{c}_h^{\ell k+1} + \overline{c}_h^{\ell k+1}\right].$$

[Cf. Eq. (3.1.7).] Equation (3.1.10) yields

$$\|\tilde{J}^*_{h_f} - J^{lk+1}\|_\infty \leq \frac{(1-\rho)^l}{2(1-\alpha)}\|\tilde{T}_{h_f} J^0\|_s \leq \frac{(1-\rho)^l}{2(1-\alpha)} 2K \qquad (3.2.3)$$

We now bound the complexity of the algorithm. We desire to have $\|\tilde{J}^*_{h_f} - J^{lk+1}\|_\infty \leq \epsilon/2$ and, from Eq. (3.2.3), this can be achieved with

$$l \leq \frac{\log \frac{2K}{(1-\alpha)\epsilon}}{|\log(1-\rho)|} + 1 = \mathrm{O}\left(\log \frac{1}{(1-\alpha)\epsilon}\right).$$

So, the complexity of the algorithm is

$$\mathrm{O}\left(\log \frac{1}{(1-\alpha)\epsilon}\left[\frac{1}{(1-\alpha)\epsilon}\right]^{2n+m}\right).$$

We summarize our results in the following theorem:

**Theorem 3.2.1** *There holds*

$$\mathbf{C}_{\text{gen}}(\alpha, \epsilon) = O\left(\frac{\log \frac{1}{(1-\alpha)\epsilon}}{|\log \alpha|}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right),$$

$$\mathbf{C}_{\text{mix}}(\alpha, \epsilon) = O\left(\log \frac{1}{(1-\alpha)\epsilon}\left[\frac{1}{(1-\alpha)\epsilon}\right]^{2n+m}\right).$$

*Furthermore, the complexity of the single-grid successive approximation algorithm is within these bounds.*

### 3.2.2 Multigrid Successive Approximation

We now introduce a multigrid version of the algorithm of Section 3.2.1 and estimate its complexity. The first iterations of this algorithm are executed with a relatively large value of $h$ (coarse grid) and the value of $h$ is gradually reduced (grid refinement) as the algorithm proceeds. The basic idea is that the results of the initial iterations are fairly inaccurate approximations of $J^*$, so the use of a very fine grid is unnecessary. Thus, most iterations are executed on relatively coarse grids, with much less computational costs, and the overall complexity of the algorithm is improved.

Multigrid methods have been extensively studied in the context of partial differential equations, and have been found to lead to substantially faster convergence (both theoretically, and in practice) [Brandt (1986) and Hackbusch(1985)]. Some alternative methods [Akian, et al (1988) and Hoppe (1986)] are discussed at the end of the section.

As in Section 3.2.1, we will analyze the complexity of the multigrid algorithm for the general case and for the special case where an ergodicity condition is imposed. Our results show that the complexity of multigrid successive approximation is (for both cases) better than that of the single-grid method by a factor of $O(\log[1/((1-\alpha)\epsilon)])$, and is optimal in a sense to be discussed in Section 3.3.

### The General Case

The algorithm starts by fixing an appropriate coarsest grid-level (discretization parameter) $h_0$. The choice of $h_0$ is independent of $\alpha$ and $\epsilon$, but we require that $h_0 \leq h_a$, so that the discretization error bound of Theorem 2.4.2 applies. We then compute the function $\tilde{J}^*_{h_0}$ exactly, and let $J^F_{h_0} = \tilde{J}^*_{h_0}$. We switch to a new grid-level by replacing $h_0$ by $h_0/2$, and use $J^F_{h_0}$ to initialize the computations at the new grid-level.

More generally, at any grid-level $h$, we do the following. We start with an initial estimate $J^I_h$ and we compute $\tilde{T}^t_h J^I_h$, $t = 1, 2, \ldots, t(h)$, where $t(h)$ is the smallest positive integer such that

$$\|\tilde{T}^{t(h)}_h J^I_h - \tilde{T}^{t(h)-1}_h J^I_h\|_\infty \leq \frac{K'h}{\alpha(1-\alpha)}. \tag{3.2.4}$$

[The fact that such a $t(h)$ exists is evident because $\tilde{T}_h^t J_h^I$ converges.] At that point, we let $J_h^F = \tilde{T}_h^{t(h)} J_h^I$ which is our final estimate at the current grid-level. Then, Eq. (3.2.3) yields

$$\|\tilde{J}_h^* - J_h^F\|_\infty \leq \frac{\alpha}{(1-\alpha)} \|\tilde{T}_h^{t(h)} J_h^I - \tilde{T}_h^{t(h)-1} J_h^I\|_\infty \leq \frac{\alpha^{t(h)}}{(1-\alpha)} \|\tilde{T}_h J_h^I - J_h^I\|_\infty. \quad (3.2.5)$$

If

$$\frac{K'h}{(1-\alpha)^2} \leq \frac{\epsilon}{2}, \quad (3.2.6)$$

the algorithm terminates. Otherwise, we replace $h$ by $h/2$ and use the final function $J_h^F$ of the current grid-level to initialize the computations at the next grid-level. That is, $J_{h/2}^I = J_h^F$.

It is clear that after a finite number of grid-level changes, Eq. (3.2.6) will be satisfied, and this shows that the algorithm eventually terminates.

We now verify correctness of the algorithm. Let $h_f$ be the final grid-level at which the algorithm terminates. Using Theorem 2.4.2, we have

$$\|J^* - \tilde{J}_{h_f}^*\|_\infty \leq \frac{K'h_f}{(1-\alpha)^2} \leq \frac{\epsilon}{2}. \quad (3.2.7)$$

Furthermore, Eqs. (3.2.4) and (3.2.5) yield

$$\|\tilde{J}_{h_f}^* - J_{h_f}^F\|_\infty \leq \frac{\alpha}{(1-\alpha)} \|\tilde{T}_{h_f}^{t(h_f)} J_{h_f}^I - \tilde{T}_{h_f}^{t(h_f)-1} J_{h_f}^I\|_\infty \leq \frac{K'h_f}{(1-\alpha)^2} \leq \frac{\epsilon}{2}. \quad (3.2.8)$$

Equations (3.2.7)-(3.2.8) and the triangle inequality yield $\|J^* - J_{h_f}^F\|_\infty \leq \epsilon$, as desired.

In order to develop a complexity estimate, we need to bound the number $t(h)$ of iterations at each grid-level. This is done in the following two lemmas.

**Lemma 3.2.1** *For $h \in \{h_0/2, h_0/4, \ldots, h_f\}$ and every $t \in \{1, \ldots, t(h)\}$, we have $\|\tilde{T}_h^t J_h^I\|_\infty \leq K/(1-\alpha)$. In particular, $\|J_h^F\|_\infty \leq K/(1-\alpha)$.*

**Proof** The proof proceeds by induction. We have $\|J_{h_0/2}^I\|_\infty = \|\tilde{J}_{h_0}^*\|_\infty \leq K/(1-\alpha)$. Then using Lemma 3.1.2, we have $\|\tilde{T}_h J_h^I\|_\infty \leq K/(1-\alpha)$ and continuing inductively, the same bounds hold for $\|\tilde{T}_h^t J_h^I\|_\infty$, $t = 1, 2, \ldots, t(h)$. And since $J_{h/2}^I = J_h^F$, we have

$$\|J_{h/2}^I\|_\infty = \|J_h^F\|_\infty = \|\tilde{T}_h^{t(h)} J_h^I\|_\infty \leq K/(1-\alpha).$$

□

**Lemma 3.2.2** *There exists a constant $c$, independent of $\alpha$ and $\epsilon$, such that $t(h) \leq c/|\log \alpha|$, for $h = h_0/2, h_0/4, \ldots, h_f$.*

**Proof** Fix some $h \in \{h_0/4, h_0/8, \ldots, h_f\}$ and let $\hat{J} = \tilde{T}_{2h}^{t(2h)-1} J_{2h}^I$. (Thus, $\hat{J}$ is the function available just before the last iteration at grid-level $2h$.) Then, Eq. (3.2.4) yields

$$\|\tilde{T}_{2h}\hat{J} - \hat{J}\|_\infty \le \frac{2K'h}{\alpha(1-\alpha)}. \tag{3.2.9}$$

Using the triangle inequality and Eq. (3.2.9), and Theorem 2.4.2, we have

$$\begin{aligned}
&\|\tilde{T}_h\tilde{T}_{2h}\hat{J} - \tilde{T}_{2h}\hat{J}\|_\infty \\
&\quad \le \|\tilde{T}_h\tilde{T}_{2h}\hat{J} - T\tilde{T}_{2h}\hat{J}\|_\infty + \|T\tilde{T}_{2h}\hat{J} - \tilde{T}_{2h}\tilde{T}_{2h}\hat{J}\|_\infty + \|\tilde{T}_{2h}\tilde{T}_{2h}\hat{J} - \tilde{T}_{2h}\hat{J}\|_\infty \\
&\quad \le \|\tilde{T}_h\tilde{T}_{2h}\hat{J} - T\tilde{T}_{2h}\hat{J}\|_\infty + \|T\tilde{T}_{2h}\hat{J} - \tilde{T}_{2h}\tilde{T}_{2h}\hat{J}\|_\infty + \alpha\|\tilde{T}_{2h}\hat{J} - \hat{J}\|_\infty \\
&\quad \le (K_1 + \alpha K_2\|\tilde{T}_{2h}\hat{J}\|_\infty)h + (K_1 + \alpha K_2\|\tilde{T}_{2h}\hat{J}\|_\infty)2h + \alpha\frac{2K'h}{\alpha(1-\alpha)}.
\end{aligned} \tag{3.2.10}$$

We have $\|\tilde{T}_{2h}\hat{J}\|_\infty = \|J_{2h}^F\|_\infty \le \frac{K}{1-\alpha}$ (cf. Lemma 3.2.1). Using this inequality in Eq. (3.2.10), we obtain

$$\|\tilde{T}_h\tilde{T}_{2h}\hat{J} - \tilde{T}_{2h}\hat{J}\|_\infty \le 3\Big(K_1 + \alpha K_2\frac{K}{1-\alpha}\Big)h + \frac{2K'h}{(1-\alpha)} \le \frac{5K'}{1-\alpha}h, \tag{3.2.11}$$

where the last inequality follows from the fact $K' = K_1 + K_2K$ [cf. Theorem 2.4.2]. Note that the left-hand side of Eq. (3.2.11) is equal to $\|\tilde{T}_hJ_h^I - J_h^I\|_\infty$. Therefore, using Eq. (3.2.11) and the fact that $\tilde{T}_h$ is a contraction operator, with contraction factor $\alpha$, we obtain

$$\|\tilde{T}_h^tJ_h^I - \tilde{T}_h^{t-1}J_h^I\|_\infty \le \alpha^{t-1}\|\tilde{T}_hJ_h^I - J_h^I\|_\infty \le \alpha^{t-1}\frac{5K'}{1-\alpha}h. \tag{3.2.12}$$

In particular, if $t$ is chosen so that $5\alpha^t \le 1$, then the termination condition of Eq. (3.2.4) is satisfied. This shows that $t(h)$ is no larger than the smallest $t$ such that $5\alpha^t \le 1$ and, therefore, $t(h) \le c/|\log\alpha|$, where $c = \log 5$.

The proof for the case $h = h_0/2$ is identical, provided that we define $\hat{J} = J_{h_0/2}^I = \tilde{J}_{h_0}^*$. We then have $\|\tilde{T}_{2h}\hat{J} - \hat{J}\|_\infty = \|\tilde{T}_{h_0}\tilde{J}_{h_0}^* - \tilde{J}_{h_0}^*\|_\infty = 0$ and inequality (3.2.4) is trivially true. The rest of the argument holds without any changes. □

Note that at each grid-level $h$ we start with a function $J_h^I$ that is simple on $\mathcal{S}_{2h}$ and, therefore, simple on $\mathcal{S}_h$. Since only simple functions are involved, Lemma 3.1.1 provides an estimate of the complexity of each iteration. Using also Lemma 3.2.2 to estimate the number of iterations at each grid-level, the total complexity of the algorithm is

$$\begin{aligned}
\mathbf{C}_{\text{gen}}(\alpha,\epsilon) &= O\left(\frac{1}{|\log\alpha|}\left[(1/h_f)^{2n+m} + (1/2h_f)^{2n+m} + (1/4h_f)^{2n+m} + \cdots\right]\right) \\
&= O\left(\frac{1}{|\log\alpha|}\left[\frac{1}{h_f}\right]^{2n+m}\left[1 + \frac{1}{2} + \frac{1}{4} + \cdots\right]\right)
\end{aligned}$$

$$= O\left(\frac{1}{|\log\alpha|}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right). \tag{3.2.13}$$

[The last step in Eq. (3.2.13) uses the relation $h_f = \Omega\left(\epsilon(1-\alpha)^2\right)$ which is a consequence of the termination criterion (3.2.6).] Note that we have ignored the computations involved at the first grid-level $h_0$. This is justifiable because we can compute $J^*_{h_0}$ with a number of operations that is independent of $\alpha$ and $\epsilon$ (for example, using linear programming or policy iteration) and let $J^F_{h_0} = \tilde{J}^*_{h_0}$. In practice, we might only compute an approximation of $\tilde{J}^*_{h_0}$, for example, by using the successive approximation algorithm at grid-level $h_0$. It is easily verified that such a modification does not change our complexity estimate.

## The Special Case

We now assume that Assumption A.4 is in effect and the problem satisfies a $k$-stage ergodicity condition. The algorithm is almost the same except for the following differences. The initial grid-size $h_0$ is chosen to satisfy $h_0 \leq h_b$, where $h_b$ is the constant of Theorem 2.4.1. Furthermore, at any grid level $h$, we do the following. Starting with an initial estimate $J^I_h$, we compute $\tilde{T}^{kt+1}_h J^I_h$ for $t = 1, 2, \dots, t(h)$, where $t(h)$ is the smallest positive integer such that

$$\|\tilde{T}^{kt(h)+1}_h J^I_h - \tilde{T}^{t(h)}_h J^I_h\|_s \leq \frac{2K''h}{\alpha(1-\alpha)},$$

where $K''$ is the constant of Theorem 2.4.3. We let [cf. Eqs. (3.1.7)]

$$J^F_h = \tilde{T}^{kt(h)+1}_h J^I_h + \frac{\alpha}{2(1-\alpha)}\left[\underline{c}^{kt(h)+1}_h + \bar{c}^{kt(h)+1}_h\right],$$

which is our final estimate at the current grid-level. Then Eq. (3.1.8) yields

$$\|\tilde{J}^*_h - J^F_h\|_\infty \leq \frac{\alpha}{2(1-\alpha)}\|\tilde{T}^{kt(h)+1}_h J^I_h - \tilde{T}^{kt(h)}_h J^I_h\|_s \leq \frac{(1-\rho)^{t(h)}}{2(1-\alpha)}\|\tilde{T}_h J^I_h - J^I_h\|_s.$$

The termination criterion of Eq. (3.2.6) is replaced by

$$\frac{K''h}{1-\alpha} \leq \frac{\epsilon}{2}, \tag{3.2.14}$$

The proof of termination is the same as in the general case (except that we use $\|\cdot\|_s$ instead of $\|\cdot\|_\infty$). Correctness of the algorithm also follows similarly, except that we have to invoke Theorem 2.4.3 instead of Theorem 2.4.2. We now bound the number of iterations at each grid-level.

**Lemma 3.2.3** *For $h \in \{h_0, h_0/2\dots, h_f\}$ and every $t \in \{1, 2, \dots, t(h)\}$, we have $\|\tilde{T}^{tk+1}_h J^I_h\|_s \leq 2(k+1)K/\rho$. In particular, $\|J^F_h\|_s \leq 2(k+1)K/\rho$.*

**Proof** The proof is similar to the proof of Lemma 3.2.1, except that we use Lemma 3.1.3 instead of Lemma 3.1.2 and the fact that $\|J^F_{h_0/2}\|_s = \|\tilde{J}^*_{h_0/2}\|_s \leq 2(k+1)K/\rho$. □

**Lemma 3.2.4** *Under the ergodicity condition (Assumption E.3), there exists a constant c, independent of $\alpha$ and $\epsilon$, such that $t(h) \leq c$, for $h = h_0/2, h_0/4, \ldots, h_f$.*

**Proof** The proof is identical with the proof of Lemma 3.2.2. The only differences are that, under Assumption A.4 and the $k$-stage ergodicity condition, Eq. (3.2.12) gets replaced by

$$\|J_h^{k\ell+1} J_h^I - \tilde{T}_h^{k\ell} J_h^I\|_s \leq (1-\rho)^\ell \|\tilde{T}_h J_h^I - J_h^I\|_s.$$

As $\alpha$ is replaced by the absolute constant $1-\rho$, it follows that $t(h)$ is also bounded by an absolute constant independent of $\alpha$. □

We now use Lemma 3.2.2 to estimate the complexity of the algorithm. We obtain

$$\begin{aligned}
\mathbf{C}_{\mathrm{mix}}(\alpha,\epsilon) &= O\left((1/h_f)^{2n+m} + (1/2h_f)^{2n+m} + (1/4h_f)^{2n+m} + \cdots\right) \\
&= O\left(\left[\frac{1}{h_f}\right]^{2n+m}\left[1 + \frac{1}{2} + \frac{1}{4} + \cdots\right]\right) \\
&= O\left(\left[\frac{1}{(1-\alpha)\epsilon}\right]^{2n+m}\right).
\end{aligned}$$

We have used in the last step the fact $h_f = \Omega\left(\epsilon(1-\alpha)\right)$ which is a consequence of Eq. (3.2.14). We summarize our conclusions:

**Theorem 3.2.2** *There holds*

$$\mathbf{C}_{\mathrm{gen}}(\alpha,\epsilon) = O\left(\frac{1}{|\log\alpha|}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right); \qquad (3.2.15)$$

$$\mathbf{C}_{\mathrm{mix}}(\alpha,\epsilon) = O\left(\left[\frac{1}{(1-\alpha)\epsilon}\right]^{2n+m}\right). \qquad (3.2.16)$$

*Furthermore, the complexity of the multigrid successive approximation algorithms presented in this section is within these bounds.*

A comparison of Theorems 3.2.1 and 3.2.2 shows that the multigrid algorithm is an improvement over its single-grid counterpart.

### 3.2.3 A Comparison with Other Multigrid Algorithms

We now compare our multigrid algorithm with other multigrid algorithms reported in the literature and make some observations First, we compare our multigrid algorithm with the algorithms reported by Akian, et al (1988) and Hoppe (1986). The main differences are as follows:

1. The problems solved in these references are continuous-time problems that lead to an elliptic partial differential equation, while we are dealing with discrete-time problems that lead to an integral equation.

2. The algorithms of Akian, et al (1988) and Hoppe (1986) are based on policy iteration, whereas we use successive approximation. The policy iteration algorithm involves a "policy evaluation" step which amounts to solving the linear equation $T_\mu J_\mu = J_\mu$, where $\mu$ is a certain policy. It is then suggested that the solution of this equation be carried out using a multigrid algorithm. Whereas an algorithm similar to ours could be suitable for that task, the multigrid algorithm of Akian, et al, (1988) and Hoppe (1986) is radically different. Ours proceeds from coarser to finer grids; in contrast the algorithm in these references moves repeatedly up and down between different grids. This latter strategy is certainly appropriate for the solution of certain partial differential equations, but it is unclear if it could be beneficial (theoretically) for the solution of discounted-cost, discrete-time problems.

3. The complexity analysis in Akian, et al (1988) is carried out only for a specific example. Furthermore, the analysis is based on a heuristic correspondence between policy iteration and Newton's method, together with an implicit assumption that Newton's method converges very fast. [There is no complexity analysis in Hoppe (1986), only the proof of convergence is shown.]

We now compare our multigrid algorithm with the adaptive aggregation and disaggregation algorithm of Bertsekas and Castanon (1986).

1. The algorithm of Bertsekas and Castanon is for general finite-state problems whereas ours is for finite-state problems which arise from the discretization of Lipschitz continuous MDP's.

2. Again their algorithm moves up and down the "grid"—the different aggregation classes correspond to changes in grid-size—whereas ours only go from coarse to fine.

### 3.2.4 Notes

Our multigrid algorithm is often called a *one-way* multigrid algorithm in the sense that it only proceeds from coarse to fine grid. To use the algorithm for practical problems, it should be modified (based on the problem) to exploit the special structure of the problem. However, this algorithm, as it is, suits our purpose because it is easy to analyze.

There may be practical reasons for choosing policy iteration over successive approximation. If policy iteration is employed, our multigrid algorithm may be still used for policy evaluation. We will explore this topic further in Section 4.2.

On the other hand we only make use of the contraction property of the dynamic programming operator. Thus our multigrid algorithm is applicable to general nonlinear fixed point problems. (This is the topic addressed in Chapter 5.) This may not be the case with the other algorithms (based on policy iteration).

There is a simple intuitive reason for our grid-refinement criterion—we always iterate until the discretization error is comparable to the approximation error. The reason is that we should always iterate on the coarsest grid (which is also the cheapest per iteration) until the discretization error of the grid makes it not worthwhile to work on that grid. (This seems to be a general rule.) We will discuss this issue further in Chapter 5.

## 3.3 The Information-Based Complexity of Markov Decision Processes

In this section, we establish lower bounds on the computational complexity of computing $\epsilon$-optimal cost functions of MDP's and show that the multigrid successive approximation algorithm described in Section 3.2 has optimal complexity in its dependence on the accuracy parameter $\epsilon$ and nearly optimal in the discount factor $\alpha$. Our results also show that the discretization error bounds of Section 2.4 are optimal. First, we introduce the notion of information-based complexity.

### 3.3.1 Information-Based Complexity

Recall that the complexity of an algorithm is defined as the sum of

**(a)** the number of oracle queries made by the algorithm, and

**(b)** the number of arithmetic operations performed by the algorithm.

A very fruitful method for establishing lower bounds on the complexity of any algorithm consists of lower bounding the number of queries that have to be submitted for the desired accuracy to be attainable. The typical argument here is that if the number of queries is small then the available information on the problem being solved is insufficient. Accordingly, the least number of queries necessary is often called the *information–based* complexity of the problem.

To distinguish the information-based complexity from the total complexity $\mathbf{C}$, we will use the superscript "info"; in particular, $\mathbf{C}^{\text{info}} \leq \mathbf{C}$.

## 3.3.2 Simplifying Assumptions

We now introduce some simplifying assumptions which will be used throughout this chapter. We consider the special case where the state space $S = [0,1]^n$ and $U(x) = U = [0,1]^m$ for all $x \in S$. (Assumption A.3 is trivially satisfied by the assumption that $U(x) = U$ for all $x$.)

We also replace Assumptions A.2 and A.5 by the following (stronger) Assumption

**A.6** $|P(y \mid x,u) - P(y' \mid x',u')| \le K\|(y,x,u) - (y',x',u')\|_\infty, \quad \forall y,x \in S, u \in U.$

Throughout this section we will impose Assumption A.6. The problems considered in Section 3.2 are more general than the special cases studied here; thus, the lower bound to be derived in Section 3.3.4 apply more generally.

Finally we rewrite Assumption E.1 as follows:

**E.1** There exists a constant $\rho > 0$ such that

$$\int_S \min_{x\in S,\ u\in U} P(y \mid x,u)\,dy \ge \rho, \quad \forall x \in S, u \in U. \tag{3.3.1}$$

[Under Assumption A.6, it can be shown that $\min_{x\in S,\ u\in U} P(\cdot \mid x,u)$ is a Lipschitz continuous function on $S$ and that Assumption E.1 above is equivalent to the Assumption E.1 given earlier in Section 2.6.]

Our computational task is completely determined by the functions $P$ and $g$, the discount factor $\alpha$, and the desired accuracy $\epsilon$. Accordingly, a tuple $(P, g, \alpha, \epsilon)$ will be called an *instance*. In this context, a problem is a class of instances.

In this section, we will consider three different problems:

(a) Problem $\mathcal{P}_{\mathrm{std}}$ that consists of all instances that satisfy Assumption A.4. [In particular, $P(\cdot \mid x,u)$ is a probability measure for all $(x,u)$.]

(b) Problem $\mathcal{P}_{\mathrm{mix1}}$ that consists of all instances that satisfy Assumptions A.4 and E.1. (That is, an ergodicity condition is also in effect.)

(c) Problem $\mathcal{P}_{\mathrm{sub1}}$ that consists of all instances that satisfy Assumptions A.4′ and E.1. [That is, the ergodicity condition is still in effect, but $P(\cdot \mid x,u)$ is a subprobability measure.]

## 3.3.3 Information-Based Upper Bounds

Let the dimensions $n$, $m$ of the state and control spaces be fixed and let us view the constants $K$ and $\rho$ of the assumptions as absolute constants.

Let us fix some $\epsilon$ and $\alpha$. The following upper bounds, together with discretization procedures that stay within these bounds, are derived in Section 2.4:

$$\mathbf{C}^{\mathrm{info}}_{\mathrm{gen}}(\alpha,\epsilon) = O\left(\frac{1}{((1-\alpha)^2\epsilon)^{2n+m}}\right). \tag{3.3.2}$$

$$\mathbf{C}^{\text{info}}_{\text{mix}}(\alpha,\epsilon) = O\left(\frac{1}{((1-\alpha)\epsilon)^{2n+m}}\right). \qquad (3.3.3)$$

Note that the upper bounds of Eq. (3.3.2) apply to $\mathbf{C}^{\text{info}}_{\text{sub1}}$ and $\mathbf{C}^{\text{info}}_{\text{std}}$ as well, and the upper bounds of Eq. (3.3.3) apply to $\mathbf{C}^{\text{info}}_{\text{mix1}}$.

## 3.3.4 Information-Based Lower Bounds

We now prove that the upper bounds of Eqs. (3.3.2)-(3.3.3) are tight, by establishing the corresponding lower bounds. Our results rest on an "adversary" argument that is very common in the study of information-based complexity [Traub, et al (1988)]. The outline of the argument is as follows. Suppose that a certain algorithm makes at most $A$ queries. We consider a particular instance $(P, g, \alpha, \epsilon)$ and we let $X$ be the set of triples $(y, x, u)$ sampled by the algorithm when presented with that instance. We then construct an alternative instance $(\hat{P}, \hat{g}, \alpha, \epsilon)$ such that $P(y \mid x, u) = \hat{P}(y \mid x, u)$ and $g(x, u) = \hat{g}(x, u)$, for all $(y, x, u) \in X$. The algorithm has no means of distinguishing between the two instances and must produce the same output $J$ in both cases. Let $J^*$ and $\hat{J}^*$ be the optimal cost functions for the two problems. If we can manage so that $\|J^* - \hat{J}^*\|_\infty > 2\epsilon$, then at least one of the inequalities $\|J - J^*\|_\infty > \epsilon$ and $\|J - \hat{J}^*\|_\infty > \epsilon$ must hold. It follows that the algorithm cannot succeed for all problem instances, and therefore the information-based complexity of the problem is larger than the cardinality of $A$.

**Theorem 3.3.1** *(Lower bound under Assumptions A.4 and E.1) For any $K > 0$, $\rho \in (0,1)$, $m$, and $n$, we have*

$$\mathbf{C}^{\text{info}}_{\text{mix1}}(\alpha,\epsilon) = \Omega\left(\left[\frac{1}{(1-\alpha)\epsilon}\right]^{2n+m}\right).$$

*It follows that*

$$\mathbf{C}_{\text{mix1}}(\alpha,\epsilon) = \Omega\left(\left[\frac{1}{((1-\alpha)\epsilon}\right]^{2n+m}\right).$$

**Proof** We only prove the result for the case $K = 1$ and $\rho = 1/2$. The proof for the general case is identical except for a minor modification discussed at the end of the proof. Let us also fix the dimensions $m$, $n$ of the problem. Throughout the proof, an *absolute constant* will stand for a constant that can only depend on $m$, $n$, but not on any other parameters.

We fix some $\epsilon > 0$ and some $\alpha \in (1/2, 1)$. Let us consider some algorithm that is correct for the problem $\mathcal{P}_{\text{mix1}}$ and suppose that the number of queries is at most $A$ for every instance with those particular values of $\alpha$ and $\epsilon$. We will derive a lower bound on $A$.

We choose a positive scalar $\delta$ so that $1/\delta$ is an integer multiple of 16 and such that

$$\frac{1}{\delta_0} \le \frac{1}{\delta} \le \frac{1}{\delta_0} + 16, \qquad (3.3.4)$$

where $\delta_0$ satisfies

$$A = \frac{1}{4\delta_0^{2n+m}}. \tag{3.3.5}$$

We partition the set $S \times S \times U$ into cubic cells of volume $\delta^{2n+m}$. (In particular, there will be $1/\delta^{2n+m}$ cells.) This is done by first specifying the "centers" of the cells. Let $\tilde{S}$ be the set of all $x = (x_1, \ldots, x_n) \in S$ such that each component $x_i$ is of the form $x_i = (t + (1/2))\delta$, where $t$ is a nonnegative integer smaller than $1/\delta$. Similarly, we let $\tilde{U}$ be the set of all $u = (u_1, \ldots, u_m) \in U$ such that each $u_i$ is of the form $u_i = (t + (1/2))\delta$, where again $t$ is a nonnegative integer smaller than $1/\delta$. For any $(\tilde{y}, \tilde{x}, \tilde{u}) \in \tilde{S} \times \tilde{S} \times \tilde{U}$, we define the *cell* $C_{\tilde{y},\tilde{x},\tilde{u}}$ by letting

$$C_{\tilde{y},\tilde{x},\tilde{u}} = \Big\{(y, x, u) \in S \times S \times U \ \mid \|(y, x, u) - (\tilde{y}, \tilde{x}, \tilde{u})\|_\infty < \frac{\delta}{2}\Big\}.$$

Clearly, the cardinality of $\tilde{S}$ and $\tilde{U}$ is $1/\delta^n$ and $1/\delta^m$, respectively. It follows that there is a total of $1/\delta^{2n+m}$ cells. Note that distinct cells are disjoint.

For any $(\tilde{y}, \tilde{x}, \tilde{u}) \in \tilde{S} \times \tilde{S} \times \tilde{U}$, we define a function $E_{\tilde{y},\tilde{x},\tilde{u}} : S \times S \times U \mapsto \mathbf{R}$, by letting

$$E_{\tilde{y},\tilde{x},\tilde{u}}(y \mid x, u) = \begin{cases} 0 & \text{if } (y, x, u) \notin C_{\tilde{y},\tilde{x},\tilde{u}}; \\ \frac{\delta}{2} - \|(y, x, u) - (\tilde{y}, \tilde{x}, \tilde{u})\|_\infty & \text{if } (y, x, u) \in C_{\tilde{y},\tilde{x},\tilde{u}}. \end{cases} \tag{3.3.6}$$

Thus, $E_{\tilde{y},\tilde{x},\tilde{u}}$ is just a "pyramid" of height $\delta/2$ whose base is the cell $C_{\tilde{y},\tilde{x},\tilde{u}}$. The triangle inequality applied to the norm $\|\cdot\|_\infty$ shows that for all $(y, x, u)$, $(y', x', u') \in C_{\tilde{y},\tilde{x},\tilde{u}}$,

$$|E_{\tilde{y},\tilde{x},\tilde{u}}(y \mid x, u) - E_{\tilde{y},\tilde{x},\tilde{u}}(y' \mid x', u')| \leq \|(y, x, u) - (y', x', u')\|_\infty.$$

Thus, $E_{\tilde{y},\tilde{x},\tilde{u}}$ satisfies the Lipschitz continuity Assumption A.6 with Lipschitz constant $K = 1$, on the set $C_{\tilde{y},\tilde{x},\tilde{u}}$. The function $E_{\tilde{y},\tilde{x},\tilde{u}}$ is continuous at the boundary of $C_{\tilde{y},\tilde{x},\tilde{u}}$ and is zero outside $C_{\tilde{y},\tilde{x},\tilde{u}}$. Thus, $E_{\tilde{y},\tilde{x},\tilde{u}}$ is obtained by piecing together in a continuous manner a Lipschitz continuous function and a constant function. It follows that $E_{\tilde{y},\tilde{x},\tilde{u}}$ is Lipschitz continuous on the set $S \times S \times U$, with Lipschitz constant 1.

We define an instance $(P, g, \alpha, \epsilon)$ by letting

$$g(x, u) = x_1, \qquad \forall (x, u) \in S \times U, \tag{3.3.7}$$

and

$$P(y \mid x, u) = 1, \qquad \forall (y, x, u) \in S \times S \times U. \tag{3.3.8}$$

It is easily seen that this instance satisfies Assumptions A.1-A2, A.4, A.6, and E.1 with $K = 1$ and $\rho = 1/2$.

Bellman's equation reads

$$J(x) = x_1 + \alpha \int_S J(y)\, dy. \tag{3.3.9}$$

A simple calculation shows that the function $J^*$ defined by

$$J^*(x) = x_1 + \frac{\alpha}{2(1-\alpha)} \tag{3.3.10}$$

is a solution of (3.3.9) and according to the discussion of Section 2.2, it is the unique fixed point of $T$.

Let $X$ be the set of points $(y, x, u)$ sampled by the particular algorithm we are considering, when it is faced with the instance $(P, g, \alpha, \epsilon)$. In particular, the cardinality of $X$ is at most $A$. Using the definition of $\delta$ [cf. Eqs. (3.3.4)-(3.3.5)], the cardinality of $X$ is at most $1/(4\delta^{2n+m})$.

We say that a cell $C_{\tilde{y},\tilde{x},\tilde{u}}$ is *sampled* if the intersection of $X$ and $C_{\tilde{y},\tilde{x},\tilde{u}}$ is nonempty. Otherwise, we say that $C_{\tilde{y},\tilde{x},\tilde{u}}$ is *unsampled.* We say that some $(\tilde{x}, \tilde{u}) \in \tilde{S} \times \tilde{U}$ is *well-sampled* if there exist at least $1/(2\delta^n)$ elements $\tilde{y}$ of $\tilde{S}$ for which the cell $C_{\tilde{y},\tilde{x},\tilde{u}}$ is sampled. Otherwise, we say that $(\tilde{x}, \tilde{u})$ is *badly sampled.* Since the total number of samples is bounded by $1/(4\delta^{2n+m})$, there exist at most $1/(2\delta^{n+m})$ well-sampled elements $(\tilde{x}, \tilde{u}) \in \tilde{S} \times \tilde{U}$. Therefore, there are at least $1/(2\delta^{n+m})$ badly sampled $(\tilde{x}, \tilde{u})$. For each $\tilde{x} \in \tilde{S}$ there are at most $1/\delta^m$ possible choices of $\tilde{u}$ such that $(\tilde{x}, \tilde{u})$ is badly sampled. This shows that there exists a set $\tilde{S}_{\text{BAD}} \subset \tilde{S}$ of cardinality $1/(2\delta^n)$ such that for each $\tilde{x} \in \tilde{S}_{\text{BAD}}$ there exists some $\tilde{\mu}(\tilde{x}) \in \tilde{U}$ for which $(\tilde{x}, \tilde{\mu}(\tilde{x}))$ is badly sampled.

We will now construct a second instance. The cost function $g$ is left unchanged [cf. Eq. (3.3.7)], but we modify the probability density on some of the unsampled cells. This is done as follows. Let us fix some $\tilde{x} \in \tilde{S}_{\text{BAD}}$. By the definition of $\tilde{S}_{\text{BAD}}$ and $\tilde{\mu}(\tilde{x})$, if we keep $\tilde{x}$ fixed and vary $\tilde{y}$, we find at least $1/(2\delta^n)$ unsampled cells of the form $C_{\tilde{y},\tilde{x},\tilde{\mu}(\tilde{x})}$. We sort these unsampled cells in order of increasing $\tilde{y}_1$ and we let $c$ be the median value of $\tilde{y}_1$. We refer to those cells for which $\tilde{y}_1 \le c$ (respectively, $\tilde{y}_1 \ge c$) as *low* (respectively, *high*) cells. Let $\underline{c} = c - (1/16)$ and $\overline{c} = c + (1/16)$. We discard all unsampled cells $C_{\tilde{y},\tilde{x},\tilde{\mu}(\tilde{x})}$ for which $\underline{c} < \tilde{y}_1 < \overline{c}$. Thus, the number of discarded cells is bounded by $1/(8\delta^n)$. Since we started with at least $1/(4\delta^n)$ low unsampled cells, we are left with at least $1/(8\delta^n)$ such cells. By discarding some more low unsampled cells (if needed), we can assume that we are left with exactly $1/(8\delta^n)$ unsampled low cells. By a similar argument, we can also assume that we are left with exactly $1/(8\delta^n)$ unsampled high cells. Let $Q^L(\tilde{x})$ [respectively, $Q^H(\tilde{x})$] be the set of all $\tilde{y} \in \tilde{S}$ such that $C_{\tilde{y},\tilde{x},\tilde{\mu}(\tilde{x})}$ is a low (respectively, high) unsampled cell that has not been discarded. This procedure is carried out for each $\tilde{x} \in \tilde{S}_{\text{BAD}}$.

We define

$$\hat{P}(y \mid x, u) = P(y \mid x, u) + E(y \mid x, u) = 1 + E(y \mid x, u), \tag{3.3.11}$$

where

$$E(y \mid x, u) = \sum_{\tilde{x} \in \tilde{S}_{\text{BAD}},\ \tilde{y} \in Q^L(\tilde{x})} E_{\tilde{y},\tilde{x},\tilde{\mu}(\tilde{x})}(y \mid x, u) - \sum_{\tilde{x} \in \tilde{S}_{\text{BAD}},\ \tilde{y} \in Q^H(\tilde{x})} E_{\tilde{y},\tilde{x},\tilde{\mu}(\tilde{x})}(y \mid x, u). \tag{3.3.12}$$

In words, we add a pyramid at each low unsampled cell and we subtract a pyramid at each high unsampled cell. This has the effect of shifting the transition probability distribution closer to the origin, with a consequent decrease in the cost incurred after a transition.

We verify that our perturbed instance $(\hat{P}, g, \alpha, \epsilon)$ satisfies the required assumptions. Since each pyramid is Lipschitz continuous with Lipschitz constant 1, and since distinct pyramids are supported on distinct cells, it follows that Assumption A.6 is satisfied with $K = 1$. Furthermore, for each $(x, u)$, the number of added pyramids is equal to the number of subtracted pyramids. For this reason, $\int_S E(y \mid x, u)\, dy = 0$ and $\hat{P}$ satisfies Assumption A.4. Finally, the height of each pyramid is $\delta/2$. Since $\delta \leq 1$, we have $\hat{P}(y \mid x, u) \geq 1 - (\delta/2) \geq 1/2$. This shows that Assumption A.4 and Assumption E.1 (with $\rho = 1/2$) are satisfied.

Our next task is to estimate the optimal cost function $\hat{J}^*$ corresponding to the perturbed instance $(\hat{P}, g, \alpha, \epsilon)$. Let

$$B = \Big\{x \in S \;\mid \exists \tilde{x} \in \tilde{S}_{\text{BAD}} \text{ such that } \|x - \tilde{x}\|_\infty \leq \frac{\delta}{4}\Big\}.$$

For any $x \in B$, we let $\mu(x) = \tilde{\mu}(\tilde{x})$ where $\tilde{x}$ is the element of $\tilde{S}_{\text{BAD}}$ for which $\|x - \tilde{x}\|_\infty \leq \delta/4$. For any $x \notin B$, we let $\mu(x) = 0$. We now consider the quantity

$$e(x) = \int_S g(y) E(y \mid x, \mu(x))\, dy \tag{3.3.13}$$

that can be interpreted as the effect of the perturbation on the expected cost after the first transition, when the control is chosen according to the function $\mu$.

**Lemma 3.3.1** *For each $x \in S$, we have $e(x) \leq 0$. Furthermore, there exists a positive absolute constant $\kappa$ such that $e(x) \leq -\kappa\delta$, for all $x \in B$.*

**Proof** Using Eqs. (3.3.12) and (3.3.13) and the definition of $g$, we have

$$\begin{aligned} e(x) = & \sum_{\tilde{x} \in \tilde{S}_{\text{BAD}},\ \tilde{y} \in Q^L(\tilde{x})} \int_S y_1 E_{\tilde{y}, \tilde{x}, \tilde{\mu}(\tilde{x})}(y \mid x, \mu(x))\, dy \\ & - \sum_{\tilde{x} \in \tilde{S}_{\text{BAD}},\ \tilde{y} \in Q^H(\tilde{x})} \int_S y_1 E_{\tilde{y}, \tilde{x}, \tilde{\mu}(\tilde{x})}(y\ idx, \mu(x))\, dy. \end{aligned} \tag{3.3.14}$$

For any $x \notin B$, we have $\mu(x) = 0$ which implies that $E(y \mid x, \mu(x)) = 0$ and $e(x) = 0$. Let us now fix some $x \in B$ and let $\tilde{x}$ be the corresponding element of $\tilde{S}_{\text{BAD}}$. Then, Eq. (3.3.14) becomes

$$e(x) = \sum_{\tilde{y} \in Q^L(\tilde{x})} \int_S y_1 E_{\tilde{y}, \tilde{x}, \tilde{\mu}(\tilde{x})}(y \mid x, \mu(x))\, dy - \sum_{\tilde{y} \in Q^H(\tilde{x})} \int_S y_1 E_{\tilde{y}, \tilde{x}, \tilde{\mu}(\tilde{x})}(y \mid x, \mu(x))\, dy. \tag{3.3.15}$$

Let us consider the summand corresponding to a particular $\tilde{y} \in Q^L(\tilde{x})$. We only need to carry out the integration on the set $Y(\tilde{y}) = \{y \in S \mid \|y - \tilde{y}\|_\infty \leq \delta/2\}$ (instead of the entire set $S$) because $E_{\tilde{y}, \tilde{x}, \tilde{\mu}(\tilde{x})}(y \mid x, u)$ vanishes when $y \notin Y(\tilde{y})$. For $y \in Y(\tilde{y})$, we have $y_1 \leq \tilde{y}_1 + \delta/2 \leq \underline{c} + \delta/2$ We now use the definition of the function $E_{\tilde{y}, \tilde{x}, \tilde{\mu}(\tilde{x})}(y \mid x, \mu(x))$ [cf. Eq. (3.3.6)], together with the property $\mu(x) = \tilde{\mu}(\tilde{x})$, to conclude that

$$\int_S y_1 E_{\tilde{y}, \tilde{x}, \tilde{\mu}(\tilde{x})}(y \mid x, \mu(x))\, dy \leq \Big(\underline{c} + \frac{\delta}{2}\Big) I(x),$$

where
$$I(x) = \int_{Y(\tilde{y})} \Big(\frac{\delta}{2} - \max\{\|x-\tilde{x}\|_\infty, \|y-\tilde{y}\|_\infty\}\Big)\, dy. \tag{3.3.16)}$$
It is clear that the value of $I(x)$ is independent of the choice of $\tilde{y}$, which justifies our notation. By a symmetrical argument, each one of the summands corresponding to $\tilde{y} \in Q^H(\tilde{x})$ is bounded below by $(\bar{c} - \delta/2)I(x)$. Since each one of the sets $Q^H(\tilde{x})$, $Q^L(\tilde{x})$ has cardinality $1/8\delta^n$, it follows from Eq. (3.3.14) that
$$e(x) \le -(\bar{c} - \underline{c} - \delta)I(x)\frac{1}{8\delta^n} \le -I(x)\frac{1}{64\delta^n}, \qquad \forall x \in B, \tag{3.3.17}$$
where the last inequality follows because $\bar{c} - \underline{c} = 1/4$ (by construction) and $\delta \le 1/16$ (by definition). We now bound $I(x)$ for $x \in B$. We have $\|x - \tilde{x}\|_\infty \le \delta/4$, the integrand is always nonnegative, and is at least $\delta/4$ for every $y$ belonging to the set $\{y \in S \mid \|y - \tilde{y}\|_\infty \le \delta/4\}$. Therefore, for $x \in B$, $I(x)$ is bounded below by $\delta/4$ times the volume of the set $\{y \in S \mid \|y - \tilde{y}\|_\infty \le \delta/4\}$. This set is an $n$-dimensional cube, whose edges have length $\delta/2$. Thus, we obtain
$$I(x) \ge \frac{\delta}{4}\frac{\delta^n}{2^n}, \qquad \forall x \in B.$$
Combining with Eq. (3.3.17), we obtain
$$e(x) \le -\frac{\delta}{4 \cdot 64 \cdot 2^n}, \qquad \forall x \in B,$$
which proves the desired result. □

**Lemma 3.3.2** *There exists a positive absolute constant $\kappa'$ such that*
$$\int_B \hat{P}(y \mid x, u)\, dy \ge \kappa', \qquad \forall x \in S,\ \forall u \in U.$$

**Proof** The function $\hat{P}$ is bounded below by 1/2, as shown earlier. Thus, it suffices to show that the volume of $B$ is bounded below by some absolute constant. Note that $B$ consists of $1/(2\delta^n)$ cubes of volume $(\delta/2)^n$, and the result follows. □

Let $\hat{T}$ be the dynamic programming operator associated with the instance $(\hat{P}, g)$. We have
$$\begin{aligned}
\hat{T}J^*(x) &= g(x) + \alpha \min_{u\in U}\int_S J^*(y)\hat{P}(y \mid x, u)\, dy \\
&\le g(x) + \alpha \int_S J^*(y)\hat{P}(y \mid x, \mu(x))\, dy \\
&= x_1 + \alpha\int_S J^*(y)\, dy + \alpha \int_S J^*(y)E(y \mid x, \mu(x))\, dy \\
&= J^*(x) + \alpha\int_S J^*(y)E(y \mid x, \mu(x))\, dy \\
&= J^*(x) + \alpha\int_S y_1 E(y \mid x, \mu(x))\, dy + \frac{\alpha^2}{2(1-\alpha)}\int_S E(y \mid x, \mu(x))\, dy \\
&= J^*(x) + \alpha e(x),
\end{aligned} \tag{3.3.18}$$

where we have used the fact that $J^*$ satisfies Eqs. (3.3.9) and (3.3.10), the definition of $e(x)$ [cf. Eq. (3.3.15)], and the fact that $\int_S E(y \mid x, \mu(x))\, dy = 0$. It follows that

$$\hat{T}J^*(x) \le J^*(x), \qquad \forall x \in S, \tag{3.3.19}$$

$$\hat{T}J^*(x) \le J^*(x) - \alpha\kappa\delta, \qquad \forall x \in B, \tag{3.3.20}$$

where $\kappa$ is the constant of Lemma 3.3.1. Let $\hat{T}^t$ be the composition of $t$ copies of $\hat{T}$ and let $\overline{B} = \{x \in S \mid x \notin B\}$ be the complement of $B$. We have

$$\begin{aligned}
\hat{T}^2 J^*(x) &= g(x) + \alpha \min_{u \in U} \int_S \hat{T}J^*(y)\hat{P}(y \mid x, u)\, dy \\
&\le g(x) + \alpha \int_S \hat{T}J^*(y)\hat{P}(y \mid x, \mu(x))\, dy \\
&= g(x) + \alpha \int_B \hat{T}J^*(y)\hat{P}(y \mid x, \mu(x))\, dy + \alpha \int_{\overline{B}} \hat{T}J^*(y)\hat{P}(y \mid x, \mu(x))\, dy \\
&\le g(x) + \alpha \int_B (J^*(y) - \alpha\kappa\delta)\hat{P}(y \mid x, \mu(x))\, dy + \alpha \int_{\overline{B}} J^*(y)\hat{P}(y \mid x, \mu(x))\, dy \\
&= J^*(x) + \alpha e(x) - \alpha^2\kappa\delta \int_B \hat{P}(y \mid x, \mu(x))\, dy \\
&\le J^*(x) - \alpha^2\kappa\delta\kappa', \qquad \forall x \in S.
\end{aligned} \tag{3.3.21}$$

[We have used here the equality between the second and the last line of Eq. (3.3.18), as well as Lemma 3.3.2.] By Proposition 2.2.1(c), we have $\hat{T}(J + d) = \alpha d + \hat{T}J$. (Here the notation $J + d$ should be interpreted as the function which is equal to the sum of $J$ with a function on $S$ that is identically equal to $d$.) Using this property and Eq. (3.3.21), we obtain

$$\hat{T}^3 J^*(x) \le \hat{T}^2 J^*(x) - \alpha^3\kappa\delta\kappa' \le J^*(x) - \alpha^2\kappa\delta\kappa' - \alpha^3\kappa\delta\kappa', \qquad \forall x \in S.$$

We continue inductively, to obtain

$$\hat{T}^t J^*(x) \le J^*(x) - (1 + \alpha + \alpha^2 + \cdots + \alpha^{t-2})\alpha^2\kappa\delta\kappa', \qquad t = 2, 3, \ldots, \ \forall x \in S. \tag{3.3.22}$$

Taking the limit as $t \to \infty$, $\hat{T}^t J^*$ converges to the optimal cost function $\hat{J}^*$ of the perturbed instance, and Eq. (3.3.22) implies that

$$\hat{J}^*(x) \le J^*(x) - \frac{\alpha^2}{1 - \alpha}\delta\kappa\kappa', \qquad \forall x \in S. \tag{3.3.23}$$

Note that the perturbed instance coincides with the original one at all points sampled by the algorithm. For this reason, the algorithm will perform the same arithmetic operations and will return the same answer for both instances. That answer must be an $\epsilon$–approximation of both $J^*$ and $\hat{J}^*$. It follows that $\|J^* - \hat{J}^*\|_\infty \le 2\epsilon$. Therefore,

$$\frac{\alpha^2}{1 - \alpha}\delta\kappa\kappa' \le 2\epsilon.$$

Since $\alpha \geq 1/2$, we obtain

$$\delta \leq d(1-\alpha)\epsilon, \tag{3.3.24}$$

where $d$ is some absolute constant.

For $\epsilon \leq 1/(32d)$, we obtain $\delta \leq 1/32$ or $1/(2\delta) \geq 16$. Thus, using Eq. (3.3.4), we have $1/\delta_0 \geq (1/\delta) - 16 \geq (1/\delta) - (1/2\delta) = 1/(2\delta)$, and Eq. (3.3.5) yields

$$A = \frac{1}{4\delta_0^{2n+m}} \geq \frac{1}{4(2\delta)^{2n+m}} \geq \frac{1}{4(2d(1-\alpha)\epsilon)^{2n+m}} = \Omega\left(\frac{1}{((1-\alpha)\epsilon)^{2n+m}}\right).$$

When the theorem is proved for general values of $K$ and $\rho$, it is sufficient to multiply the pyramidal functions of Eq. (3.3.6) by a factor of $\min\{K, 1-\rho\}$. It is then easily seen that the perturbed problem satisfies Assumptions A.1-A.2, A.4, A.6, and E.1 for the given values of $K$ and $\rho$ and the proof goes through verbatim, except that certain absolute constants are modified. $\square$

In our next result, the ergodicity condition, Assumption E.1, is removed. It will be seen that this allows us to obtain a larger lower bound.

**Theorem 3.3.2** *(Lower bound under Assumption A.4) For every $m$, $n$, there exists some $K$ such that*

$$\mathbf{C}_{\text{std}}^{\text{info}}(\epsilon,\alpha) = \Omega\left(\frac{1}{((1-\alpha)^2\epsilon)^{2n+m}}\right);$$

*in particular,*

$$\mathbf{C}_{\text{std}}(\epsilon,\alpha) = \Omega\left(\frac{1}{((1-\alpha)^2\epsilon)^{2n+m}}\right).$$

**Proof** The structure of the proof is similar to the preceding one. We fix $n$, $m$, and some $K$ that will depend on $n$ in a way to be determined later. An absolute constant is again a constant that depends only on $m$ and $n$.

We fix some $\epsilon > 0$ and some $\alpha \in (1/2, 1)$. We consider an algorithm that is correct for the problem $\mathcal{P}$ (for the given values of $m$, $n$, $K$) and suppose that the number of queries is at most $A$ for every instance with those particular values of $\alpha$ and $\epsilon$.

We choose a positive scalar $\delta$ so that $1/\delta$ is an integer multiple of 9 and such that

$$\frac{1}{\delta_0} \leq \frac{1}{\delta} \leq \frac{1}{\delta_0} + 9, \tag{3.3.25}$$

where $\delta_0$ satisfies

$$A = \frac{1}{5}\left(\frac{2}{9}\right)^{2n} \frac{1}{\delta_0^{2n+m}}. \tag{3.3.26}$$

We partition $S \times S \times U$ into cubic cells of volume $1/\delta^{2n+m}$ exactly as in the proof of Theorem 3.3.1 and we use the same notations $\tilde{S}$, $\tilde{U}$, $C_{\tilde{y},\tilde{x},\tilde{u}}$, and $E_{\tilde{y},\tilde{x},\tilde{u}}$.

We define the first instance to be considered. Let $F_1, F_2, G : [0,1] \mapsto \mathbf{R}$ be the functions shown in Figure 3.3.1. We define a function $H : [0,1] \times [0,1] \mapsto \mathbf{R}$ by letting

$$H(y \mid x) = F_1(y)G(x) + F_2(y)(1 - G(x)), \qquad \forall x, y \in [0,1].$$

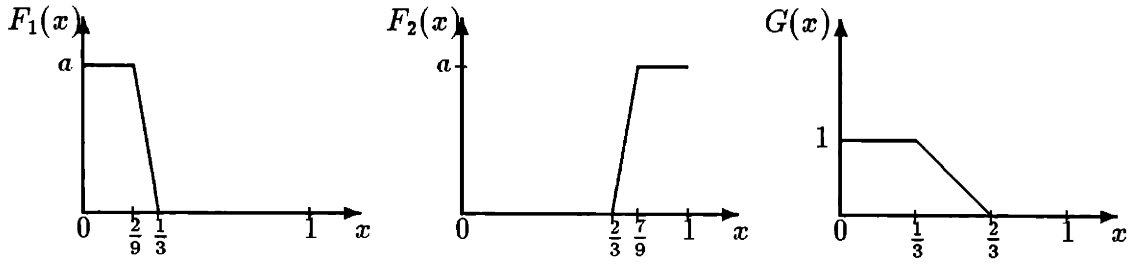

**Figure** 3.3.1: Functions used in the lower bound proof of $\mathcal{P}_{\text{std}}$
The functions $F_1$, $F_2$, and $G$. The maximum value $a$ of $F_1$ and $F_2$ is chosen so that $\int_0^1 F_1(x)\,dx = \int_0^1 F_2(x)\,dx = 1$. In particular, $3 \le a \le 9/2$.

We finally let

$$P(y \mid x, u) = \prod_{i=1}^{n} H(y_i \mid x_i), \qquad \forall (y, x, u) \in S \times S \times U, \tag{3.3.27}$$

where $x_i$ and $y_i$ is the $i$th component of $x$ and $y$, respectively. As for the cost function $g$, we only assume that $g(x,u) = 1$ for all $x \in [0,1/3]^n$ and $u \in U$, and that $g(x,u) = 0$ for all $x \in [2/3,1]^n$ and $u \in U$.

We verify that Assumptions A.1–A.2, A.4, and A.6 are satisfied. The function $P$ is certainly nonnegative. Furthermore, $F_1$ and $F_2$ integrate to 1. Consequently, $\int_{[0,1]} H(y \mid x)\,dy = 1$, for all $x \in [0,1]$. Thus, for any $x$, $u$, $P(\cdot \mid x,u)$ is a product of probability measures [cf. Eq. (3.3.27)] and is itself a probability measure. Note that $F_1$, $F_2$ and $G$ are Lipschitz continuous. It follows that $P$ is also Lipschitz continuous with Lipschitz constant $K$, provided that the absolute constant $K$ is taken large enough. Concerning the function $g$, we have not specified it in detail, but it is easily seen that there exist Lipschitz continuous functions satisfying the requirements we have imposed on $g$.

Note that the Markov chain corresponding to $P$ has the property that if the current state is in the set $[0,1/3]^n$ then the state stays forever in that set. The same property holds for the set $[2/3,1]^n$.

We now estimate $J^*(x)$ when $x \in [0,1/3]^n$. While we could argue directly in terms of the Bellman's equation, the argument is much more transparent if we use the interpretation of $J^*(x)$ as the optimal cost expressed as a function of the initial state. Starting with some initial state in $[0,1/3]^n$, the state never exits that set. Furthermore, $g(x) = 1$ for every $x \in [0,1/3]^n$. This implies that

$$J^*(x) = \sum_{t=0}^{\infty} \alpha^t = \frac{1}{1-\alpha}, \qquad \forall x \in [0,1/3]^n. \tag{3.3.28}$$

**Lemma 3.3.3** *There exists a set $\tilde{S}_{\text{BAD}} \subset [0,2/9]^n \cap \tilde{S}$ of cardinality $(2/9)^n/(2\delta^n)$ with the following property: for every $\tilde{x} \in \tilde{S}_{\text{BAD}}$ there exists some $\tilde{\mu}(\tilde{x}) \in \tilde{U}$ and two sets $Q^L(\tilde{x}) \subset [0,2/9]^n \cap \tilde{S}$, $Q^H(\tilde{x}) \subset [7/9,1]^n \cap \tilde{S}$, of cardinality $(2/9)^n/(2\delta^n)$, such that the cell $C_{\tilde{y},\tilde{x},\tilde{\mu}(\tilde{x})}$ is unsampled for every $\tilde{y} \in Q^L(\tilde{x}) \cup Q^H(\tilde{x})$.*

**Proof** Let $\tilde{S}_{GOOD}$ be the set of all $\tilde{x} \in [0, 2/9]^n \cap \tilde{S}$ that do not have the desired property. Since the cardinality of $[0, 2/9]^n \cap \tilde{S}$ is $(2/9)^n/\delta^n$ it is sufficient to show that $\tilde{S}_{GOOD}$ has cardinality less than or equal to $(2/9)^n/(2\delta^n)$. We suppose the contrary, and we will obtain a contradiction.

Fix some $\tilde{x} \in \tilde{S}_{GOOD}$. Then, for every $\tilde{u} \in \tilde{U}$ we can find at least $(2/9)^n/(2\delta^n)$ values of $\tilde{y} \in \tilde{S}$ such that the cell $C_{\tilde{y},\tilde{x},\tilde{u}}$ is sampled. This shows that the total number of sampled cells is at least $(2/9)^{2n}/(4\delta^{2n+m})$. Using Eqs. (3.3.25) and (3.3.26), this implies that the number of sampled cells is more than $A$, a contradiction. □

We now construct a perturbed instance. The cost function $g$ is left unchanged. We define

$$\hat{P}(y \mid x, u) = P(y \mid x, u) + E(y \mid x, u),$$

where

$$E(y \mid x, u) = \sum_{\tilde{x} \in \tilde{S}_{\mathrm{BAD}},\ \tilde{y} \in Q^H(\tilde{x})} E_{\tilde{y},\tilde{x},\tilde{\mu}(\tilde{x})}(y \mid x, u) - \sum_{\tilde{x} \in \tilde{S}_{\mathrm{BAD}},\ \tilde{y} \in Q^L(\tilde{x})} E_{\tilde{y},\tilde{x},\tilde{\mu}(\tilde{x})}(y \mid x, u). \tag{3.3.29}$$

In effect, we are giving positive probability to certain transitions from the set $[0, 2/9]^n$ to the set $[7/9, 1]^n$. On the other hand, the property that the state can never exit from the set $[7/9, 1]^n$ is retained. The Lipschitz continuity of $E$ and $P$ implies that $\hat{P}$ is Lipschitz continuous. Also $\hat{P}(\cdot \mid x, u)$ is nonnegative and integrates to 1, for reasons similar to those in the proof of Theorem 3.3.1 Thus, Assumptions A.1-A.2, A.4, and A.6 are satisfied.

Let

$$B = \Big\{x \in [0, 1/3]^n \mid \exists \tilde{x} \in \tilde{S}_{\mathrm{BAD}} \text{ such that } \|x - \tilde{x}\|_\infty \le \frac{\delta}{4}\Big\}.$$

**Lemma 3.3.4** *For every $x \in [0, 1/3]^n$, we have*

$$\int_B \hat{P}(y \mid x, u)\, dy \ge \kappa',$$

*where $\kappa'$ is a positive absolute constant.*

**Proof** Fix some $x \in [0, 1/3]^n$. Note that $P(y \mid x, u) = \prod_{i=1}^n F_1(y_i) \ge 3^n \ge 3$ for all $y \in [0, 2/9]^n$. Since $|E(y \mid x, u)| \le \delta/2 \le 1$, we conclude that $\hat{P}(y \mid x, u) \ge 2$, for all $y \in [0, 2/9]^n$. The set $B$ consists of $(2/9\delta)^n/2$ cubes of volume $(\delta/2)^n$. Thus, the volume of $B$ is bounded below by some absolute positive constant, and the result follows. □

Let us now define $\mu(x) = \tilde{\mu}(\tilde{x})$ for all $\tilde{x} \in B$, where $\tilde{x}$ is chosen so that $\|x - \tilde{x}\|_\infty \le \delta/4$, and we let $\mu(x) = 0$ for $x \notin B$.

**Lemma 3.3.5** *For every $x \in B$, we have*

$$\int_{[2/3,1]^n} \hat{P}(y \mid x, \mu(x))\, dy \ge \kappa\delta,$$

*where $\kappa$ is an absolute positive constant.*

**Proof** For every $\tilde{y} \in \tilde{S}$, let $Y(\tilde{y}) = \{y \in S \mid \|y - \tilde{y}\|_\infty \leq \delta/4\}$. Fix some $x \in B$ and let $\tilde{x}$ be an element of $\tilde{S}_{\text{BAD}}$ such that $\|x - \tilde{x}\|_\infty \leq \delta/4$. We have

$$\begin{aligned}
\int_{[2/3,1]^n} \hat{P}(y \mid x, \mu(x))\, dy &\geq \sum_{\tilde{y} \in Q^H(\tilde{x})} \int_{Y(\tilde{y})} E_{\tilde{y},\tilde{x},\tilde{\mu}(\tilde{x})}(y \mid x, \mu(x))\, dy \\
&\geq \sum_{\tilde{y} \in Q^H(\tilde{x})} \int_{Y(\tilde{y})} \Big(\frac{\delta}{2} - \max\{\|x - \tilde{x}\|_\infty, \|y - \tilde{y}\|_\infty\}\Big)\, dy \\
&\geq \sum_{\tilde{y} \in Q^H(\tilde{x})} \frac{\delta}{4} \int_{Y(\tilde{y})} dy.
\end{aligned}$$

The set $Y(\tilde{y})$ is a cube of volume $(\delta/2)^n$, the cardinality of the set $Q^H(\tilde{x})$ is $(2/9\delta)^n/2$, and the result follows.

□

We now estimate the cost $J_\pi(x)$ which is incurred if policy $\pi = (\mu, \mu, \ldots)$ is used, for the case where $x \in [0, 1/3]^n$. The corresponding Markov process $x_t^\pi$ evolves as follows. Whenever $x_t^\pi \in [0, 1/3]^n$, there is at least probability $\kappa'$ that the next state belongs to the set $B$ and there is a further probability of at least $\kappa\delta$ that the state after one more transition is in the set $[2/3, 1]^n$. Once the latter set is entered, the state stays forever in that set. We therefore have

$$\Pr(x_t^\pi \in [0, 1/3]^n) \leq (1 - \kappa\kappa'\delta)^{t-1}, \qquad \forall t \geq 1.$$

Since the cost is 1 on the set $[0, 1/3]^n$ and 0 on the set $[2/3, 1]^n$, we have

$$\begin{aligned}
J_\pi(x) &= \sum_{t=0}^{\infty} \alpha^t \Pr(x^\pi(t) \in [0, 1/3]^n) \\
&\leq 1 + \sum_{t=1}^{\infty} \alpha^t (1 - \kappa\kappa'\delta)^{t-1} \\
&= 1 + \frac{\alpha}{1 - \alpha(1 - \kappa\kappa'\delta)} \qquad (3.3.30) \\
&= \frac{1 + \alpha\kappa\kappa'\delta}{1 - \alpha(1 - \kappa\kappa'\delta)}, \qquad \forall x \in [0, 1/3]^n.
\end{aligned}$$

The optimal cost function $\hat{J}^*$ of the perturbed instance satisfies $\hat{J}^* \leq \hat{J}_\pi$ and, using Eq. (3.3.28), we obtain

$$\begin{aligned}
J^*(x) - \hat{J}^*(x) &\geq \frac{1}{1 - \alpha} - \frac{1 + \alpha\kappa\kappa'\delta}{1 - \alpha(1 - \kappa\kappa'\delta)} \\
&= \frac{\alpha^2 \kappa\kappa'\delta}{(1 - \alpha)(1 - \alpha(1 - \kappa\kappa'\delta))}, \qquad \forall x \in [0, 1/3]^n. \qquad (3.3.31)
\end{aligned}$$

Note that the class $\mathcal{P}_{\text{std}}$ contains the class $\mathcal{P}_{\text{mix1}}$. For this reason, the particular algorithm being considered here is also a correct algorithm for the problem $\mathcal{P}_{\text{mix1}}$. In particular, all of the intermediate results in the proof of Theorem 3.3.1 apply to the algorithm we are considering. We can therefore use Eq. (3.3.25) and conclude that $\delta \leq d(1-\alpha)\epsilon$, where $d$ is an absolute constant. (Actually, the definition of $\delta$ is somewhat different in the two proofs, but this only affects the absolute constant $d$.) This implies that for $\epsilon \leq 1/(\kappa\kappa' d)$, we have $\delta \leq (1-\alpha)/(\kappa\kappa')$ or $1-\kappa\kappa'\delta \geq \alpha$. Using this inequality in Eq. (3.3.31), together with the property $\alpha \geq 1/2$, we obtain

$$J^*(x) - \hat{J}^*(x) \geq \frac{\alpha^2\kappa\kappa'\delta}{(1-\alpha)(1-\alpha^2)} \geq \frac{(\kappa\kappa'\delta)/4}{(1-\alpha)^2 2} = \frac{1}{8}\frac{\kappa\kappa'\delta}{(1-\alpha)^2}.$$

This inequality is similar to inequality (3.3.25) in the proof of Theorem 3.3.1, except that $1-\alpha$ has been replaced by $(1-\alpha)^2$. The rest of the argument is the same, except for certain constant factors, and that $1-\alpha$ is replaced throughout by $(1-\alpha)^2$. □

**Theorem 3.3.3** *(Lower bound under Assumptions A.4′ and E.1) For every $m$, $n$, there is a choice of $K$ and $\rho$ such that*

$$\mathbf{C}^{\text{info}}_{\text{sub1}}(\epsilon,\alpha) = \Omega\left(\frac{1}{((1-\alpha)^2\epsilon)^{2n+m}}\right);$$

*in particular,*

$$\mathbf{C}_{\text{sub1}}(\epsilon,\alpha) = \Omega\left(\frac{1}{((1-\alpha)^2\epsilon)^{2n+m}}\right).$$

**Proof** The proof is almost identical to the proof of Theorem 3.3.2, and for this reason, we argue informally. For convenience, let the state space $S$ be the set $[0,1/3]^n$, instead of $[0,1]^n$, and let $P(y \mid x,u)$ be defined on that set as in proof of Theorem 3.3.2. Then, $P$ is a probability measure on the set $[0,1/3]^n$ and the corresponding function $J^*$ is identically equal to $1/(1-\alpha)$. Note that $P$ satisfies Assumption E.1. Let $\hat{P}$ be as in the proof of Theorem 3.3.2, except that it is defined only for $x,y \in [0,1/3]^n$. For this reason, $\hat{P}$ is now a subprobability measure. The function $\hat{J}^*$ for the current problem is equal to the optimal expected discounted cost until the termination of the stochastic process. However, the process considered here terminates exactly when the process considered in the proof of Theorem 3.3.2 makes a transition from $[0,1/3]^n$ to the zero–cost set $[2/3,1]^n$. For this reason, the function $\hat{J}^*$ is the same as the function $\hat{J}^*$ in the proof of Theorem 3.3.2, and the result follows with the same reasoning. □

**Remarks:**

1. Suppose that we replace the correctness requirement $\|J-J^*\|_{\infty} \leq \epsilon$ by the requirement $\|J-J^*\|_p \leq \epsilon$, where $1 \leq p < \infty$ and $\|\cdot\|_p$ is the usual $L_p$-norm. Then, Theorems 3.3.1–3.3.3 remain true, with exactly the same proofs. The reason, is that in all of our proofs we have constructed our perturbed instances so that $J^*(x) - \hat{J}^*(x)$ is "large" on a set whose measure is bounded below by an absolute constant [cf. Eq. (3.3.23) or Eq. (3.3.31)]. But this implies that $J^* - \hat{J}^*$ is also large when measured by the $L_p$–norm and the proofs remain valid, except that certain constants have to be changed.

2. The lower bounds of Theorem 3.3.3 can also be proved for all values of the constants $K$ and $\rho$. The proof is similar except that we should let $P(y|x,u) = 3^n$ for all $(y,x,u) \in S \times S \times U$, so that $P$ satisfies the Lipschitz continuity assumption for any value of $K$. Furthermore, the perturbing pyramids should be multiplied by a factor that ensures that their Lipschitz constant is less than $K$ and that Assumption E.1 is not violated.

3. We are not able to establish the lower bound of Theorem 3.3.2 for an arbitrary choice of $K$. There is a simple reason for that: if $K$ is taken very small, then Assumption E.1 is automatically satisfied and the best provable lower bound is the one in Theorem 3.3.1.

### 3.3.5 Optimality of Multigrid Successive Approximation

The lower bounds of Section 3.3.4 agree with the upper bounds of Section 3.2.3. Thus, we have completely characterized the information-based complexity of the approximate computation of $J^*$. This leaves the further question of evaluating the total complexity of approximating $J^*$, when arithmetic computations are taken into account (the total complexity).

Recall from Section 3.2 that by using a multigrid version of the iterative algorithm $J := TJ$ we have shown that the total number of arithmetic operations and comparisons for $\mathcal{P}_{\text{gen}}$ and $\mathcal{P}_{\text{mix}}$ are, respectively, given by

$$\mathbf{C}_{\text{gen}} = \mathrm{O}\left(\frac{1}{|\log\alpha|}\cdot\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right)$$

$$= \mathrm{O}\left(\frac{1}{1-\alpha}\cdot\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right); \tag{3.3.32}$$

$$\mathbf{C}_{\text{mix}} = \mathrm{O}\left(\left[\frac{1}{(1-\alpha)\epsilon}\right]^{2n+m}\right). \tag{3.3.33}$$

Thus, for problem $\mathcal{P}_{\text{mix}}$, we have an optimal algorithm. For the problems $\mathcal{P}_{\text{gen}}$,$\mathcal{P}_{\text{subl}}$, and $\mathcal{P}_{\text{std}}$, multigrid successive approximation is within a factor of $\mathrm{O}\left(\frac{1}{1-\alpha}\right)$ from the optimum. One might wish to close this gap but the prospects are not particularly bright because (i) there are no effective methods for proving lower bounds tighter than those provided by the information-based approach, and (ii) it will be shown in Chapter 5 that no algorithm in a certain family of multigrid methods can have complexity better than the one provided by Eq. (3.3.32).

Finally, we expect that our results can be extended to the cases where bounds are imposed on second derivatives (more generally, derivatives of order $r$) of the functions $P$ and $g$. Of course, the bounds should change, with the exponent $2n + m$ being replaced by a lower exponent, depending on $r$.

## 3.4 Computing $\epsilon$-Optimal Policies

In this section, we consider the computation of an *$\epsilon$-optimal policy,* that is, a stationary policy whose expected cost is within $\epsilon$ of the optimal. The main result of this section is that the upper bounds of Section 3.2 and the lower bounds of Section 3.3 are applicable to this problem as well; furthermore, computing an $\epsilon$-optimal policy is "as hard as" computing an $\epsilon$-optimal cost function (that is, the cost of computing the former is within a constant factor of the cost of computing the latter, and vice versa).

### 3.4.1 A Definition of $\epsilon$-Optimal Policies

Given a value of the discretization parameter $h$, we consider the set $\tilde{\Pi}_h$ of all policies at grid-level $h$ [see Eq. (2.4.8)]. These policies are easy to deal with computationally because they are simple functions on $\mathcal{S}_h$. Recall that if $\tilde{\mu} \in \tilde{\Pi}_h$, we must have $\tilde{\mu}(x) \in \tilde{U}_h(x)$ for all $x \in S$. However, this does not always imply that $\tilde{\mu}(x) \in U(x)$; that is, we have $\tilde{\mu} \notin \Pi$, in general.

To each $\tilde{\mu} \in \tilde{\Pi}_h$ we associate the operator $T_{\tilde{\mu}} : \mathcal{B}(S) \mapsto \mathcal{B}(S)$ defined by

$$T_{\tilde{\mu}}J(x) \overset{\text{def}}{=} g(x, \tilde{\mu}(x)) + \alpha \int_S J(y)P(y|x, \tilde{\mu}(x))\, dy. \tag{3.4.1}$$

[Note that, if $\tilde{\mu} \in \Pi$, this definition is consistent with our earlier definition of $T_{\tilde{\mu}}$; see Eq. (2.2.8)]. We also associate to $\tilde{\mu}$ the operator $\tilde{T}_{\tilde{\mu}} : \mathcal{B}(S) \mapsto \mathcal{B}(S)$ defined by

$$\tilde{T}_{\tilde{\mu}}J(x) \overset{\text{def}}{=} \tilde{g}_h(x, \tilde{\mu}(x)) + \alpha \int_S J(y)\tilde{P}_h(y|x, \tilde{\mu}(x))\, dy. \tag{3.4.2}$$

Similarly to $T_\mu$, $T_{\tilde{\mu}}$ and $\tilde{T}_{\tilde{\mu}}$ are monotone contraction operators and satisfy Proposition 2.2.1(c). (Under the more general Assumption A.4$''$, $T_{\tilde{\mu}}$ and $\tilde{T}_{\tilde{\mu}}$ remain contraction operators.) Let $J_{\tilde{\mu}}$ and $\tilde{J}_{\tilde{\mu}}$ be the fixed points of $T_{\tilde{\mu}}$ and $\tilde{T}_{\tilde{\mu}}$, respectively. Note that $J_{\tilde{\mu}}$ (respectively, $\tilde{J}_{\tilde{\mu}}$) can be interpreted as the expected cost function associated with stationary policy $\tilde{\mu}$ for the original MDP (respectively, for the discretized MDP).

Finally, let $\epsilon > 0$. A function $\tilde{\mu} : S \mapsto C$ is called an *$\epsilon$-optimal policy* if there exists some $h > 0$ such that $\tilde{\mu} \in \tilde{\Pi}_h$, $\|J_{\tilde{\mu}} - J^*\|_\infty \leq \epsilon$, and $\|\tilde{J}_{\tilde{\mu}} - J^*\|_\infty \leq \epsilon$.

### 3.4.2 Upper Bounds for Computing $\epsilon$-Optimal Policies

We now proceed to analyze the complexity of computing an $\epsilon$-optimal policy. We will show that computing an $\epsilon$-optimal policy is "no harder than" (within a constant factor in cost of) computing an $\epsilon$-optimal cost function; thus, the upper bounds of Theorem 3.2.2 [Eqs. (3.2.15) and (3.2.16)] apply to the computation of an $\epsilon$-optimal policy as well. To show this, we use the well known fact that the policy used in the final iteration of successive approximation algorithm is basically an $\epsilon$-optimal policy. The proof of this result depends on the following lemma (which is stated for the general case when Assumption A.4$''$ is in effect):

**Lemma 3.4.1** *(Under Assumption A.4″) Let $\hat{J}$ be an element of $\mathcal{B}(S)$. Suppose that $\tilde{\mu} \in \tilde{\Pi}_h$ is a policy that attains the minimum in the formula for $\tilde{T}_h\hat{J}$, that is,*

$$\tilde{T}_h\hat{J}(x) = \tilde{g}_h(x, \tilde{\mu}(x)) + \alpha \int_S \hat{J}(y)\tilde{P}_h(y|x, \tilde{\mu}(x))dy, \quad \forall x \in S.$$

*(Equivalently, $\tilde{T}_h\hat{J} = \tilde{T}_{\tilde{\mu}}\hat{J}$.) Then for all $h \in (0, h_a]$ there hold:*

**(a)** $\|\tilde{J}_h^* - \tilde{J}_{\tilde{\mu}}\|_\infty \leq \frac{2\alpha}{1-\alpha}\|\tilde{T}_h\hat{J} - \hat{J}\|_\infty$,

**(b)** $\|\tilde{J}_{\tilde{\mu}} - J_{\tilde{\mu}}\|_\infty \leq \frac{1}{1-\alpha}(K_1 + \alpha K_2\|J_{\tilde{\mu}}\|_\infty)h$,

*where $h_a$ is the constant of the discretization conditions D.1 and D.2 (see Section 2.6) and $K_1$ and $K_2$ are the constants of Theorem 2.4.2*

**Proof** To prove (a), since $\tilde{T}_h\hat{J} = \tilde{T}_{\tilde{\mu}}\hat{J}$, we have

$$\begin{aligned}\|\tilde{J}_h^* - \tilde{J}_{\tilde{\mu}}\|_\infty &\leq \|\tilde{J}_h^* - \tilde{T}_h\hat{J}\|_\infty + \|\tilde{T}_{\tilde{\mu}}\hat{J} - \tilde{J}_{\tilde{\mu}}\|_\infty \\ &\leq \frac{\alpha}{1-\alpha}\|\tilde{T}_h\hat{J} - \hat{J}\|_\infty + \frac{\alpha}{1-\alpha}\|\tilde{T}_{\tilde{\mu}}\hat{J} - \hat{J}\|_\infty \\ &= 2\frac{\alpha}{1-\alpha}\|\tilde{T}_h\hat{J} - \hat{J}\|_\infty,\end{aligned}$$

where we have used Eq. (3.1.2) to obtain the second inequality.

To prove (b), it is clear from the definition of $\tilde{T}_{\tilde{\mu}}$ and $T_{\tilde{\mu}}$ (and the proof of Theorem 2.4.2) that

$$\|\tilde{T}_{\tilde{\mu}}J - T_{\tilde{\mu}}J\|_\infty \leq (K_1 + \alpha K_2\|J\|_\infty)h, \quad \forall J \in \mathcal{B}(S).$$

So, using Lemma 2.4.3 we obtain

$$\|\tilde{J}_{\tilde{\mu}} - J_{\tilde{\mu}}\|_\infty \leq \frac{1}{1-\alpha}(K_1 + \alpha K_2\|J_{\tilde{\mu}}\|_\infty)h,$$

as required. □

We note that if Assumption A.4 is in effect, then for any scalar $c$, we have $\tilde{T}_h(\hat{J}+c) = \tilde{T}_{\tilde{\mu}}(\hat{J}+c)$, that is, $\tilde{\mu}$ is still a minimizing policy if the function $\hat{J}$ is shifted by any constant. So,

$$\|\tilde{J}_h^* - \tilde{J}_{\tilde{\mu}}\|_\infty \leq 2\frac{\alpha}{1-\alpha}\|\tilde{T}_h(\hat{J}+c) - (\hat{J}+c)\|_\infty = 2\frac{\alpha}{1-\alpha}\|\tilde{T}_h\hat{J} - \hat{J} - (1-\alpha)c\|_\infty.$$

By letting $c = \frac{1}{2(1-\alpha)}\left[\sup_x(\tilde{T}_h\hat{J} - \hat{J})(x) + \inf_x(\tilde{T}_h\hat{J} - \hat{J})(x)\right]$, Lemma 3.4.1(a) can be strengthened to yield

$$\|\tilde{J}_h^* - \tilde{J}_{\tilde{\mu}}\|_\infty \leq \frac{\alpha}{1-\alpha}\|\tilde{T}_h\hat{J} - \hat{J}\|_s.$$

[Actually, under Assumption A.4 (or A.4′) and using a different proof, it can be shown (see Section 4.2) that $\|\tilde{J}_h^* - \tilde{J}_{\tilde{\mu}}\|_\infty \leq \|\tilde{J}_h^* - \tilde{T}_h\hat{J}\|_\infty$, which implies the above bound.]

Moreover, under Assumption A.4, Lemma 3.4.1(b) can be strengthened to yield

$$\|\tilde{J}_{\tilde{\mu}} - J_{\tilde{\mu}}\|_\infty \leq \frac{1}{1-\alpha}(K_1 + \frac{\alpha}{2}K_2\|J_{\tilde{\mu}}\|_s)h.$$

To use Lemma 3.4.1, suppose that we compute an $\epsilon$-optimal cost function for the general case, using the multigrid successive approximation algorithm of Section 3.2.2. Let $\hat{J} = \tilde{T}_{h_f}^{t(h_f)-1} J_{h_f}^I$, so that $\tilde{T}_{h_f}\hat{J}$ corresponds to the last successive approximation iteration [cf. Eq. (3.2.8)]. Let $\tilde{\mu}$ be a policy that attains the minimum in $\tilde{T}_{h_f}\hat{J}$. Then by Lemma 3.4.1(a),

$$\|\tilde{J}_{h_f}^* - \tilde{J}_{\tilde{\mu}}\|_\infty \leq \frac{\alpha}{1-\alpha}\|\tilde{T}_{h_f}\hat{J} - \hat{J}\|_\infty \leq \epsilon, \tag{3.4.3}$$

where the last inequality follows from Eq. (3.2.8). Furthermore, since $\|J_{\tilde{\mu}}\|_\infty \leq \frac{K}{1-\alpha}$ we see from Theorem 2.4.2 and Eq. (3.2.7) that

$$\frac{1}{1-\alpha}(K_1 + \alpha K_2\|J_{\tilde{\mu}}\|_\infty)h_f \leq \frac{K'}{(1-\alpha)^2}h_f \leq \frac{\epsilon}{2}. \tag{3.4.4}$$

By Lemma 3.4.1(b),

$$\|\tilde{J}_{\tilde{\mu}} - J_{\tilde{\mu}}\|_\infty \leq \frac{\epsilon}{2}. \tag{3.4.5}$$

Lastly, the choice of $h_f$ [cf. Eq. (3.4.4)] ensures that the discretization error

$$\|J^* - \tilde{J}_{h_f}^*\|_\infty \leq \frac{\epsilon}{2}. \tag{3.4.6}$$

Using the triangle inequality and Eqs. (3.4.3), (3.4.5), (3.4.6), we conclude that

$$\|J^* - \tilde{J}_{\tilde{\mu}}\|_\infty \leq \|J^* - \tilde{J}_{h_f}^*\|_\infty + \|\tilde{J}_{h_f}^* - \tilde{J}_{\tilde{\mu}}\|_\infty \leq \frac{3}{2}\epsilon, \tag{3.4.7}$$

$$\|J^* - J_{\tilde{\mu}}\|_\infty \leq \|J^* - \tilde{J}_{\tilde{\mu}}\|_\infty + \|\tilde{J}_{\tilde{\mu}} - J_{\tilde{\mu}}\|_\infty \leq 2\epsilon. \tag{3.4.8}$$

(Thus, Eqs. (3.4.7)-(3.4.8) show that $\tilde{\mu}$ is a $2\epsilon$-optimal policy.) We note that a similar reasoning yields the bounds of Eqs. (3.4.7) and (3.4.8) for the special case where Assumption A.4 is in effect and the ergodicity condition is satisfied.

We conclude that the work needed to compute an $\epsilon$-optimal policy is no greater than that of computing an $\frac{\epsilon}{2}$-optimal cost function, and the upper bounds of Theorem 3.2.2 apply to the computation of an $\epsilon$-optimal policy.

Let us now consider the problem of computing an $\epsilon$-optimal *admissible* policy, that is, a policy $\mu \in \Pi$ such that $\|J_\mu - J^*\|_\infty \leq \epsilon$. This can be done, in principle, by first computing an $\epsilon$-optimal policy (for some smaller $\epsilon$) and approximating it by an element of $\Pi$, and using the following lemma:

**Lemma 3.4.2** *(Under Assumption A.4″) Let $J \in \mathcal{B}(S)$, $\mu \in \Pi$, $\tilde{\mu} \in \tilde{\Pi}_h$. Then,*

$$\|T_\mu J - T_{\tilde{\mu}}J\|_\infty \leq (K + \alpha K\|J\|_\infty)\|\mu - \tilde{\mu}\|_\infty.$$

*Furthermore,*

$$\|J_\mu - J_{\tilde{\mu}}\|_\infty \leq \frac{1}{1-\alpha}(K + \alpha K\|J_{\tilde{\mu}}\|_\infty)\|\mu - \tilde{\mu}\|_\infty.$$

*For the special case when Assumption A.4 is in effect, the bounds can be strengthened to yield*

$$\|T_\mu J - T_{\tilde{\mu}} J\|_\infty \leq (K + \frac{\alpha}{2}K\|J\|_s)\|\mu - \tilde{\mu}\|_\infty;$$

$$\|J_\mu - J_{\tilde{\mu}}\|_\infty \leq \frac{1}{1-\alpha}(K + \frac{\alpha}{2}K\|J_{\tilde{\mu}}\|_s)\|\mu - \tilde{\mu}\|_\infty.$$

**Proof** The first part of the lemma follows from the fact that $|g(x,\mu(x))-g(x,\tilde{\mu}(x))|$ and $|P(y|x,\mu(x)) - P(y|x,\tilde{\mu}(x))|$ are both bounded by $K\|\mu-\tilde{\mu}\|_\infty$; the second bound follows from Lemma 2.4.3.

By shifting $J$ by the constant $c = -[\sup_x J(x)+\inf_x J(x)]/2$, we obtain the second part of the lemma. □

The computation of an $\epsilon$-optimal admissible policy $\mu$ proceeds as follows. We first choose a discretization parameter $h$ which is small enough so that the discretization error $K'h/(1-\alpha)^2$ is no greater that $\frac{\epsilon}{8}$. We use the multigrid successive approximation algorithm to compute an $\frac{\epsilon}{4}$-optimal cost function and, according to our earlier discussion, we obtain as a by-product an $\frac{\epsilon}{2}$-optimal policy $\tilde{\mu} \in \Pi_h$; that is, $\|J^*-J_{\tilde{\mu}}\|_\infty \leq \epsilon/2$.

We note from Lemma 2.4.1 that there exists some $\mu \in \Pi$ such that $\|\mu - \tilde{\mu}\|_\infty \leq (K+1)h$; so, by Lemma 3.4.2, $\|J_\mu - J_{\tilde{\mu}}\|_\infty \leq \frac{1}{1-\alpha}(K + \alpha K\|J_{\tilde{\mu}}\|_s)(K+1)h$. It can be seen from the proof of Theorem 2.4.2 that $K(K+1)$ is less than $K_1$ and $K_2$. Proceeding as in Eq. (3.4.4), we obtain $\|J_\mu - J_{\tilde{\mu}}\|_\infty \leq \frac{\epsilon}{8}$. So, by the triangle inequality,

$$\|J^* - J_\mu\|_\infty \leq \|J^* - J_{\tilde{\mu}}\|_\infty + \|J_{\tilde{\mu}} - J_\mu\|_\infty \leq \frac{\epsilon}{2} + \frac{\epsilon}{8} = \frac{5}{8}\epsilon.$$

Thus, $J_\mu$ is indeed an $\epsilon$-optimal admissible policy, as desired.

If the method in the preceding paragraph is to be used, we must be able, given any $\tilde{\mu} \in \tilde{\Pi}_h$, to compute an admissible $\mu \in \Pi$ such that $\|\tilde{\mu} - \mu\|_\infty$ is smaller than $c_1 h$, for some constant $c_1$. This is, in general, impossible under our model of computation; in fact, it is even impossible, in general, to represent an element of $\Pi$ using a finite data structure. On the other hand, for problems that arise in practice, the sets $U(x)$ often have a simple structure and this task is feasible. In those cases, the computation of an $\epsilon$-optimal admissible policy is no harder than the computation of an $\epsilon$-optimal cost function.

### 3.4.3 Lower Bounds for Computing $\epsilon$-Optimal Policies

We observe that an $\epsilon$-optimal policy, by definition, determines the optimal cost function $J^*$ to within $\epsilon$; so, the lower bounds of Theorems 3.3.1–3.3.3 apply to the computation of an $\epsilon$-optimal policy as well. It remains to argue that computing an $\epsilon$-optimal policy is "no easier than" computing an $\epsilon$-optimal cost function (that is, the cost of computing latter is within a constant factor of the cost of computing the former).

For the special case where an ergodicity condition is imposed, the upper bound for computing an $\epsilon$-optimal cost function is within a constant factor of the lower bound [cf. Eqs. (3.2.16) and (3.3.33)]. We conclude that computing an $\epsilon$-optimal policy is no easier than computing an $\epsilon$-optimal cost function. Thus, we have shown that for problems satisfying an ergodicity condition, computing an $\epsilon$-optimal policy is as hard as computing an $\epsilon$-optimal cost function.

We now consider the general case. We fix $\alpha$ and concentrate on the dependence on $\epsilon$. The upper bound for computing an $\epsilon$-optimal cost function is within a constant factor of the lower bound [cf. Eqs. (3.2.15) and (3.3.32)]. Arguing as in the preceding paragraph, we conclude that, with respect to the dependence on $\epsilon$, computing an $\epsilon$-optimal policy is as hard as computing an $\epsilon$-optimal cost function. But because of the "gap" of $O\left(\frac{1}{1-\alpha}\right)$ between the upper and lower bounds, we cannot draw the same conclusion for the dependence on $\alpha$; a different argument is needed.

The basic idea of the argument is as follows. We will show that if an $\frac{\epsilon}{2}$-optimal policy is available, then an $\epsilon$-optimal cost function can be quickly computed (with complexity better than the lower bound). Thus, an algorithm can first compute an $\frac{\epsilon}{2}$-optimal policy, then use the policy to compute an $\epsilon$-optimal cost function with total computational cost within some constant factor of the cost of computing the policy. It follows that computing an $\epsilon$-optimal policy is no easier than computing an $\epsilon$-optimal cost function.

To use the method described in the preceding paragraph, additional assumptions are required. First, we define

$$H(\alpha,\epsilon) = \left\{h \in (0,1) \mid h \geq \frac{1}{8K'}(1-\alpha)^2\epsilon\right\}, \quad \alpha,\epsilon \in (0,1),$$

where $K'$ is the constant of Theorem 2.4.2. It is clear from the discussion in Section 3.4.2 that, for any discount factor $\alpha < 1$ and accuracy parameter $\epsilon > 0$, there exists some $h \in H(\alpha,\epsilon)$ such that $\tilde{\Pi}_h$ contains an $\epsilon$-optimal policy. We next introduce the assumptions:

**C.1** The dimension of the control space $m$ is at least 1.

**C.2** For any $\epsilon$, the $\epsilon$-optimal policy $\tilde{\mu}$ belongs to $\tilde{\Pi}_h$ for some $h \in H(\alpha,\epsilon)$. [That is, we consider only $\epsilon$-optimal policies that are simple on a grid of size in $H(\alpha,\epsilon)$.]

Note that Assumption C.1 excludes problems with finite control space. And we "need" Assumption C.2 to ensure that the policy under consideration is not unnecessarily complicated and can be used to quickly compute an $\epsilon$-optimal cost function.

For the remainder of the discussion, Assumptions C.1–C.2 will be in effect. Let $\tilde{\mu}$ be an $\frac{\epsilon}{2}$-optimal policy and $J^0(x) = 0$ for all $x \in S$. From the successive approximation error bounds [cf. Eq. (3.1.2)], if $J^t = \tilde{T}_{\tilde{\mu}}^t J^0$, then

$$\|\tilde{J}_{\tilde{\mu}} - J^t\|_\infty \leq \frac{\alpha^t}{(1-\alpha)}\|\tilde{T}_{\tilde{\mu}}J^0 - J^0\|_\infty \leq \frac{\alpha^t}{1-\alpha}K,$$

where we have used the fact that $\|\tilde{T}_{\tilde{\mu}}J^0\|_\infty \leq K$.

We now apply $\tilde{T}_{\tilde{\mu}}$ on $J^0$ for $t$ times, where $t$ is the smallest integer such that $\frac{\alpha^t}{1-\alpha}K \leq \frac{\epsilon}{2}$. This ensures that $\|\tilde{J}_{\tilde{\mu}} - J^t\|_\infty \leq \frac{\epsilon}{2}$, and by the triangle inequality,

$$\|J^* - J^t\|_\infty \leq \|J^* - \tilde{J}_{\tilde{\mu}}\|_\infty + \|\tilde{J}_{\tilde{\mu}} - J^t\|_\infty \leq \epsilon.$$

Thus, $J^t$ is an $\epsilon$-optimal cost function, as desired.

To bound the complexity of computing $J^t$, it is seen (cf. Section 3.2.1) that

$$t = \mathrm{O}\left(\frac{\log\left(\frac{1}{(1-\alpha)\epsilon}\right)}{1-\alpha}\right).$$

And by Lemma 3.1.1, the cost of an iteration of $\tilde{T}_{\tilde{\mu}}$ is $\mathrm{O}\left(h^{-(n+m)}\right)$, which by Assumption C.2 is $\mathrm{O}\left(\frac{1}{(1-\alpha)^2\epsilon}\right)$. Thus, using the $\frac{\epsilon}{2}$-optimal policy, we can compute an $\epsilon$-optimal cost function with cost

$$\mathrm{O}\left(\frac{\log\left(\frac{1}{(1-\alpha)\epsilon}\right)}{1-\alpha}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{n+m}\right),$$

which is less than the lower bound $\Omega\left(\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right)$. This completes the proof that under Assumptions C.1 and C.2, computing an $\epsilon$-optimal policy is no easier than computing an $\epsilon$-optimal cost function. Hence, we have shown that for problems not assumed to satisfy an ergodicity condition, computing an $\epsilon$-optimal policy (under Assumptions C.1 and C.2) is as hard as computing an $\epsilon$-optimal cost function.

## 3.5 Extensions

We have completely characterized the computational complexity of MDP's—both the information-based and the total complexity. We have shown that multigrid successive approximation is within a factor of $\mathrm{O}\left(\frac{1}{1-\alpha}\right)$ from the information-based lower bound and is actually optimal when the MDP satisfies certain ergodicity conditions. We have also shown that computing an $\epsilon$-optimal policy is in a certain sense as hard as computing an $\epsilon$-optimal cost function. We discuss here certain extensions of our results.

### 3.5.1 Fredholm Equations of the Second Kind

A Fredholm equation of the second kind is an equation of the form

$$g(x) + \int_S G(x,y)J(y)\,dy = J(y),$$

where $S$ is a bounded subset of $\mathbf{R}^n$, $g$ and $G$ are given functions, and $J$ is the unknown.

The numerical solution of this equation has been well studied (see, for example, Hackbusch (1985), Schippers (1979), and Werschulz (1985) and the references therein). Let us assume that $G$ is a bounded function and that $\int_S |G(x,y)|\, dy \leq \alpha$ for all $x \in S$, where $\alpha \in (0,1)$. If we let $P(y|x) = G(x,y)/\alpha$, it is clear that we are dealing with the general problem when Assumption A.4″ is in effect, except that the the control variable $u$ is absent. (Thus, $m = 0$.) It follows that (under Lipschitz continuity assumptions) our multigrid algorithm can be used to compute an $\epsilon$-approximation of the solution and has complexity

$$O\left(\frac{1}{|\log \alpha|}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n}\right).$$

Furthermore, the lower bound becomes

$$\Omega\left(\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n}\right),$$

and therefore our algorithm is optimal as far as the dependence on $\epsilon$ is concerned.

Multigrid algorithms for Fredholm's equation can also be found in Hackbusch (1985) and Schippers (1979), and they are different in the following respects. First, the algorithms in these references are more general because they do not require a contraction assumption. Furthermore, these algorithms perform computations on fine grids and then use certain coarse-grid corrections. This is in contrast to our method that only proceeds from coarse to fine grids. According to our results, for the problems we are considering, our method has optimal dependence on the accuracy parameter $\epsilon$ and close to optimal dependence on $\alpha$. (Note that $\alpha$ can be viewed as a measure of ill-conditioning of the problem.) It is unclear what the $\alpha$ dependence of the algorithms in Hackbusch (1985) is.

### 3.5.2 Different Error Criteria

Let us consider the $L_p$-norm on $\mathcal{B}(S)$ defined by

$$\|J\|_p \overset{\text{def}}{=} \left[\int_S |J(y)|^p dy\right]^{1/p}, \quad p \in [1,\infty).$$

Since the volume of $S$ is bounded by 1, it is easily shown that $\|J\|_p \leq \|J\|_\infty$ for any $J \in \mathcal{B}(S)$ and any $p \in [1,\infty)$. For this reason, the function $J$ returned by our algorithms automatically satisfies $\|J - J^*\|_p \leq \epsilon$.

We have shown in Section 3.3 that the lower bounds on the computational complexity of the problem do not change when $L_p$-norms are used to measure the error $J - J^*$. It follows such a different choice of norm does not affect the optimality properties of our algorithms.

# Chapter 4

# Further Results

In this chapter we present some further results on the computational aspects of Markov Decision Processes (MDP's). The results here are of an exploratory nature, suggesting potential future research.

An outline of this chapter is as follows. In Section 4.1, we introduce another model of discrete-time stochastic control that is more general than the model discussed in Chapters 2 and 3. We show that this model is computational harder. In Section 4.2, we discuss the policy iteration algorithm and look at some unresolved issues regarding the complexity of a policy-iteration-based multigrid algorithm. In Section 4.3, we discuss average-cost problems. We show that these problems are, in general, computationally ill-posed. We give a condition for well-posedness, and show that multigrid successive approximation is optimal under this condition. In Section 4.4, we discuss some simple numerical experiments comparing single-grid and multigrid successive approximations. In Section 4.5, we summarize the main open problems in the first part of this report (Chapters 2–4) and pose some questions that will be addressed in the second part (Chapter 5).

## 4.1 Another Discrete-Time Stochastic Control Model

In this section, we discuss another discrete-time stochastic model which we call the *f-model*. Although for finite-state problems, this model is equivalent to the model studied in Chapters 2 and 3 (which we call the *P-model*), the f-model is computationally "harder".

Before discussing the f-model, we first look at another class of discrete-time control problems—deterministic problems.

### 4.1.1 Deterministic Problems

In Section 2.4, we have seen that the presence of noise (an ergodicity condition) reduces discretization error; we now show that its absence increases the error.

In deterministic problems, the dynamics of the system with state space $S$ and control space $C$ are described by a mapping $f : S \times C \mapsto S$, and the cost function by a real-valued function $g : S \times C \mapsto \mathbf{R}$. Let $\Pi$ denote the set of all policies $\mu : S \mapsto C$.

(We can also introduce constraint sets as in MDP's.) The optimal cost function is given by

$$J^*(x_0) = \min_{(\mu_0,\mu_1,\ldots)\in\Pi^\infty} \sum_{t=0}^{\infty} \alpha^t g(x_t, \mu_t(x_t)), \qquad x_0 \in S,$$

subject to $x_{t+1} = f(x_t, u_t)$ for all integer $t \geq 0$. It should be clear that the P-model does not include deterministic problems as a special case. (Assumption A.2 excludes them.)

We now show that the discretization error bounds of Section 2.4 do not apply to deterministic problems.

To keep things simple, let $S = [0,1]$ and $C = \{0\}$ (no control). Let

$$f(x) = \begin{cases} Kx & \text{if } x \leq 1/K \\ 1 & \text{otherwise,} \end{cases}$$

and $g(x) = x$, for all $x \in S$. If $\alpha K > 1$, we show that the optimal cost function $J^*$ is not Lipschitz continuous at $x = 0$.

To see this, let $\delta_i = K^{-i}$, for some positive integer $i$. It is clear that $J^*(0) = 0$ and we have

$$\begin{aligned} J^*(\delta_i) &= \sum_{t=0}^{\infty} \alpha^t \left[K^t \delta_i \wedge 1\right] \\ &= \delta_i(1 + (\alpha K) + \cdots + (\alpha K)^{i-1}) + \sum_{t=i}^{\infty} \alpha^t \\ &= \delta_i \frac{(\alpha K)^i - 1}{(\alpha K) - 1} + \frac{\alpha^i}{1-\alpha}. \end{aligned}$$

Therefore, there cannot exist a constant $L_1$ such that

$$|J^*(x) - J^*(0)| \leq L_1 x, \qquad \forall x \in [0,1].$$

Thus, $J^*$ is not Lipschitz continuous at $x = 0$ and the discretization error bounds of Section 2.4 do not apply.

### 4.1.2 The f-Model

We now discuss the more general f-model which includes deterministic problems as a special case. For the f-model, the dynamics are described by $f : S \times C \times D \mapsto S$ where, as before, $S$ and $C$ are the state and control spaces, respectively; $D$ is the disturbance space (which is a measurable space).

For any integer $t \geq 0$, the dynamics are given by

$$x_{t+1} = f(x_t, u_t, w_t), \qquad x_{t+1}, x_t \in S, u_t \in C, w_t \in D.$$

The variable $w_t$ represents the noise and has probability density $P(\cdot \mid x_t, u_t)$.

The optimal cost function is given by

$$J^*(x) = \inf_{(\mu_0,\mu_1,\ldots)\in\Pi^\infty} \mathrm{E}\left\{\sum_{t=0}^{\infty} \alpha^t g(x_t, \mu_t(x_t)) \mid x_0 = x\right\}, \qquad x \in S,$$

where $g : S \times C \mapsto \mathbf{R}$ is the cost function.

It is clear that even if we impose Lipschitz continuity conditions on $g$, $f$, and $P$, the f-model includes deterministic problems as a special case—just let $f(x, u, w)$ be independent of $w$. Therefore we conclude that the discretization error bounds of Section 2.4 do not apply to the f-model. This opens up many questions for future research, which we now discuss.

### 4.1.3 Future Research

1. One research problem is to obtain tight discretization error bounds for general Lipschitz continuous f-models. This involves obtaining both discretization error upper bounds and information-based lower bounds. (It seems that deterministic problems have the worst discretization error—therefore, we should consider them first to obtain upper bounds.)

2. The information-based lower bounds in Section 3.3 show that the piecewise constant discretization procedures (of Sections 2.4–2.5) are optimal for the P-model; moreover, adaptation does not help. However, this may not be the case for the f-model. So, linear interpolation or adaptive discretization schemes might improve discretization error bounds.

3. Finally, the most interesting problem is to find conditions that guarantee a Lipschitz continuous optimal cost function and lead to a linear discretization error bound for the f-model. Ergodicity and controllability conditions are the natural candidates.

## 4.2 Multigrid Policy Iteration

In this section we discuss an alternative method for computing an approximation of the optimal cost function—policy iteration. This algorithm (which at each step involves solving a linear equation) is often the method of choice in practice because of its fast convergence rate (and the wide availability of linear equation solvers).

However, we have not been able to fully analyze the computational complexity of a multigrid version of this algorithm—in particular, its dependence on $\alpha$. We will discuss some preliminary observations. [The main reason for considering this multigrid algorithm is to explore the possibility of closing the $O(1/(1-\alpha))$ "complexity gap" between the upper bounds of Section 3.2 and the lower bounds of Section 3.3.

An outline of the section is as follows. First, we describe policy iteration. We show that it converges as fast as successive approximation. Second, we consider how policy iteration may be incorporated into a multigrid algorithm and identify some of the issues that must be addressed before we can fully analyze the complexity of the multigrid policy-iteration algorithm.

## 4.2.1 Basic Ideas

We now discuss the basic ideas of policy iteration. The algorithm proceeds as follows. Given any initial policy $\mu \in \Pi$, we can compute its cost function $J_\mu$ by solving the following linear equation:

$$g_\mu + \alpha P_\mu J = J, \tag{4.2.1}$$

where $g_\mu(x) = g(x, \mu(x))$ and $P_\mu J(x) = \int_S P(y \mid x, u) J(y)\, dy$. If the state space is finite (the integral is replaced by a sum), we can solve Eq. (4.2.1) exactly using a direct method such as Gaussian elimination.

Suppose that at the $t$-step of the algorithm we have $\mu_t$ and have solved for $J_{\mu_t}$. We can obtain a better policy $\mu_{t+1} \in \Pi$ by letting $\mu_{t+1}$ be the minimizing policy in $TJ_{\mu_t}$; that is,

$$TJ_{\mu_t} = T_{\mu_{t+1}} J_{\mu_t}. \tag{4.2.2}$$

Once we have $\mu_{t+1}$ we can solve for $J_{\mu_{t+1}}$, and so on.

A simple stopping criterion is $\mu_{t+1} = \mu_t$, in which case $\mu_t$ is an optimal policy (and $J_{\mu_t}$ is the optimal cost function). Another stopping criterion is when $J_{\mu_{t+1}}$ is within some desired threshold of $J^*$. However, unlike successive approximation, there are no "nice" error bounds for $\|J^* - J_{\mu_{t+1}}\|_\infty$. But it is known that the sequence of $\{J_{\mu_t}\}$ converges to $J^*$ as fast as the iterates of successive approximation. We now review this fact. [For a proof of a more general result, see Denardo (1967).]

By definition, we have

$$J^* \leq J_{\mu_t}.$$

Using the monotonicity of $T$, we obtain

$$J^* = TJ^* \leq TJ_{\mu_t} = T_{\mu_{t+1}} J_{\mu_t} \leq J_{\mu_t}.$$

More generally, we have

$$J^* \leq \ldots \leq T^2_{\mu_{t+1}} J_{\mu_t} \leq T_{\mu_{t+1}} J_{\mu_t} \leq J_{\mu_t}.$$

Since $T^i_{\mu_{t+1}} J_{\mu_t}$ converges to $J_{\mu_{t+1}}$, as $i \uparrow \infty$, we have $J^* \leq J_{\mu_{t+1}} \leq J_{\mu_t}$ and

$$\begin{aligned} \|J^* - J_{\mu_{t+1}}\|_\infty &\leq \|J^* - T_{\mu_{t+1}} J_{\mu_t}\|_\infty \\ &= \|J^* - TJ_{\mu_t}\|_\infty \\ &\leq \alpha \|J^* - J_{\mu_t}\|_\infty. \end{aligned} \tag{4.2.3}$$

Thus, $J_{\mu_{t+1}}$ converges to $J^*$ *linearly* with rate $\alpha$.

Note that the above argument requires the monotonicity of $T$ and, therefore, requires Assumption A.4$'$ or A.4. In particular, the multigrid policy-iteration which we will discuss next does not apply to $\mathcal{P}_{\text{gen}}$ (where Assumption A.4$''$ is in effect)—in contrast with the multigrid successive approximation algorithm of Section 3.2.

### 4.2.2 A Policy-Iteration-Based Multigrid Algorithm

We now discuss how policy iteration may be incorporated into a multigrid algorithm.

Fix a grid-level $h$. Recall that in the multigrid successive approximation algorithm (Section 3.2), we start with an initial estimate $J_h^I$ and we want to compute a final estimate $J_h^F$ (before going to a finer grid-level) such that

$$\|\tilde{J}_h^* - J_h^F\|_\infty \leq \frac{1}{L}\|\tilde{J}_h^* - J_h^I\|_\infty, \tag{4.2.4}$$

for some $L > 1$. (That is, we want to reduce the approximation error by some constant factor at each grid-level.) Also recall that in multigrid successive approximation, the number of iterations at each grid-level is $O(1/(1-\alpha))$.

We can incorporate policy iteration in the multigrid algorithm by using policy iteration instead of successive approximation to compute the final estimate $J_h^F$. The idea is that given the initial estimate $J_h^I$ we can compute an initial policy $\tilde{\mu}_0$ by letting $\tilde{T}_h J_h^I = \tilde{T}_{\mu_0} J_h^I$. We then use policy iteration to compute the final estimate $J_h^F = \tilde{J}_{\tilde{\mu}_t}$, for some $\tilde{\mu}_t \in \tilde{\Pi}_h$, so that Eq. (4.2.4) is satisfied. [This algorithm has been suggested and studied in the context of continuous-time stochastic control by Akian, et al (1988) and Hoppe (1986). But the complexity of the algorithm has not been rigorously analyzed.] To analyze the complexity of this algorithm, there are two questions which must be addressed.

1. Given a policy $\tilde{\mu} \in \tilde{\Pi}_h$, can we solve for $\tilde{J}_{\tilde{\mu}}$ with $O(1/h^{2n+m})$ operations?
2. Can policy iteration reduce the number of policy-iteration steps at each grid-level from $O(1/(1-\alpha))$ to $O(1)$?

[Note that each policy iteration step involves a successive approximation iteration, which requires $O(1/h^{2n+m})$ operations.] We will address these two issues separately.

#### Solving For The Cost Function of A Policy

At grid-level $h$, the linear equation

$$\tilde{g}_{\tilde{\mu}} + \alpha \tilde{P}_{\tilde{\mu}} J = J, \qquad \tilde{\mu} \in \tilde{\Pi}_h,$$

consists of the "vector" $\tilde{g}_{\tilde{\mu}}$ with $O(1/h^n)$ entries and the "matrix" $\tilde{P}_{\tilde{\mu}}$ with $O(1/h^{2n})$ entries. Using Gaussian elimination, we can solve for $\tilde{J}_{\tilde{\mu}}$ in $O(1/h^{3n})$ operations. If $n \geq m$ this is certainly within $O(1/h^{2n+m})$.

On the other hand we do not need to solve for $\tilde{J}_{\tilde{\mu}}$ exactly. We only need to compute it approximately. Using a Full-Multigrid V-cycle [see Böhmer and Stetter (1984) and Briggs (1987)], one can solve for $\tilde{J}_{\tilde{\mu}}$ to within an accuracy of $O(h)$ (which seems to be the accuracy needed) with $O(1/h^{2n})$ operations. However, the analysis assumes certain special structure in $\tilde{P}_{\tilde{\mu}}$, and it is not clear whether $\tilde{P}_{\tilde{\mu}}$ has the assumed structure; therefore, the analysis may not be applicable. Moreover, we also do not know the $\alpha$ dependence of the algorithm.

Another approach is to use $\tilde{T}^i_{\tilde{\mu}_{t+1}} \tilde{J}_{\tilde{\mu}_t}$, for some $i > 0$, as an estimate of $\tilde{J}_{\tilde{\mu}_{t+1}}$. [See L'Ecuyer (1989) and the references therein for related algorithms.] At grid-level $h$, each iteration of $\tilde{T}_{\tilde{\mu}_{t+1}}$ costs $\mathrm{O}\left(1/h^{2n}\right)$; furthermore, we have

$$\|\tilde{J}_{\tilde{\mu}_{t+1}} - \tilde{T}^i_{\tilde{\mu}_{t+1}} \tilde{J}_{\tilde{\mu}_t}\|_\infty \le \alpha^i \|\tilde{J}_{\tilde{\mu}_{t+1}} - \tilde{J}_{\tilde{\mu}_t}\|_\infty.$$

If we use a threshold such as

$$\|\tilde{J}_{\tilde{\mu}_{t+1}} - \tilde{T}^i_{\tilde{\mu}_{t+1}} \tilde{J}_{\tilde{\mu}_t}\|_\infty \le \frac{1}{2L} \|\tilde{J}^*_h - J^I_h\|_\infty$$

[cf. Eq. (4.2.4)], then it is seen that $i = \mathrm{O}\left(1/(1-\alpha)\right)$; that is, we only need to reduce $\|\tilde{J}_{\tilde{\mu}_{t+1}} - \tilde{J}_{\tilde{\mu}_t}\|_\infty$ by a constant factor. Hence, the total complexity of the algorithm is $\mathrm{O}\left(1/[(1-\alpha)h^{2n}]\right)$. For $m \ge 1$, this is less than $\mathrm{O}\left(1/h^{2n+m}\right)$ [since $h \sim (1-\alpha)\epsilon$, with ergodicity, or $h \sim (1-\alpha)^2\epsilon$, in general.] However, a more detailed analysis is needed to determine how accurately we should approximate $\tilde{J}_{\tilde{\mu}_{t+1}}$ (the actual threshold needed) and the overall impact of approximating $\tilde{J}_{\tilde{\mu}_{t+1}}$ (instead of computing it exactly) on the convergence of the multigrid algorithm.

**The Convergence of Policy Iteration**

We now consider the number of policy-iteration steps needed at each grid-level and whether we can close the "complexity gap" $\mathrm{O}\left(1/1-\alpha\right)$ in multigrid successive approximation. For the rest of the discussion we assume that we can solve $\tilde{J}_{\tilde{\mu}}$ exactly with complexity no more that $\mathrm{O}\left(1/h^{2n+m}\right)$.

It is clear that the linear convergence bound on policy iteration [cf. Eq. (4.2.3)] is not good enough since it requires $\mathrm{O}\left(1/1-\alpha\right)$ steps. However, under certain convexity and differentiability conditions, the rate of convergence of policy iteration can be shown to be superlinear or even quadratic [Puterman and Brumelle (1979)]. Some of the issues that need to be resolved are: (i) how a superlinear or quadratic convergence rate affects the number of iterations at each grid-level, (ii) the $\alpha$ dependence of the convergence rate of policy iteration, and (iii) how do the convexity and differentiability conditions affect the information-based lower bounds [cf. Theorems 3.3.1–3.3.3].

## 4.3 Average Cost Problems

In this section, we consider Markov Decision Processes (MDP's) with the average cost criterion. These MDP's are called *average cost problems,* in contrast with the *discounted cost problems* considered in the preceding chapters. The results (and their derivations) are similar to those in Chapters 2 and 3. For this reason, we argue informally, keeping the proofs simple. The notation in this section, unless otherwise stated, remains the same as before.

An outline of the section is as follows. First, we formulate the average cost problems and define the various problem classes. Second, we show that average cost problems are, in general, *computationally ill-posed*—that is, the information-based complexity of determining the optimal average-cost to within $\epsilon$ is unbounded in $\epsilon$. (This means that there cannot exist any discretization error bound for these problems.) Third, we show that if the dynamics of the problem satisfy a $k$-stage ergodicity condition (Assumption E.3), then the problem is well-posed; moreover, the multigrid successive approximation algorithm of Section 3.2.2 can be used to compute an $\epsilon$-approximation of the optimal average cost. Furthermore, the complexity ($\epsilon$ dependence) of the algorithm is within a constant factor of the information-based lower bound; thus, the algorithm is optimal. Similar optimality results hold for the computation of an $\epsilon$-optimal policy.

## 4.3.1 Problem Formulation

We now define the average cost problems and their problem classes. Average cost problems are in every respect similar to the discounted cost problems introduced in Chapter 2 and studied in Chapter 3. The only difference is that instead of the discounted-cost criterion [Eq. (2.2.3)], we use the following *average cost criterion:*

$$\lambda_\pi(x) \overset{\text{def}}{=} \lim_{N\uparrow\infty} \frac{1}{N} \mathrm{E}\left\{ \sum_{t=0}^{N-1} g(x_t^\pi, \mu_t(x_t^\pi) \;\middle|\; x_0^\pi = x \right\}, \quad \pi \in \Pi, x \in S. \tag{4.3.1}$$

[If the limit in Eq. (4.3.1) is not known to exist, then "lim sup" is used instead.] The *optimal average cost* is given by

$$\lambda^*(x) \overset{\text{def}}{=} \inf_{\pi\in\Pi^\infty} \lambda_\pi(x), \quad x \in S. \tag{4.3.2}$$

The function $\lambda^*(x)$ corresponds to the long term cost per stage (the average cost) of operating the system starting at the initial state $x$, and with actions chosen optimally. For most problems of interest, $\lambda^*$ is actually a scalar, independent of the state—this is why we call $\lambda^*$ a "cost" instead of a "cost function." The problem is to compute an $\epsilon$-approximation to $\lambda^*$ (with respect to the sup-norm).

It is well known that average cost problems are "harder" than their discounted cost counterparts; moreover, some measurability issues are still unresolved [see Bertsekas and Shreve (1978)]. We do not address these issues here; instead, we focus on the computational aspects of the problem.

### Problem Classes

We now define different classes of average cost problems. The definitions here are similar to those for discounted-cost problems. One difference is that $\alpha$ is no longer a parameter of the problem; therefore our definitions and notation will have to reflect this. For simplicity, we will keep the same notation.

We define the problem class $\mathcal{P}_{\text{std}}$ similarly as in Section 3.1.2—it denotes the class of all average-cost MDP's that satisfy Assumptions A.1–A.5 (see Section 2.6). We also define $\mathbf{C}_{\text{std}}$ as in Section 3.1.2.

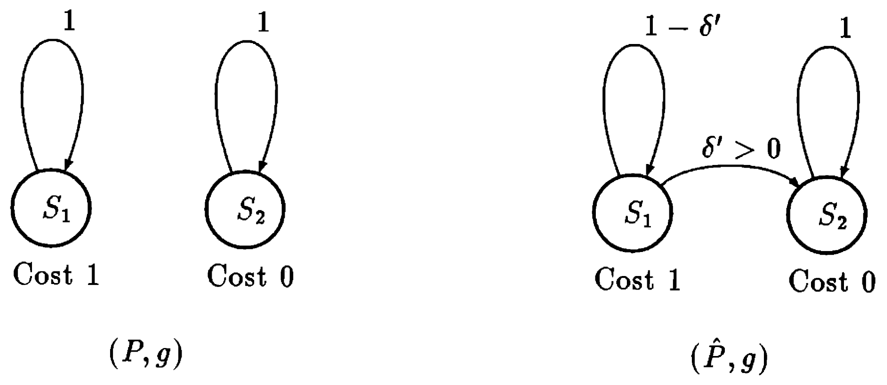

**Figure** 4.3.1: Two instances illustrating the ill-posedness of $\mathcal{P}_{\text{std}}$

In addition, we consider the following three problem classes in this section: the class $\mathcal{P}_{\text{mix}}$, where a $k$-stage ergodicity condition is assumed (Assumption E.3); $\mathcal{P}_{\text{subl}}$, where Assumption A.4′ and a 1-stage ergodicity condition (Assumption E.1) are in effect; and $\mathcal{P}_{\text{mixl}}$, where Assumptions A.4 and E.1 are in effect. The definitions of $\mathbf{C}_{\text{mix}}$, $\mathbf{C}_{\text{subl}}$, and $\mathbf{C}_{\text{mixl}}$ are similar to those for discounted-cost problems (see Section 3.1.2 for details).

## 4.3.2 Ill-Posed Average Cost Problems

We now show that $\mathbf{C}_{\text{std}}(\epsilon)$ and $\mathbf{C}_{\text{subl}}(\epsilon)$ are unbounded in $\epsilon$; that is, no (finite) number of oracle queries can determine an $\epsilon$-approximation of the optimal average cost for $\mathcal{P}_{\text{std}}$ and $\mathcal{P}_{\text{subl}}$. The proofs for these two results parallel the proofs of Theorems 3.3.2 and 3.3.3, respectively. Therefore, instead of repeating the proofs we argue informally (with the help of some pictures)

**Theorem 4.3.1** *(Ill-posedness under Assumption A.4) For every $m$, $n$, there exists some $K$ such that $\mathbf{C}^{\text{info}}_{\text{std}}(\epsilon)$ is unbounded in $\epsilon$; therefore, the problem class $\mathcal{P}_{\text{std}}$ is ill-posed.*

**Proof** In the proof of Theorem 3.3.2, we have constructed two instances of $\mathcal{P}_{\text{std}}$. The differences between these two instances [denoted by $(P,g)$ and $(\hat{P},g)$] are illustrated in Figure 4.3.1.

These two instances agree on all the points sampled by the computer. For the first instance, $(P,g)$, it is clear that the optimal average cost is 1 on $S_1$ and 0 on $S_2$ (see Figure 4.3.1). To construct the second instance, $(\hat{P},g)$, we perturb the dynamics of the first instance by channeling some of the probability from $S_1$ to $S_2$. It is clear that if the process starts in $S_1$ it eventually enters $S_2$ (in finite steps) and stays there forever with 0 cost. Thus, the average cost for the second instance is 0 on both $S_1$ and $S_2$. This result holds no matter how many points the algorithm samples.

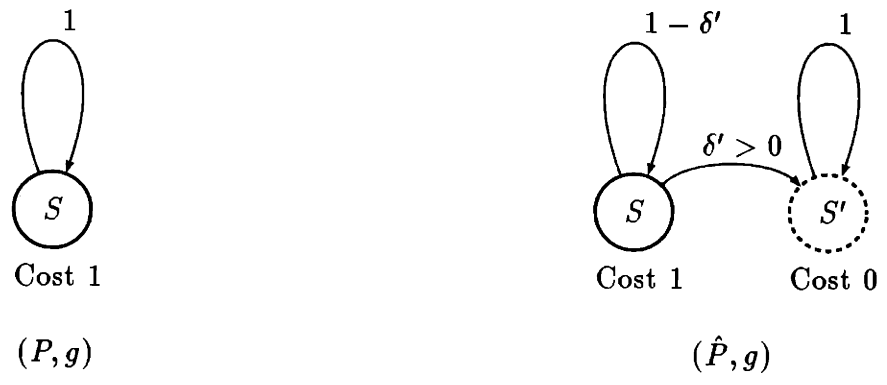

**Figure** 4.3.2: Two instances illustrating the ill-posedness of $\mathcal{P}_{\mathsf{sub1}}$

Therefore, the (maximum) difference between the average costs of the two problems is 1, independent of the number of oracle queries. Thus, $\mathbf{C}^{\mathsf{info}}_{\mathsf{std}}(\epsilon)$ is unbounded in $\epsilon$ as required. □

**Theorem 4.3.2** *(Ill-posedness under Assumptions A.4′ and E.1) For every $m$, $n$, $K > 0$, $\rho \in (0,1)$, $\mathbf{C}^{\mathsf{info}}_{\mathsf{sub1}}(\epsilon)$ is unbounded in $\epsilon$; therefore, the problem $\mathcal{P}_{\mathsf{sub1}}$ is ill-posed.*

**Proof** The proof of this result is similar to the proof of Theorem 3.3.3. We construct two instances $[(P, g)$ and $(\hat{P}, g)]$ that agree on all the points sampled. The key differences between these two instances are illustrated in Figure 4.3.2.

In Figure 4.3.2, the dynamics of the first instance are described by a uniform probability density $P$ on the whole state space $S$. [Therefore, it satisfies all the Lipschitz continuity assumptions and also the 1-stage ergodicity condition (Assumption E.1).] The cost is 1 on the whole state space; therefore the average cost is 1.

In the second instance, $\hat{P}$ is a subprobability density, where we have removed some probability at those points not sampled by the algorithm. We can view this system as equivalent to another system with an augmented state $S'$ and the missing probability is being channeled into $S'$. The augmented state is absorbing and has 0 cost. Following the same reasoning as in the proof of Theorem 4.3.1, we conclude that the average cost for the second instance is 0; this shows the (computational) ill-posedness of $\mathcal{P}_{\mathsf{sub1}}$. □

### 4.3.3 Well-Posed Average Cost Problems

We now show that $\mathcal{P}_{\mathsf{mix}}$, the set of average cost problems (under Assumption A.4) that satisfy a $k$-stage ergodicity condition (Assumption E.3) is well-posed. Moreover, the multigrid successive approximation algorithm of Section 3.2.2 can be used to compute an $\epsilon$-approximation of the optimal average cost with total complexity upper bounded by $O\left(1/\epsilon^{-(2n+m)}\right)$. We also show that the information-based complexity is lower bounded by $\Omega\left(1/\epsilon^{-(2n+m)}\right)$; therefore, the multigrid algorithm is optimal. Similar optimality results hold for the computation of $\epsilon$-optimal policies.

The results here are similar to the results in Chapter 3. To avoid repetitions, the proofs in this section are especially brief. For the rest of the section, Assumptions A.1–A.5 are in effect.

An outline of this subsection is as follows. First, we introduce the average-cost version of the dynamic programming equation—the optimality equation. We show that this equation is always satisfied, when the problem satisfies a $k$-stage ergodicity condition. Second, we obtain discretization error bounds for approximating the optimal average cost. This shows the computational well-posedness of the problem. Third, we show how the single-grid and multigrid successive approximation algorithms of Section 3.2 can be used for approximating the optimal average cost. We analyze the complexity of these algorithms. Fourth, we derive a lower bound on the information-based complexity and show the optimality of multigrid successive approximation. Finally, we discuss the problem of computing an $\epsilon$-optimal policy.

## The Optimality Equation

For average cost problems, the dynamic programming operator $T : \mathcal{B}(S) \mapsto \mathcal{B}(S)$ is given by

$$TJ(x) = \inf_{u \in C} \left\{ g(x,u) + \int_S J(y) P(y \mid x,u)\, dy \right\}, \quad x \in S.$$

By letting $\alpha = 1$ in the proof of Lemma 2.2.1, we see that $T$ maps $\mathcal{B}(S)$ into $\mathcal{C}(S)$.

For any (stationary) policy $\mu \in \Pi$, we write $\lambda_\mu$ instead of $\lambda_\pi$ [cf. Eq. (4.3.1)], where $\pi = (\mu, \mu, \ldots)$. The operator $T_\mu : \mathcal{B}(S) \mapsto \mathcal{B}(S)$ is given by

$$T_\mu J(x) = g(x, \mu(x)) + \int_S J(y) P(y \mid x, \mu(x))\, dy, \quad x \in S.$$

Note that the definitions of $T$ and $T_\mu$ are the same as the definitions for discounted-cost problems [cf. Eqs. (2.2.4) and (2.2.8)], except that $\alpha = 1$.

The following theorem is well known; it gives an alternative characterization of the optimal average cost and a condition for the existence of a stationary policy.

**Theorem 4.3.3** *(Under Assumption A.4) Suppose that there exists a scalar $\lambda^{**}$ and a function $J^* \in \mathcal{B}(S)$ such that*

$$\lambda^{**} + J^*(x) = TJ^*(x), \quad \textit{for all } x \in S.$$

*Then*

1. *$\lambda^{**} \le \inf_{\pi \in \Pi^\infty} J_\pi(x)$ for all $x \in S$; equivalently $\lambda^{**} \le \lambda^*$.*
2. *If $\mu^* \in \Pi$ is a (stationary) policy such that*

$$\lambda^{**} + J^*(x) = T_{\mu^*} J^*(x), \quad \textit{for all } x \in S, \tag{4.3.3}$$

*(equivalently, $TJ^* = T_{\mu^*} J^*$), then $\lambda^{**} = \lambda_{\mu^*}$; in particular, $\lambda^{**} = \lambda^*$.*

**Proof** See Hernández-Lerma (1989) or Bertsekas (1987). □

[Note that the important assumption in Theorem 4.3.3 is that $\lambda^{**}$ is a scalar.]

Equation (4.3.3) is the average-cost version of the dynamic programming equation, also called the *optimality equation.* Some of the questions in (continuous-state) average-cost problems are under what conditions the following results are true: (i) there exists a *solution pair* to the optimality equation; namely, there exist a scalar $\lambda^{**}$ and a function $J^* \in \mathcal{B}(S)$ satisfying Eq. (4.3.3), and (ii) there exists an optimal stationary policy. The following theorem gives an answer to these questions.

**Theorem 4.3.4** *(Under Assumption A.4) If the problem satisfies a k-stage ergodicity condition (Assumption E.3), then there exist a scalar $\lambda^{**}$ and a $J^* \in \mathcal{C}(S)$ satisfy the optimality equation [Eq. (4.3.3)]. Furthermore, there exists a stationary policy $\mu^* \in \Pi$ such that $TJ^* = T_{\mu^*}J^*$. In particular, $\lambda^{**}$ is the optimal average cost and $\mu^*$ is an optimal stationary policy.*

**Proof** It follows from the ergodicity condition (Assumption E.3) that $T$ is a $k$-stage contraction operator on the Banach space $(C(S), \|\cdot\|_s)$. By the Banach's fixed-point theorem (Proposition 2.1.1), there exists a $J^* \in \mathcal{C}(S)$ such that $\|TJ^* - J^*\|_s = 0$. Equivalently, there exists a scalar $\lambda^{**}$ such that $\lambda^{**} + J^* = TJ^*$; hence the optimality equation is satisfied.

Furthermore, by the measurable selection theorem (Proposition 2.2.3), there exists a $\mu^* \in \Pi$ such that $T_{\mu^*}J^* = TJ^*$. It follows from Theorem 4.3.3 that $\lambda_{\mu^*} = \lambda^{**} = \lambda^*$, as required. □

When $\lambda^{**} = \lambda^*$, $J^*$ is called the *optimal differential cost function.* For the rest of this section, we will assume that Assumption E.3 is in effect; therefore, $\lambda^{**}$ and $\lambda^*$ are identical.

Note that for discounted cost problems, $J^*$ denotes the optimal total (discounted) cost function; but for average cost problems, $J^*$ denotes the optimal differential cost function.

The optimal differential cost function $J^*$ has the following interpretation. The difference $|J^*(x) - J^*(y)|$ represents the difference in total costs starting at state $x$ and starting at state $y$ [see Bertsekas (1987)]. This difference $|J^*(x) - J^*(y)|$ makes sense only when $x$ and $y$ have the same optimal average cost. Finally, note that $J^*$ can only be determined up to a constant [since $T(J^* + 1) = TJ^* + 1$].

### Discretization Error Bounds

We now derive discretization error bounds for the average-cost problems. We use the same discretization procedures used for discounted-cost problems (see Section 2.4). We also define the discretized dynamic programming operator $\tilde{T}_h$ as in Eq. (2.4.7), but with $\alpha = 1$.

Let $\lambda_h^*$ and $\tilde{J}_h^*$ denote the solution pair to the optimality equation associated with $\tilde{T}_h$. Similarly to Theorem 2.4.3, we have the following discretization error bounds.

**Theorem 4.3.5** *(Under Assumption A.4) Let $h_b$ be the constant of Theorem 2.4.1. If the problem satisfies a $k$-stage ergodicity condition (Assumption E.3) with ergodicity rate $2\rho$, then there exists a constant $K'''$ (depending only on $K$, $k$, and $\rho$) such that*

$$|\lambda^* - \lambda_h^*| \leq K'''h, \qquad \forall h \in (0, h_b].$$

**Proof** Fix an $h \in (0, h_b]$. By letting $\alpha = 1$ in Theorem 2.4.3, we have

$$\|TJ - \tilde{T}_h J\|_\infty \leq (K_1 + \frac{K_2}{2}\|J\|_s)h.$$

Using the fact that $\|\cdot\|_s \leq 2\|\cdot\|_\infty$, we have

$$\|TJ - \tilde{T}_h J\|_s \leq (2K_1 + K_2\|J\|_s)h.$$

It follows from Theorem 2.4.1 that $\tilde{T}_h$ is a $k$-stage contraction operator, with contraction factor $(1-\rho)$, on $(\mathcal{B}(S), \|\cdot\|_s)$.

Using Lemma 2.4.3, we have

$$\begin{aligned}\|J^* - \tilde{J}_h^*\|_s &\leq \frac{k}{\rho}\|\tilde{T}_h J^* - J^*\|_s \\ &\leq \frac{k}{\rho}(2K_1 + K_2\|J^*\|_s)h.\end{aligned}$$

It follows from Lemma 2.4.5 that $\|J^*\|_s \leq kK/\rho$. Therefore,

$$\|J^* - \tilde{J}_h^*\|_s \leq \frac{k}{\rho}(2K_1 + kKK_2/\rho)h.$$

Since $J^*$ is determined only up to a constant, we can choose $J^*$ such that $J^*(x) = \tilde{J}_h^*(x)$ for some $x \in S$. In particular,

$$\|J^* - \tilde{J}_h^*\|_\infty \leq \|J^* - \tilde{J}_h^*\|_s \leq \frac{k}{\rho}(2K_1 + kKK_2/\rho)h.$$

From the optimality equation [Eq. (4.3.3)] we have $\lambda^* = TJ^* - J^*$ and $\lambda_h^* = \tilde{T}_h\tilde{J}_h^* - \tilde{J}_h^*$. Therefore using the triangle inequality, we have

$$\begin{aligned}|\lambda^* - \lambda_h^*| &= \|TJ^* - J^* - \tilde{T}_h\tilde{J}_h^* + \tilde{J}_h^*\|_\infty \\ &\leq \|TJ^* - \tilde{T}_h\tilde{J}_h^*\|_\infty + \|J^* - \tilde{J}_h^*\|_\infty \\ &\leq \|TJ^* - \tilde{T}_h J^*\|_\infty + \|\tilde{T}_h J^* - \tilde{T}_h\tilde{J}_h^*\|_\infty + \|J^* - \tilde{J}_h^*\|_\infty \\ &\leq (K_1 + \frac{K_2}{2}\|J^*\|_s)h + 2\|J^* - \tilde{J}_h^*\|_\infty \\ &\leq K'''h,\end{aligned}$$

for a suitable constant $K'''$. The result follows. □

## Successive Approximation Algorithms

We now consider the computation of an $\epsilon$-approximation of $\lambda^*$. Similarly to the successive approximation algorithm introduced in Section 3.1.3, for the average-cost problems, we have the following successive approximation algorithm.

**Lemma 4.3.1** *(Under Assumptions A.4 and E.3) For all $h \in (0, h_b]$, there holds*

$$\min_{x \in S}\{\tilde{T}_h^{t+1}J(x) - \tilde{T}_h^t J(x)\} \le \lambda_h^* \le \max_{x \in S}\{\tilde{T}_h^{t+1}J(x) - \tilde{T}_h^t J(x)\}, \quad \forall t \ge 0, J \in \mathcal{B}(S). \tag{4.3.4}$$

*In particular, if $\tilde{T}_h$ is a $k$-stage contraction operator on $(\mathcal{B}(S), \|\cdot\|_s)$ with contraction factor $(1-\rho)$ and we let*

$$\lambda_h^{t+1} = \left[\min_x\{\tilde{T}_h^{t+1}J(x) - \tilde{T}_h^t J(x)\} + \max_x\{\tilde{T}_h^{t+1}J(x) - \tilde{T}_h^t J(x)\}\right]/2, \tag{4.3.5}$$

*then*

$$\begin{aligned}\left|\lambda_h^* - \lambda_h^{t+1}\right| &\le \|\tilde{T}_h^{t+1}J - \tilde{T}_h^t J\|_s \\ &\le (1-\rho)^{\lfloor t/k \rfloor}\|\tilde{T}_h J - J\|_s, \quad \forall t \ge 0, J \in \mathcal{B}(S).\end{aligned}$$

**Proof** See Bertsekas (1987). □

We now describe the successive approximation algorithms for the average-cost problems. Recall that if the continuous problem satisfies a $k$-stage ergodicity condition with ergodicity rate $2\rho$, then for all $h \in (0, h_b]$, the discretized problem with discretization parameter $h$ also satisfies a $k$-stage ergodicity condition with ergodicity rate $\rho$; in particular, $\tilde{T}_h$ is a $k$-stage contraction operator on $(\mathcal{B}(S), \|\cdot\|_s)$. Also recall (from Lemma 3.1.1) that each iteration of $\tilde{T}_h$ costs $O(1/h^{2n+m})$.

The single-grid algorithm for the average cost problems is identical to the single-grid algorithm in Section 3.2.1. We choose a grid-size $h_f$ such that $\left|\lambda_{h_f}^* - \lambda^*\right| \le \epsilon/2$; we iterate on grid-level $h_f$ to get an estimate within $\epsilon/2$ of $\lambda_{h_f}^*$. The only difference is that the algorithm returns the scalar $\lambda_{h_f}^t$ [given by Eq. (4.3.5)]. The termination criterion is the same, and the complexity of algorithm is $O\left(\log(1/\epsilon)\epsilon^{-(2n+m)}\right)$.

The multigrid algorithm for the average cost problems is also identical to the multigrid algorithm of Section 3.2.2. The grid-refinement criterion at each grid-level is to iterate until the successive approximation error bound is no more than the discretization error bound. Following the same complexity analysis, we obtain a complexity of $O\left(\epsilon^{-(2n+m)}\right)$ for the algorithm. We summarize our results by the following theorem

**Theorem 4.3.6** *(Under Assumptions A.4 and E.3) To compute an $\epsilon$-approximation of $\lambda^*$, the complexity of single-grid successive approximation is $O\left(\log(1/\epsilon)\epsilon^{-(2n+m)}\right)$ and the complexity of multigrid successive approximation is $O\left(\epsilon^{-(2n+m)}\right)$.*

## Information-Based Bounds

We now show the optimality of the multigrid algorithm. We note that from the discretization error bounds, the information-based upper bound for $\mathcal{P}_{\text{mix}}$, $\mathbf{C}^{\text{info}}_{\text{mix}}(\epsilon) = O\left(\epsilon^{-(2n+m)}\right)$. We now show that the information-based lower bound is $\Omega\left(\epsilon^{-(2n+m)}\right)$; it follows that the multigrid algorithm is optimal. The following theorem is analogous to Theorem 3.3.1, and the proof is almost identical except for the changes noted.

**Theorem 4.3.7** *(Lower bound under Assumption A.4 and E.1) For any $K > 0$, $\rho \in (0,1)$, $m$, and $n$, we have*

$$\mathbf{C}^{\text{info}}_{\text{mix1}}(\epsilon) = \Omega\left(\epsilon^{-(2n+m)}\right);$$

*in particular,*

$$\mathbf{C}_{\text{mix1}}(\epsilon) = \Omega\left(\epsilon^{-(2n+m)}\right).$$

**Proof** The proof is similar to the proof of Theorem 3.3.1 (except that $\alpha = 1$). The construction of the two instances are the same. It is easy to see that for the first instance $(P, g)$, we have the optimal average cost $\lambda^* = 1/2$ (and the differential optimal cost function $J^*(x) = x$, $\forall x \in S$, modulus a constant difference). We now bound the optimal average cost for the second instance $(\hat{P}, \hat{g})$. We proceed as in the proof of Theorem 3.3.1. Instead of Eq. (3.3.22), we have

$$\hat{T}^t J^*(x) \leq J^*(x) + t\lambda^* - (t-2)\kappa\delta, \quad \forall t \geq 2, x \in S. \tag{4.3.6}$$

Since differential cost functions are determined only up to a constant, we can choose the differential cost for the second instance $\hat{J}^* \leq 0$; in particular, $\hat{J}^* \leq J^*$. Using the monotonicity of $\hat{T}$, we have

$$\begin{aligned} \hat{T}^t J^*(x) &\geq \hat{T}^t \hat{J}^*(x) \\ &= t\hat{\lambda}^* + \hat{J}^*(x), \quad \forall t \geq 0, x \in S. \end{aligned} \tag{4.3.7}$$

Combining Eqs. (4.3.6) and (4.3.7), we obtain

$$t\hat{\lambda}^* + \hat{J}^*(x) \leq J^*(x) + t\lambda^* - (t-2)\kappa\delta, \quad \forall t \geq 2, x \in S.$$

Dividing both sides of the equation by $t$ and letting $t \uparrow \infty$, we obtain

$$\hat{\lambda}^* \leq \lambda^* - \kappa\delta.$$

Since we want $\left|\hat{\lambda}^* - \lambda^*\right| \leq 2\epsilon$, therefore we need $\kappa\delta \leq 2\epsilon$. And using the same argument as in the proof of Theorem 3.3.1, we obtain $\mathbf{C}^{\text{info}}_{\text{mix1}}(\epsilon) = \Omega\left(\epsilon^{-(2n+m)}\right)$, as required. □

Therefore, the discretization procedures of Sections 2.4–2.5 and the multigrid successive approximation algorithm are optimal for the average cost problem $\mathcal{P}_{\text{mix}}$.

### 4.3.4 Other Results

We formulate the problem of computing an $\epsilon$-optimal policy as in Section 3.4.1. As before, for any $\tilde{\mu} \in \tilde{\Pi}_h$ we define $T_{\tilde{\mu}}$ and $\tilde{T}_{\tilde{\mu}}$ as in Eqs. (3.4.1) and (3.4.2), respectively (but with $\alpha = 1$). We let $\lambda_{\tilde{\mu}}$ and $\tilde{\lambda}_{\tilde{\mu}}$ denote the optimal average costs corresponding to the operators $T_{\tilde{\mu}}$ and $\tilde{T}_{\tilde{\mu}}$, respectively. We have the following definition. Let $\epsilon > 0$. An $\epsilon$-optimal policy (for the average cost problems) is a discretized policy $\tilde{\mu} \in \tilde{\Pi}_h$, for some $h > 0$, such that $|\lambda^* - \lambda_{\tilde{\mu}}|$ and $\left|\lambda^* - \tilde{\lambda}_{\tilde{\mu}}\right|$ are both no greater than $\epsilon$.

We follow the derivation for the discounted-cost problem $\mathcal{P}_{\text{mix}}$ (see Section 3.4 for details) and obtain similar results. The main result is that the total complexity of using multigrid successive approximation to compute an $\epsilon$-optimal policy is $O\left(\epsilon^{-(2n+m)}\right)$, which is also the information-based complexity of the problem. Therefore, for problems in $\mathcal{P}_{\text{mix}}$, computing an $\epsilon$-optimal policy is "as hard as" computing an $\epsilon$-optimal average cost.

### 4.3.5 Notes

The computational ill-posedness of continuous-state average cost problems illustrates the technical difficulties of such problems. But if a $k$-stage ergodicity condition is assumed then the problem is well-posed; moreover, we have an optimal algorithm algorithm for problem. However, it seems that there should be weaker conditions for the well-posedness. For example, in finite-state average-cost problems, weaker conditions are known for the existence of solution to the optimality equation [see Bertsekas (1987)]. These conditions are ergodicity conditions on the dynamics of the problem (but are weaker than the $k$-stage ergodicity of Assumption E.3). A problem for future research is to find similar conditions for the well-posedness of continuous-state average cost problems.

Based on the proof of Theorems 4.3.1 and 4.3.2 , we can construct a family of average-cost problems $(P_t, g)$ and another problem $(P, g)$ such that $\|P_t - P\|_\infty \to 0$ as $t \to \infty$ but the average cost $\lambda_t^*$ does not converge to $\lambda^*$. The theorems also illustrate why it is important to normalize the discretized dynamics $\tilde{P}_h$ when Assumption A.4 is in effect. If such a normalization is not performed, the successive approximation algorithm of Section 4.3.3 may not converge.

We are not able to establish Theorem 4.3.1 for all $K$ because if $K$ is sufficiently small then the problem satisfies a 1-stage ergodicity condition.

In the actual implementation of the successive approximation algorithm we can keep $\tilde{T}_h^t J$ bounded by shifting it by a constant after each iteration.

## 4.4 Numerical Results

In this section, we present some simple numerical results comparing multigrid successive approximation with single-grid successive approximation. The purpose of this section is to understand some of the practical issues involved in the implementation of multigrid successive approximation, and not to find the most efficient algorithm for solving MDP's.

Before we discuss the numerical results we first make the following observations.

### 4.4.1 Some Observations

1. Since $J^*$ is a Lipschitz continuous function, a piecewise constant approximation (although is theoretically optimal in our framework) is a bad approximation scheme in practice. For example, if the slope of $J^*$ is 1, then using a piecewise constant approximation scheme would require a grid-size $h \sim 0.01$ to obtain an accuracy $\epsilon = 0.01$. [Actually a much smaller grid-size is needed since $\epsilon \sim h/(1-\alpha)^2$, in general; and $\epsilon \sim h/(1-\alpha)$, with ergodicity.] Therefore, in practice, it is always better to use a linear or higher order interpolation scheme. [See L'Ecuyer (1989) and the references therein for some practical interpolation schemes.]

2. The discretization error bounds of Section 2.4 are always too pessimistic in practice. (More precisely, the theoretical estimates of the constants in the bounds are too pessimistic.)

3. The approximate solution of MDP's is computationally expensive. For example, consider a 1-dimensional control and 1-dimensional state space problem with a discount factor 0.90, and to be solved to an accuracy of 0.01; we have $n = m = 1$, $\alpha = 0.90$, and $\epsilon = 0.01$. The number of grid-points needed is $O\left(1/[(1-\alpha)^2\epsilon]^3\right)$, which is $\sim 10^{12}$ (and with a large constant factor in front of it). If the state space has dimension $n = 2$, then the number of grid-points needed is $\sim 10^{20}$.

4. There are practical limitations on the number of grid-levels a multigrid algorithm can have.

### 4.4.2 A Numerical Example

We use as our test example a discounted-cost *inventory control problem* where daily customers' demands for an inventory are stochastic and the objective is to replenish the inventory at the beginning of each day to meet the expected demands of the day. There is a cost for unsold inventory at the end of a day and for unable to meet customers' demands; moreover, unmet demands are lost. [See Bertsekas (1987) for more details.] We choose this example because it is well-studied and we can check our numerical results with the known properties of the solution.

We consider a problem with state space $S = [0,1]$ and control space $C = [0,1]$. The constraint set is given by $U(x) = [0, 1-x]$, $x \in S$. We assume that the demands are exponentially distributed so that the transition density is as shown in Figure 4.4.1. [In Figure f4.4.1, $x_k$ denotes the inventory level and $u_k$ denotes the replenishment (at the beginning of the $k$-th day). Customers' demands on the $k$-th day are denoted by $w_k$.]

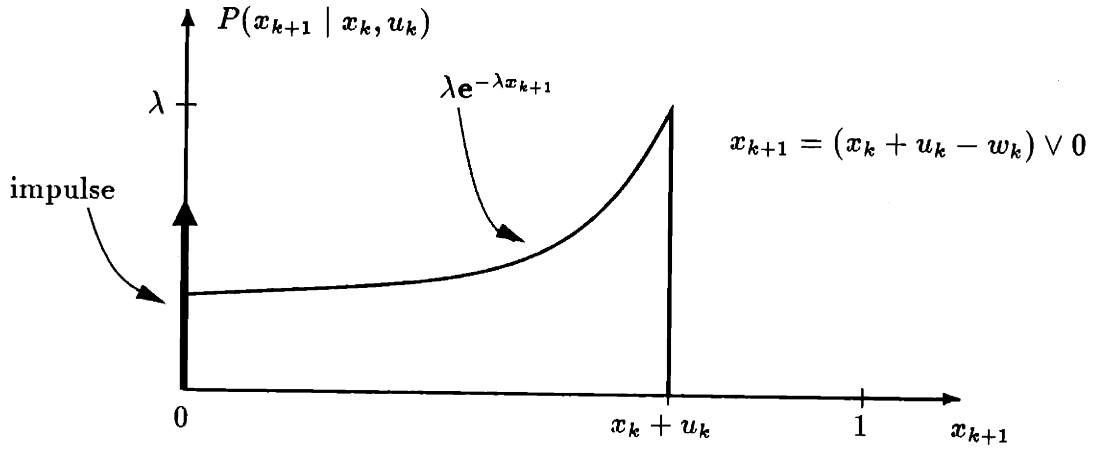
$P(x_{k+1} \mid x_k, u_k)$
$\lambda$
$\lambda e^{-\lambda x_{k+1}}$
$x_{k+1} = (x_k + u_k - w_k) \vee 0$
impulse
0
$x_k + u_k$
1
$x_{k+1}$

**Figure** 4.4.1: the transition density of an inventory control example

Let $g_1(x,u,w) = c_1u + c_2(x+u-w)^2$, where $x, u \in [0,1]$, $w \in [0,\infty)$, and $w \sim \lambda e^{-\lambda w}$. The cost function is given by

$$\begin{aligned} g(x,u) &= \mathrm{E}_w \{g_1(x,u,w)\} \\ &= c_1u + c_2((x+u)^2 - 2(x+u)/\lambda + 2/\lambda^2). \end{aligned}$$

Our choice for these parameters is $c_1 = 2$, $c_2 = 6$ , and $\lambda = 2.5$.

### 4.4.3 Implementation

We implemented the standard (Jacobi) versions of the single-grid and multigrid successive approximation algorithms. For simplicity, we used a uniform grid-size and a piecewise constant approximation of $g$ and $P$. (The "impulse" at $x = 0$ in $P$ was handled separately.)

We did not use the discretization error bounds of Section 2.4 in the algorithms; instead, we simply chose a suitable constant times the grid-size as the discretization error bound. The final grid-level and a threshold $\epsilon_a$ on the successive approximation error bounds on this level determined the stopping criterion of the multigrid algorithm.

### 4.4.4 Results

The results are tabulated in Figure 4.4.2.

$\alpha = 0.9$

| $N$ | $\epsilon_a$ | MG Running times | SG Running Times |
|---|---|---|---|
| 4 | 0.25 | 0.04 (4) | 0.02 (4) |
| 8 | 0.125 | 0.16 (3) | 0.22 (5) |
| 16 | 0.0625 | 1.06 (4) | 1.36 (6) |
| 32 | 0.03125 | 6.66 (4) | 10.16 (7) |
| 64 | 0.015625 | 59.08 (5) | 72.70 (7) |

$\alpha = 0.99$

| N | $\epsilon_a$ | MG Running Times | SG Running Times |
|---|---|---|---|
| 4 | 0.25 | 0.04 (4) | 0.04 (4) |
| 8 | 0.125 | 0.24 (5) | 0.22 (6) |
| 16 | 0.0625 | 1.32 (5) | 1.80 (8) |
| 32 | 0.03125 | 9.88 (6) | 13.16 (9) |
| 64 | 0.015625 | 69.76 (4) | 94.06 (9) |

$\alpha = 0.999$

| N | $\epsilon_a$ | MG Running Times | SG Running Times |
|---|---|---|---|
| 4 | 0.25 | 0.06 (4) | 0.04 (4) |
| 8 | 0.125 | 0.30 (6) | 0.28 (7) |
| 16 | 0.0625 | 1.74 (7) | 2.04 (9) |
| 32 | 0.03125 | 11.36 (7) | 14.68 (10) |
| 64 | 0.015625 | 80.24 (7) | 115.12 (11) |

**Remarks:**

1. $N = 1/h \sim$ maximum number of grid-points per dimension.
   Total number of grid points $\sim N^3$.
2. The parameter $\epsilon_a$ is the successive approximation error at the finest grid-level, which we use as the termination criterion for both successive approximation algorithms.
3. (Total) Running Times are in seconds on a Sun 3/260 with a Motorola MC68881 floating point processor.
4. In parenthesis is the number of iterations at the finest grid-level.

**Figure** 4.4.2: Numerical data showing MG versus SG

**Notes**

1. Our example satisfies Assumptions A.1–A.4 and a 1-stage ergodicity condition (Assumption E.1); moreover, for $x \in (0,1]$, the transition density $P$ satisfies Assumption A.5. Since we handle the "impulse" at $x = 0$ separately, the discretization error bound of Theorem 2.4.3 applies to the example.

2. Since $\|J^*\|_\infty \sim 1/(1-\alpha)$, if the problem satisfies an ergodicity condition, then its percentage error $\|J^* - J^F\|_\infty / \|J^*\|_\infty \sim h$, independent of $\alpha$. [In general, the percentage error is $\sim h/(1-\alpha)$.] Therefore, to get a 1% accuracy in our example, we need $h \sim 0.01$ or $N \sim 100$.

3. The Gauss-Seidel version of the successive approximation algorithm [see Bertsekas (1987) for details] is often used in practice, instead of the standard (Jacobi) version. This is because the former has a better convergence rate than the latter. [But one drawback of the Gauss-Seidel version is that it does not give any "nice" approximation error bound, as in the standard version. So the latter is often incorporated into the algorithm, whenever such error bounds are needed.] We have not implemented the Gauss-Seidel versions of single-grid and multigrid successive approximation algorithms. If we had, we expect the multigrid algorithm would be better than the single-grid algorithm, and the savings would be similar to the Jacobi versions (which we implemented).

### 4.4.5 Conclusions

Multigrid successive approximation is easy to implement and grid-level changes are of negligible computational cost. Multigrid successive approximation improves on single-grid successive approximation by 10–20% in running times. [This is far less than the theoretical estimates, which predict a factor of 2–3 improvement; that is, a factor of $O(|\log[(1-\alpha)\epsilon]|)$.] We expect the savings to be more substantial if the multigrid algorithm is better tuned, and the problem has a larger discount factor and has to be solved at a higher accuracy.

The cost of an iteration of successive approximation at the finest grid-level is about an order of magnitude more than the cost on the preceding grid-level. Thus, the savings in multigrid is directly correlated to the reduction in the number of iterations at the finest grid-level.

We have observed that the policy converges to an optimal policy in the first few iterations. This suggests that a policy-iteration-based multigrid algorithm may perform better in practice.

However, the dominant computational cost is simply from the number of grid-points needed to discretize the problem (not the number of iterations). Therefore, a more promising approach, where more substantial savings in computational costs can be expected, is to use a higher order interpolation scheme. For example, if we use a linear interpolation scheme on a problem that is twice differentiable and has bounded first and second derivatives, then we expect a discretization error bound with $\epsilon = O(h/(1-\alpha)^2)$ in general. And the total number of grid-points needed is $O\left([(1-\alpha)^2\epsilon]^{-(2n+m)/2}\right)$ instead of $O\left([(1-\alpha)^2\epsilon]^{-(2n+m)}\right)$. Even more substantial savings can be expected if the problem has bounded higher derivatives and we use a higher order interpolation scheme.

## 4.5 Conclusions

We now conclude our discussion with a summary of the main open problems in the first part of the report. This discussion also serves to motivate the results of Chapter 5.

The open problems we have are of two types. The first type is related to obtaining the computational complexity of other discrete-time stochastic control models. For example, finding weaker conditions for the well-posedness of average-cost problems (see Section 4.3 for details) and finding conditions to ensure linear discretization error bounds for the f-model (see Section 4.1 for details).

The second type of problems is related to closing the $O(1/(1-\alpha))$ "complexity gap" between the multigrid successive approximation algorithm of Section 3.2.2 and the information-based lower bounds of Section 3.4.3. One approach is to find an algorithm with a better complexity than the multigrid algorithm we have.

We have taken this approach in Section 4.2; we have considered a multigrid policy-iteration algorithm and discussed some unresolved issues related to analyzing the complexity ($\alpha$ dependence) of the algorithm. But it seems unlikely that there exists any algorithm that can close the $O(1/(1-\alpha))$ "complexity gap" without additional assumptions (such as ergodicity or differentiability condition) on the problem. [Note that even for the 1-dimensional Fredholm equation of the second kind (see Section 3.5.1), this problem has not been resolved.] This motivates a different approach—proving another kind of lower bounds.

In Chapter 5, we will prove *algorithmic-based lower bounds;* that is, we will prove that within a certain family of multigrid algorithms, there is no algorithm with a better complexity than the multigrid successive approximation algorithm of Section 3.2.2.

It is intuitively clear that if we consider algorithms that use only the discretized dynamic programming operators $\{T_h\}$, then it is unlikely there exists any algorithm with a better complexity than the multigrid successive approximation of Section 3.2.2. In Chapter 5, we will make precise this intuition. Furthermore, it seems that a multigrid algorithm should be one-way (from coarse to fine). In Section 5.4, we will also show that one-way multigrid algorithms are, in a certain sense, optimal.

Lastly, another question we will address in Chapter 5 is how to apply multigrid algorithms to problems where the discretization error bounds are $O(h^2)$, $O(h^3)$, or $O(h^s)$ for some $s > 0$. Such situations arise when we use higher order interpolation schemes (see Section 4.4.5). The question is how to apply multigrid successive approximation to these problems? Is there a *multigrid principle* for designing optimal multigrid algorithms? We will give an answer to this question in Section 5.2.

# Chapter 5

# Multigrid Algorithms and Complexity Results For General Fixed-Point Problems

In this chapter, we develop an abstract framework for studying and analyzing multigrid algorithms for the approximate solution of a general fixed-point problem. We assume that we can discretize the problem to various levels of accuracy. And at each grid (or discretization) level we have an iterative algorithm for solving the discretized problem. There are three grid-level dependent parameters: (i) the rate of convergence of the iterative algorithm, (ii) the accuracy of the discretized solution, and (iii) the iteration cost of the algorithm. The objective is to find the most cost-efficient way of computing an approximation of the solution to the original problem, by performing iterations on different grid-levels, and at the same time minimize the total computational cost.

We first give an informal overview of the basic results in this chapter. Consider a fixed-point equation

$$Ax = x,$$

where $A$ is a contraction mapping [on a metric space $(\mathsf{M}, \mathsf{d})$] with fixed point $x^*$. Suppose that we can discretize $A$ and obtain a family of fixed-point equations

$$A_h x = x \qquad h \in (0, 1],$$

where $A_h$ is a contraction mapping with fixed point $x_h^*$. (Our convention is that $h$ denotes the grid-size; that is, a smaller $h$ implies a more accurate discretization.)

We make the following assumptions:

1. The discretization error $\mathsf{d}(x^*, x_h^*)$ is bounded by a function $\mathsf{D}(h)$, where $\mathsf{D}(h) \sim h^s$ for some $s > 0$.

2. The contraction factor of $A_h$ is $\alpha_h \sim 1 - h^q$ for some $q \geq 0$.

3. The cost of an iteration of $A_h$ is $\mathsf{C}(h) \sim h^{-r}$ for some $r \geq 0$.

We are interested in computing an $\epsilon$-approximation of $x^*$, using only the contraction mappings $\{A_h\}$ and, at the same time, minimizing the total computational cost.

We show (in Section 5.2.2) that the complexity of a single-grid algorithm is $\mathrm{O}\left(\ln(1/\epsilon)[1/\epsilon]^{(q+r)/s}\right)$ and (in Section 5.2.3) that the complexity of a one-way multigrid algorithm is $\mathrm{O}\left([1/\epsilon]^{(q+r)/s}\right)$. We show the optimality of the one-way multigrid algorithm by proving an algorithmic-based lower bound (see Section 5.2.4). We also show the optimality of one-way multigrid algorithms in general, under certain monotonicity conditions on $\alpha_h$ and $\mathsf{C}(h)$ (see Section 5.4).

We introduce a number of extensions to the basic results. We let $\alpha_h = 1 - f_1 h^q$, $\mathsf{C}(h) = f_2 h^{-r}$, and $\mathsf{D}(h) = f_3 h^s$, for some positive constants $f_1$, $f_2$, and $f_3$; we also analyze the dependence of the algorithms on these constants. We introduce this extension to model (see Section 5.3.1) the discount-factor dependence of the discrete-time stochastic control problem of Chapters 2–3.

To model an even more general class of problems, we introduce the notion of a "contraction process", which has the property that for all $x$,

$$\mathsf{d}(x_h^*, A_h^t x) \le \alpha_h^t a \mathsf{d}(x_h^*, x), \qquad \forall t \ge 0,$$

where $a$ is the delay factor. The delay factor allows us to model algorithms with eventual geometric convergence but possibly unpredictable initial behavior. This extension is needed to model the simulated annealing problem (see Section 5.3.2). [However, for the most of Chapter 5, the delay factor can be ignored (by letting it be 1).]

The presentation in this chapter is especially formal because we are proving a new type of lower bounds: algorithmic-based lower bounds. The interpretations of these lower bounds are quite specific to the framework, therefore it is important to define the framework precisely.

An outline of this chapter is as follows: In Section 5.1, we introduce the notion of contraction processes. They are generalizations of contraction mappings and allow us to model different types of iterative algorithms. We then develop the general framework and introduce a minimum computational cost problem. This minimum cost problem models the problem of finding the most cost-efficient way of computing an approximate solution to the fixed-point problem. In Section 5.2, we consider some special cases of the minimum cost problems, propose a single-grid algorithm and a multigrid algorithm for them. We show that multigrid is an improvement of the single-grid algorithm, and is in a certain sense optimal. In Section 5.3, we apply the algorithms and the analyses of Section 5.2 to various practical problems; in particular, we apply the results to the fixed-point problem studied in the first part of this report. In Section 5.4, we prove some general results about the framework. In Section 5.5, we state our conclusions and suggest future research.

## 5.1 A Minimum Computational Cost Problem

In this section we develop an abstract framework for studying multigrid algorithms. This is the framework we will use throughout the chapter.

An outline of the section is as follows. First, we introduce the notion of contraction processes; they are generalizations of contraction mappings and model the behavior of many algorithms. Second, we develop a framework for studying multigrid algorithms and introduce a minimum computational cost problem. Third, we formalize the objective of the minimum computational cost problem. Finally, we close the section with some observations.

## 5.1.1 Contraction Processes

In this subsection we introduce the notion of contraction processes. We begin by introducing the basic notation of this chapter.

### Basic Notation

Let $\mathbf{Z}^0$ (respectively, $\mathbf{Z}^+$) denote the set of non-negative (respectively, positive) integers; let $\mathbf{R}^0$ (respectively, $\mathbf{R}^+$) denote the set of non-negative (respectively, positive) real numbers. We use $\mathcal{T}$ (respectively, $\mathcal{T}^+$) to denote either $\mathbf{Z}^0$ or $\mathbf{R}^0$ (respectively, either $\mathbf{Z}^+$ or $\mathbf{R}^+$). If $x$ and $y$ are real numbers, then $x \wedge y$ (respectively, $x \vee y$) denotes the minimum (respectively, maximum) of $x$ and $y$.

Let $(\mathsf{M}, \mathsf{d})$ denote a metric space with metric $\mathsf{d}$. Elements of $\mathsf{M}$ are called *points.* Throughout this chapter, the pair $(\mathsf{M}, \mathsf{d})$ will denote a given metric space and the word "points" is used exclusively for elements of $\mathsf{M}$.

### The Definition of a Contraction Process

A mapping $A : \mathcal{T} \times \mathsf{M} \mapsto \mathsf{M}$ is called a *contraction process* on $(\mathsf{M}, \mathsf{d})$ if there exists an $\alpha \in [0,1)$, an $a \in [1, \infty)$, and a (unique) point $x^*$, such that

$$\mathsf{d}\left(x^*, A(t,x)\right) \leq \alpha^t a \mathsf{d}(x^*, x), \qquad \forall t \in \mathcal{T}, x \in \mathsf{M}. \tag{5.1.1}$$

We call $\alpha$ the *contraction factor,* $a$ the *delay factor,* and $x^*$ the *fixed point* of $A$. [Note that the uniqueness of $x^*$ follows from Eq. (5.1.1).] If $\mathcal{T} = \mathbf{Z}^0$ (respectively, $\mathcal{T} = \mathbf{R}^0$), we call $A$ a discrete-time (respectively, continuous-time) contraction process.

A contraction process can be viewed as a family of mappings $\{A(t, \cdot)\}_t$ from $\mathsf{M}$ into itself, parametrized by $t \in \mathcal{T}$. For brevity, we will write $A^t x$ instead of $A(t,x)$. When a contraction process is applied to an *initial point* $x$, it generates a family of points $(A^t x)_{t \in \mathcal{T}}$ which (eventually) converges to $x^*$ geometrically with rate $\alpha$. The delay factor $a$ allows a contraction process to model iterative algorithms with unpredictable initial behaviors. In general, the delay factor may depend on the initial point $x$.

### Composition of Contraction Processes

We now consider the composition of contraction processes. Let $A_1$ and $A_2$ be two contraction processes on $(\mathsf{M}, \mathsf{d})$. For any $t_1, t_2 \in \mathcal{T}$, we define a mapping $A_1^{t_1} A_2^{t_2}$ :

$\mathsf{M} \mapsto \mathsf{M}$, called a *composition* of $A_1$ and $A_2$, by letting

$$A_1^{t_1} A_2^{t_2} x = A_1^{t_1}(A_2^{t_2} x), \quad \forall x \in \mathsf{M}.$$

More generally, for any contraction processes $A_1, A_2, \ldots, A_i$ on $\mathsf{M}$ and any $t_1, t_2, \ldots, t_i \in \mathcal{T}$, we can form the composition $A_1^{t_1} A_2^{t_2} \cdots A_i^{t_i} : \mathsf{M} \mapsto \mathsf{M}$, where the order of composition is always right associative. If $A'$ and $A''$ are two mappings from $\mathsf{M}$ into itself, we write $A' = A''$ iff $A'x = A''x$ for all $x = \mathsf{M}$.

## An Example

We now look at an example of a (discrete-time) contraction process—a contraction mapping. Let $T$ be a contraction mapping on $(\mathsf{M}, \mathsf{d})$ with contraction factor $\alpha$ (see Section 2.1.3). If the fixed point of $T$, denoted by $x^*$, exists (which is always the case when $\mathsf{M}$ is complete), then $T$ satisfies

$$\mathsf{d}(x^*, T^t x) \leq \alpha^t \mathsf{d}(x^*, x), \qquad \forall t \in \mathbf{Z}^0, x \in \mathsf{M},$$

where $T^t$ denotes the composition of $t$ copies of $T$. Thus, $T$ is a contraction process with delay factor 1.

However, if $T$ is a $k$-stage contraction mapping with contraction factor $\alpha$ (see Section 2.1.3), then

$$\mathsf{d}(x^*, T^t x) \leq \alpha^{t/k} \alpha^{-1} \mathsf{d}(x^*, x), \qquad \forall t \in \mathbf{Z}^0, x \in \mathsf{M}.$$

Here, $T$ is a contraction process with contraction factor $\alpha^{1/k}$ and delay factor $\alpha^{-1}$.

## Summary

Discrete-time contraction processes are generalizations of contraction mappings. [Similarly, it is seen that continuous-time contraction processes are generalizations of (continuous-time) semi-groups of contraction mappings.] The reasons for considering contraction processes are as follows: (i) A contraction process "contracts" the distance between the fixed point and any point, whereas a contraction mapping "contracts" the distance between any two points. The latter may not be true or is hard to verify in some cases. (ii) A contraction process can model algorithms with "memory", whereas a contraction mapping is "memoryless". For example, if $T$ is a contraction mapping, then $T^{t+1}x$ depends only on the previous point $T^t x$; in contrast, for a contraction process $A$, the point $A^{t+1}x$ may possibly depend on all earlier points, $A^t x$, ..., $A^1 x$, $x$. (iii) Delay factors allow the general case of eventual geometric convergence, whereas a contraction mapping contracts $\alpha$ at each step.

## 5.1.2 The General Framework

We now introduce a framework for studying multigrid algorithms that arise in the approximate solution of a certain fixed-point problem. We assume that we can discretize the problem at various grid-levels so that at each grid-level we have a discretized problem; furthermore, we have an iterative algorithm for approximating the solution (of the discretized problem) at that grid-level. The salient feature of the framework is that the following parameters are grid-dependent: (i) the rate of convergence of the iterative algorithm, (ii) the iteration cost of the algorithm, and (iii) the distance between the solutions of the discretized problem and the original problem.

A multigrid algorithm computes an approximation of the original problem to within some specified accuracy. This algorithm uses algorithms at different grid-levels for the computation. The objective is to choose the grid-levels and the duration at each grid-level so that we obtain the desired approximation. We are interested in finding a multigrid algorithm that compute the approximation and at the same time minimize the total cost incurred.

We now present the general framework to model this minimum computational cost problem.

### A Family of Contraction Processes

Let $\mathcal{H}$ be a subset of $(0,1]$. The parameter $h \in \mathcal{H}$ denotes the *grid-size* or *grid-level,* with the convention that a smaller $h$ represents a finer grid.

We model the algorithms at various grid-levels by a family of contraction processes $\{A_h\}_{h\in\mathcal{H}}$. Each $A_h$ is a contraction process on $(\mathsf{M}, \mathsf{d})$ with contraction factor $\alpha_h$ and fixed point $x_h^*$. The contraction factor $\alpha_h$ is a function of the grid-level, and the fixed point $x_h^*$ denotes the solution of the discretized problem at grid-level $h$. We let $x^* \in \mathsf{M}$ denote the solution to the original problem. And we adopt the convention that $x_0^* \overset{\text{def}}{=} x^*$ and $\mathcal{H}_0 \overset{\text{def}}{=} \mathcal{H} \cup \{0\}$.

We also introduce delay factors* in the contraction processes. The delay factors are described by a function $a : \mathcal{H} \times \mathcal{H}_0 \mapsto [1, \infty)$. And for all $t \in \mathcal{T}$, $h \in \mathcal{H}$, $h' \in \mathcal{H}_0$, $x \in \mathsf{M}$ the following bounds hold:

$$\mathsf{d}(x_h^*, A_h^t x_{h'}^*) \leq \alpha_h^t a(h,h') \mathsf{d}(x_h^*, x_{h'}^*); \tag{5.1.2}$$

$$\mathsf{d}(x_h^*, A_h^t A_{h'}^t x) \leq \alpha_h^t a(h,h') \mathsf{d}(x_h^*, A_{h'}^t x). \tag{5.1.3}$$

[And for general $x \in \mathsf{M}$, $\mathsf{d}(x_h^*, A_h^t x) \leq \alpha_h^t \sup_{h'\in\mathcal{H}_0}\{a(h,h')\}\mathsf{d}(x^*, x)$.] The delay factor $a(h,h')$ allows us to model various interactions between different grid-levels (see Section 5.1.4). For brevity, we will write $a_h^{h'}$ instead of $a(h,h')$.

### Iteration Cost and Grid-Transfer Error Bound

To model the computational cost, we assume that to use $A_h$ for a *duration* of $t \in \mathcal{T}^+$ (that is, applying $A_h^t$) incurs a cost of $t\mathsf{C}(h)$. Here, $\mathsf{C} : \mathcal{H} \mapsto \mathbf{R}^0$ is the *iteration cost.*

To bound the discretized solution and the original solution, we assume that

$$\mathsf{d}(x^*, x_h^*) \leq \mathsf{D}(h), \qquad \forall h \in \mathcal{H}, \tag{5.1.4}$$

*Delay factors are needed to model simulated annealing (Section 5.3.2). For the most of our discussion, delay factors can be ignored by letting them be 1.

where $\mathsf{D} : \mathcal{H} \mapsto \mathbf{R}^0$. We call $\mathsf{d}(x^*, x_h^*)$ the *discretization error* and $\mathsf{D}(h)$ the *discretization error bound.* We adopt the convention that $\mathsf{D}(0) \overset{\text{def}}{=} 0$.

More generally, we assume that

$$\mathsf{d}(x_h^*, x_{h'}^*) \leq \mathsf{B}(h, h'), \qquad \forall h, h' \in \mathcal{H}_0, \tag{5.1.5}$$

where $\mathsf{B} : \mathcal{H}_0 \times \mathcal{H}_0 \mapsto \mathbf{R}^0$ and $\mathsf{B}(0, h) \overset{\text{def}}{=} \mathsf{D}(h)$. We call $\mathsf{d}(x_h^*, x_{h'}^*)$ the *grid-transfer error* and $\mathsf{B}(h, h')$ the *grid-transfer error bound.*

Since $\mathsf{d}$ is a metric, we have

$$\begin{aligned} \mathsf{d}(x_h^*, x_{h'}^*) &\leq \mathsf{d}(x_h^*, x^*) + \mathsf{d}(x^*, x_{h'}^*) \\ &\leq \mathsf{D}(h) + \mathsf{D}(h'). \end{aligned} \tag{5.1.6}$$

Therefore, we can assume that

$$\mathsf{B}(h, h') \leq \mathsf{D}(h) + \mathsf{D}(h') = \mathsf{B}(h, 0) + \mathsf{B}(0, h'). \tag{5.1.7}$$

More generally, we assume that $\mathsf{B}$ is a metric on $\mathcal{H}_0$; namely, for all $h, h', h'' \in \mathcal{H}_0$, $\mathsf{B}$ satisfies the triangle inequality,

$$\mathsf{B}(h, h') \leq \mathsf{B}(h, h'') + \mathsf{B}(h'', h'); \tag{5.1.8}$$

is symmetric, $\mathsf{B}(h, h') = \mathsf{B}(h', h)$; and satisfies $\mathsf{B}(h, h) = 0$.

## Problems and Instances

Finally, a *Minimum Computational Cost Problem (MCCP)* is specified by

$$\left(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h, h'), \mathsf{C}(h)\right),$$

where $h, h' \in \mathcal{H}_0$. If $\mathcal{T} = \mathbf{Z}^0$ (respectively, $\mathcal{T} = \mathbf{R}^0$), then the problem is a discrete-time (respectively, a continuous-time) problem.

An MCCP, denoted by $\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h, h'), \mathsf{C}(h))$, actually represents a class of instances. We call $\iota = \left(\mathsf{M}, \mathsf{d}, \{A_h\}_{h \in \mathcal{H}}, \{x_h^*\}_{h \in \mathcal{H}_0}\right)$ an *instance* of the MCCP, written $\iota \in \mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h, h'), \mathsf{C}(h))$, if $\iota$ satisfies the following requirements:

1. For every $h \in \mathcal{H}$, $A_h$ is a contraction process on $(\mathsf{M}, \mathsf{d})$ with fixed point $x_h^*$, contraction factor $\alpha_h$, and delay factor $a_h^{h'}$ that satisfies Eqs. (5.1.2) and (5.1.3).

2. For every $h, h' \in \mathcal{H}_0$, the distance $\mathsf{d}(x_h^*, x_{h'}^*)$ is bounded by $\mathsf{B}(h, h')$; that is, Eq. (5.1.5) holds.

### The Objective

Let $h, h_0 \in \mathcal{H}_0$. For any $\epsilon > 0$, an $x \in \mathsf{M}$ is called an $\epsilon$-approximation of $x_h^*$, if $\mathsf{d}(x_h^*, x) \leq \epsilon$. In an MCCP we are interested in computing an $\epsilon$-approximation of $x^*$, starting at some initial point $x_{h_0}^*$. We use the contraction processes for the computation. The problem is to choose the grid-levels and a duration at each grid-level (for using the contraction process) so that we eventually obtain the desired approximation of $x^*$. The objective is to choose the grid-levels and the durations to minimize the total computation cost incurred. We will give a more precise formulation of the problem's objective after we define the notion of an algorithm.

## 5.1.3 Problem Definition

In this subsection we define the objective of an MCCP. We first introduce the notion of "control trajectories", which are used to described algorithms. We then formalize the notion of algorithms and define the objective. Finally, we introduce a way of bounding distances that is valid for all instances.

For the rest of this subsection, let us fix an MCCP

$$\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h)).$$

### Control Trajectories

We can compute an $\epsilon$-approximation of $x^*$ using the family of contraction processes. The reason for introducing "control trajectories" is that we can think of "controlling" the contraction processes through a choice of grid-levels and durations on the grid-levels; each choice corresponds to a control trajectory.

More formally, let $h_1, h_2, \ldots, h_i \in \mathcal{H}$ denote some $i$ choices of grid-levels; let $t_1, t_2, \ldots, t_i \in \mathcal{T}^+$ denote the corresponding durations; and let

$$\eta = \Big((t_i, h_i), (t_{i-1}, h_{i-1}), \ldots, (t_1, h_1)\Big). \tag{5.1.9}$$

[The indices are in a reversed order for a reason which will be explained later.] Such a sequence is called a *control trajectory* (or simply a *trajectory*). The *length* of $\eta$, denoted by $|\eta|$, is $i$. The trajectory is *multigrid* if $|\eta| \geq 2$; otherwise, it is *single-grid.*

The set of *grid-levels* of $\eta$ is $\{h_i, h_{i-1}, \ldots, h_1\}$; its *initial grid-level* is $h_1$ and its *final grid-level* is $h_i$. We call $\eta$ *one-way* if $h_i \leq h_{i-1} \leq \cdots \leq h_1$, namely, the grid-levels only go from coarse to fine; it is *strictly one-way* if all the inequalities are strict.

Furthermore, we use the notation $A_{\mathcal{H}}^{\eta}$ to represent the composition $A_{h_i}^{t_i} A_{h_{i-1}}^{t_{i-1}} \cdots A_{h_1}^{t_1}$. [This is why we have written the indices in Eq. (5.1.9) in the reversed order.] We say that $A_{\mathcal{H}}^{\eta} x$ is the result of applying $\eta$ to the initial point $x$. The *total cost* of using $\eta$ is

$$\mathsf{C}_{\mathcal{H}}^{\eta} \overset{\text{def}}{=} \sum_{j=1}^{|\eta|} t_j \mathsf{C}(h_j). \tag{5.1.10}$$

Finally, let

$$\mathcal{A} \overset{\text{def}}{=} \bigcup_{i=1}^{\infty} \left(\mathcal{T}^{+} \times \mathcal{H}\right)^{i} \quad \text{and} \quad \mathcal{A}_0 \overset{\text{def}}{=} \mathcal{A} \cup \{0\}, \tag{5.1.11}$$

where 0 denotes the zero-length trajectory, which is called the *null trajectory.* For the null trajectory we have $\mathsf{C}_{\mathcal{H}}^{0} = 0$ and $A_{\mathcal{H}}^{0} x = x$ for all $x \in \mathsf{M}$. Suppose that

$$\tilde{\eta} = \Big((t'_j, h'_j), (t'_{j-1}, h'_{j-1}), \ldots, (t'_1, h'_1)\Big).$$

The *composition* of $\tilde{\eta}$ and $\eta$ results in a new trajectory given by

$$\tilde{\eta} \cdot \eta = \Big((t'_j, h'_j), \ldots, (t'_1, h'_1), (t_i, h_i), \ldots, (t_1, h_1)\Big).$$

It is clear that

$$A_{\mathcal{H}}^{\tilde{\eta}\cdot\eta} = A_{\mathcal{H}}^{\tilde{\eta}} A_{\mathcal{H}}^{\eta} \quad \text{and} \quad \mathsf{C}_{\mathcal{H}}^{\tilde{\eta}\cdot\eta} = \mathsf{C}_{\mathcal{H}}^{\tilde{\eta}} + \mathsf{C}_{\mathcal{H}}^{\eta}, \qquad \forall \tilde{\eta}, \eta \in \mathcal{A}_0.$$

Therefore, control trajectories provide a concise notation of describing the compositions of contraction processes and their associated costs.

## Algorithms

We now give a precise definition of an algorithm. By an algorithm for an MCCP that computes an $\epsilon$-approximation to $x^*$, we mean a prescription, for each $\epsilon > 0$, of a trajectory which, when used to operate the contraction processes, gives an $\epsilon$-approximation to $x^*$.[†] We assume that the parameters $(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h, h'), \mathsf{C}(h))$ are also available to the algorithm.

More generally, for any $h, h_0 \in \mathcal{H}_0$, an *algorithm* that computes an $\epsilon$-approximation to $x_h^*$, starting from an initial point $x_{h_0}^*$, is a mapping $\gamma : \mathbf{R}^+ \mapsto \mathcal{A}_0$ such that for all instances and all $\epsilon > 0$,

$$\mathsf{d}(x_h^*, A_{\mathcal{H}}^{\gamma(\epsilon)} x_{h_0}^*) \leq \epsilon.$$

The set of all such algorithms is denoted by $G(x_{h_0}^*, x_h^*)$.

An algorithm $\gamma$ is said to be *single-grid* if $|\gamma(\epsilon)| \leq 1$ for all $\epsilon$; otherwise, it is *multigrid.* We call $\gamma$ a (*strictly*) *one-way algorithm* if $\gamma(\epsilon)$ is a (strictly) one-way trajectory for all $\epsilon$. The total cost of using $\gamma$ is a function of $\epsilon$ denoted by

$$\mathsf{C}^{\gamma}(\epsilon) \overset{\text{def}}{=} \mathsf{C}_{\mathcal{H}}^{\gamma(\epsilon)}, \quad \epsilon > 0.$$

[The function $\mathsf{C}^{\gamma}(\cdot)$ is the *complexity* of $\gamma$, and we have omitted the subscript $\mathcal{H}$ since there is no ambiguity.]

[†] In general, such trajectories may not exist. Therefore, an assumption such as 0 is a limit point of $\mathcal{H}$ and $\lim_{h\downarrow 0} \mathsf{D}(h) = 0$ is needed to ensure their existence. Indeed, this assumption is sufficient to guarantee the existence of the desired trajectories—namely, choose an $h$ such that $\mathsf{D}(h) \leq \epsilon/2$ and apply $A_h$ to the initial point until the successive approximation error is also less than $\epsilon/2$—the single-grid method.

The objective is to find an algorithm with the "best complexity" in the limit as $\epsilon \searrow 0$. To this effect we introduce the *complexity* of the MCCP

$$\mathsf{C}^*(x^*_{h_0}, x^*_h; \epsilon) \stackrel{\text{def}}{=} \inf_{\gamma \in G(x^*_{h_0}, x^*_h)} \mathsf{C}^\gamma(\epsilon), \qquad h_0, h \in \mathcal{H}_0, \epsilon > 0,$$

which is a lower bound on the complexity of any algorithm. For the remaining of this subsection, we introduce a way of bounding $\mathsf{d}(x^*_h, A^\eta_{\mathcal{H}} x^*_{h_0})$ that is valid for all instances.

## The Basic Bound

For any $\eta \in \mathcal{A}$, $h_0, h \in \mathcal{H}_0$, we would like to bound $\mathsf{d}(x^*_h, A^\eta_{\mathcal{H}} x^*_{h_0})$ for all instances. We can achieve this by using the triangle inequality and what we know about the general problem. We now describe a particular way of doing this.

Using the triangle inequality, the grid-transfer error bound, and the contraction property of $A_h$, we obtain

$$\begin{aligned}\mathsf{d}(x^*_h, A^{t_i}_{h_i} A^{t_{i-1}}_{h_{i-1}} \cdots A^{t_1}_{h_1} x^*_{h_0}) &\le \mathsf{d}(x^*_h, x^*_{h_i}) + \mathsf{d}(x^*_{h_{i-1}}, A^{t_i}_{h_i} A^{t_{i-1}}_{h_{i-1}} \cdots A^{t_1}_{h_1} x^*_{h_0}) \\ &\le \mathsf{B}(h, h_i) + \alpha^{t_i}_{h_i} a^{h_{i-1}}_{h_i} \mathsf{d}(x^*_{h_{i-1}}, A^{t_{i-1}}_{h_{i-1}} \cdots A^{t_1}_{h_1} x^*_{h_0}). \quad (5.1.12)\end{aligned}$$

By recursively expanding the right-hand side of Eq. (5.1.12), we obtain an upper bound on $\mathsf{d}(x^*_h, A^{t_i}_{h_i} A^{t_{i-1}}_{h_{i-1}} \cdots A^{t_1}_{h_1} x^*_{h_0})$ in terms of $\alpha^t_h$'s, $a^{h'}_h$'s, and $\mathsf{B}(h, h')$'s.

To facilitate the writing of this upper bound, let

$$\eta = ((t_{i-1}, h_{i-1}), \ldots, (t_1, h_1)),$$

and define a mapping $\mathsf{P} : \mathcal{H}_0 \times \mathcal{A}_0 \times \mathcal{H}_0 \times \mapsto \mathbf{R}^0$ recursively as follows

$$\mathsf{P}(h, ((t_i, h_i)) \cdot \eta, h_0) \stackrel{\text{def}}{=} \mathsf{B}(h, h_i) + \alpha^{t_i}_{h_i} a^{h_{i-1}}_{h_i} \mathsf{P}(h_i, \eta, h_0); \qquad (5.1.13)$$

$$\mathsf{P}(h, 0, h_0) \stackrel{\text{def}}{=} \mathsf{B}(h, h_0). \qquad (5.1.14)$$

We now introduce a shorthand notation for writing $\mathsf{P}$; this notation also serves as a reminder that $\mathsf{P}(h, \eta, h_0)$ is an upper bound on $\mathsf{d}(x^*_h, A^\eta_{\mathcal{H}} x^*_{h_0})$. We let

$$\mathsf{p}(x^*_h, A^\eta_{\mathcal{H}} x^*_{h_0}) \stackrel{\text{def}}{=} \mathsf{P}(h, \eta, h_0), \qquad h, h_0 \in \mathcal{H}_0, \eta \in \mathcal{A}_0. \qquad (5.1.15)$$

Therefore, it follows that

$$\mathsf{p}(x^*_h, A^{((t_i,h_i))\cdot\eta}_{\mathcal{H}} x^*_{h_0}) = \mathsf{p}(x^*_h, x^*_{h_i}) + \alpha^{t_i}_{h_i} a^{h_{i-1}}_{h_i} \mathsf{p}(x^*_{h_i}, A^\eta_{\mathcal{H}} x^*_{h_0}); \qquad (5.1.16)$$

$$\mathsf{p}(x^*_h, x^*_{h_0}) = \mathsf{B}(h, h_0). \qquad (5.1.17)$$

[We can view Eqs. (5.1.16)–(5.1.17) as an alternative definition of $\mathsf{p}$.] Therefore we have a particular way of upper bounding $\mathsf{d}(x^*_h, A^\eta_{\mathcal{H}} x^*_{h_0})$ that is independent of the instance. We call $\mathsf{p}(x^*_h, A^\eta_{\mathcal{H}} x^*_{h_0})$ the *basic bound* for $\mathsf{d}(x^*_h, A^\eta_{\mathcal{H}} x^*_{h_0})$. Note that $\mathsf{d}(x^*_h, A^\eta_{\mathcal{H}} x^*_{h_0}) \le \epsilon$ holds for all instances, if $\mathsf{p}(x^*_h, A^\eta_{\mathcal{H}} x^*_{h_0}) \le \epsilon$.

Finally, comparing Eq (5.1.12) with Eqs. (5.1.13)–(5.1.17), we summarize our results so far by the following lemma.

**Lemma 5.1.1** *For any instance $\iota \in \mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a^{h'}_h, \mathsf{B}(h, h'), \mathsf{C}(h))$ there holds,*

$$\mathsf{d}(x^*_h, A^\eta_{\mathcal{H}} x^*_{h_0}) \le \mathsf{p}(x^*_h, A^\eta_{\mathcal{H}} x^*_{h_0}), \quad \forall h_0, h \in \mathcal{H}, \eta \in \mathcal{A}_0.$$

### 5.1.4 Notes

First, note that we make an MCCP $\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h))$ "harder" (by enlarging it) if we replace $\alpha_h$ by a "larger" function. For example, consider another MCCP $\hat{\mathcal{P}}(\mathcal{T}, \mathcal{H}, \hat{\alpha}_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h))$ where $\alpha_h \leq \hat{\alpha}_h$ for all $h \in \mathcal{H}$. Then an algorithm $\gamma$ for the second problem [namely, $\gamma \in \hat{G}(x_h^*, x_{h'}^*)$] is also an algorithm for the first problem [namely, $\gamma \in G(x_h^*, x_{h'}^*)$]; thus, $\hat{G}(\hat{x}_h^*, \hat{x}_{h'}^*) \subset G(x_h^*, x_{h'}^*)$. It follows that the complexities of the two problems satisfy

$$\mathsf{C}^*(x_h^*, x_{h'}^*; \epsilon) \leq \hat{\mathsf{C}}^*(x_h^*, x_{h'}^*; \epsilon), \qquad \forall h, h' \in \mathcal{H}_0.$$

Similarly, we can also make $\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h))$ "harder" (thereby increasing the problem's complexity) by making any of the following changes: (i) replace $\mathcal{T} = \mathbf{R}^0$ by $\mathcal{T} = \mathbf{Z}^0$, (ii) replace $\mathcal{H}$ by a subset, (iii) replace $a_h^{h'}$by a larger function, or (iv) replace $\mathsf{B}(h,h')$ by a larger function. We can also increase the complexity by replacing $\mathsf{C}(h)$ by a larger function.

Second, we let an MCCP be specified by $\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h))$ because it is supposed to reflect what we know about the problem. In many cases of interest, we also know the metric space $(\mathsf{M}, \mathsf{d})$. For these cases we denote the class of instances by

$$\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h) \mid \mathsf{M}, \mathsf{d}),$$

and an instance $\iota = \left(\{A_h\}_{h\in\mathcal{H}}, \{x_h^*\}_{h\in\mathcal{H}_0}\right)$ satisfies the same requirements [Eqs. (5.1.2), (5.1.3), and (5.1.5)], except that $(\mathsf{M}, \mathsf{d})$ is specified by the problem. Thus, it is trivially true that

$$\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h) \mid \mathsf{M}, \mathsf{d}) \subset \mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h)).$$

When we develop algorithms for an MCCP, we want the algorithm to be applicable to the largest class of instances. On the other hand, when we derive lower bounds for the complexity of an MCCP, the lower bound is strongest if it belongs to the most restrictive class.

Finally, the delay factor $a_h^{h'}$ can model diverse applications. For example, in certain applications the metric space $\mathsf{M}$ is decomposed into a family of subsets $\{\mathsf{M}_h\}_h$. The contraction processes $A_h$ is only defined on $\mathsf{M}_h$. In order to change from one grid-level to another, there is a grid-transfer mapping $I_h^{h'} : \mathsf{M}_{h'} \mapsto \mathsf{M}_h$, so that we use $A_h^t I_h^{h'} A_{h'}^{t'}$ instead of $A_h^t A_{h'}^{t'}$. There may be computational costs or extra errors associated with using the grid-transfer mapping. The general framework can model such applications as follows:

1. Make the grid-transfer mapping implicit; that is, if $x \in \mathsf{M}_{h'}$, then $A_h^t x$ will implicitly mean $A_h^t I_h^{h'} x$.

2. Use the delay factors to model (multiplicative) errors introduced by the grid-transfer mapping; for example, $\mathsf{d}(x_h^*, I_h^{h'} x) \leq a_h^{h'} \mathsf{d}(x_h^*, x)$. [Cf. Eqs. (5.1.2)–(5.1.3).]

3. Use the delay factors to model the cost of using $I_h^{h'}$; for example, choose $a_h^{h'}$ so that $\alpha_h^t a_h^{h'} = 1$, in which case the cost of using $I_h^{h'}$ is essentially $t\mathsf{C}(h)$.

## 5.2 Algorithms and Their Complexities

In this section, we consider three special cases of the Minimum Computational Cost Problems (MCCP's). We first introduce the assumptions and define the three cases. We then introduce two algorithms for these problems: a single-grid and a multigrid algorithm. We analyze the complexity (computational cost) of these algorithms and obtain upper bounds on the problems' complexities (minimum computational cost). We next establish lower bounds on the problems' complexities and show that the multigrid algorithm is in a certain sense optimal. Finally, we discuss the application of Bellman's optimality principle to MCCP's.

### 5.2.1 Special Cases

We consider three special cases of the MCCP $\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h))$ which model many practical problems of interest. We begin by stating the assumptions for the first case.

**Assumptions**

**S.1** $\mathcal{T} = \mathbf{Z}^0$.

**S.2** $\mathcal{H} = \{1/2^i \mid i \in \mathbf{Z}^0\}$.

**S.3** $\alpha_h = \mathrm{e}^{-f_1 h^q}, \forall h \in \mathcal{H}$, where $f_1 \in (0,1]$ and $q \geq 0$ are constants independent of $h$.

**S.4** $a_h^{h'} \leq a_0, \forall h, h' \in \mathcal{H}$, where $a_0 \geq 1$ is a constant independent of $h, h'$.

**S.5** $\mathsf{B}(h,h') = f_3 h^s + f_3 h'^s, \forall h \neq h' \in \mathcal{H}_0$ $[\mathsf{B}(h,h) = 0, \forall h \in \mathcal{H}_0]$, where $f_3 \geq 1$ and $s > 0$ are constants independent of $h$ and $h'$.

**S.6** $\mathsf{C}(h) = f_2 h^{-r}, \forall h \in \mathcal{H}$, where $f_2 > 0, r \geq 0$ are constants independent of $h$.

We now discuss each of the above assumptions. By letting $\mathcal{T} = \mathbf{Z}^0$ in Assumption S.1, we are choosing a discrete-time MCCP instead of an "easier" continuous-time problem (see Section 5.1.4). Assumption S.2 says that we have a family of increasingly fine grid-levels with grid-size at each level half that of the preceding level. Assumption S.3 says that the contraction factor increases to 1 as the grid-size goes to 0. We have restricted the value of $f_1 \leq 1$ because we are primarily interested in the case where the contraction factor is large (close to 1). (A larger contraction factor makes the problem "harder".) Assumption S.4 bounds the delay factor by a constant. Assumption S.5 says that the grid-transfer error goes to 0 as $h \to h'$; in particular, it says that the discretization error bound $\mathsf{D}(h) = \mathsf{B}(0,h) = f_3 h^s$ goes to 0 as the grid-size becomes smaller. (It is easy to check that $\mathsf{B}$ is a metric on $\mathcal{H}_0$.) In this assumption we have restricted $f_3 \geq 1$ because we are primarily interested in the case where the discretization error is not too small. (A smaller discretization error only makes the problem "easier".) Finally, Assumption S.6 says that the iteration cost increases with decreasing grid-size.

## Three Types of MCCP's

We now define the three special cases of MCCP's. For the first case (called Type I), Assumptions S.1–S.6 are in effect. A Type I MCCP is specified by $(f_1, f_2, f_3, q, r, s, a_0)$ and is denoted by

$$\mathcal{P}_{\mathrm{I}}(f_1, f_2, f_3, q, r, s, a_0),$$

where we have used the subscript I to specify that it is a Type I problem.

For the second case (called Type II), we use the same assumptions as before except that we replace S.3 by

**S.3′** $\alpha_h = (1 - f_1 h^q), \forall h \in \mathcal{H}$ where $f_1 \in (0,1], q \geq 0$ are constants independent of $h$.

An MCCP of Type II is denoted by

$$\mathcal{P}_{\mathrm{II}}(f_1, f_2, f_3, q, r, s, a_0).$$

For the last case (called Type III), we make the same assumptions as in Type II except that we replace S.3′ by

**S.3″** $\alpha_h = \alpha, \forall h \in \mathcal{H}$, where $\alpha \in [1/e, 1)$ is independent of $h$.

A Type III problem is denoted by

$$\mathcal{P}_{\alpha}(f_2, f_3, r, s, a_0),$$

where we have used the subscript $\alpha$ (instead of III) to specify that it is a Type III problem.

A Type III problem can be viewed as a special case of a Type I problem, by letting $q = 0$ and $f_1 = -\ln \alpha$. However, we consider it separately because it is the most relevant to the discrete-time stochastic control problem in the first part of this report and, in practice, is also the most basic and most often encountered.

## The Objective

For each of the three types of problems, we are interested in finding a minimal cost algorithm for computing an $\epsilon$-approximation of $x^*$ starting from some initial point $x^*_{h_0}$, $h_0 \in \mathcal{H}$. More precisely, we are interested in finding some algorithm $\gamma \in G(x^*_{h_0}, x^*)$ with the minimum complexity $\mathsf{C}^{\gamma}(\epsilon)$ and, thereby, obtaining a good upper bound on the complexity of the problem $\mathsf{C}^*(x^*_{h_0}, x^*; \epsilon)$.

We are mainly interested in the $\epsilon$ dependence of these cost functions $[\mathsf{C}^{\gamma}(\epsilon)$ for $\gamma \in G(x^*_{h_0}, x^*)$ and $\mathsf{C}^*((x^*_{h_0}, x^*; \epsilon)]$ in the limit as $\epsilon \searrow 0$. It is clear that this $\epsilon$

dependence is independent of the initial point $x^*_{h_0}$. Therefore, we will, without loss of generality, use $x^*_1$ as the initial point, and write $\mathsf{C}^*(\epsilon)$ instead of $\mathsf{C}^*(x^*_1, x^*; \epsilon)$.

Furthermore, we are also interested in the dependence on $\alpha$, $f_1$, $f_2$, and $f_3$. To simplify the complexity analysis, we introduce the following "order of magnitude" notation.

#### Order of Magnitude Notation

Let $X_1, X_2, \dots, X_n$ be some subsets of $\mathbf{R}^0$. Let $f, g : \mathbf{R}^0 \times X_1 \times X_2 \times \dots \times X_n \mapsto \mathbf{R}^0$. We write $f(\epsilon \mid x_1, x_2, \dots, x_n) = \mathrm{O}\left(g(\epsilon \mid x_1, x_2, \dots, x_n)\right)$ [or simply $f = \mathrm{O}(g)$, when it is clear what the variables are] iff there exists some constants $c, \epsilon_0 > 0$ such that

$$f(\epsilon \mid x_1, \dots, x_n) \leq c g(\epsilon \mid x_1, \dots, x_n), \quad \forall x_1 \in X_1, \dots, x_n \in X_n, \epsilon \in (0, \epsilon_0].$$

We write $f = \Omega(g)$ iff $g = \mathrm{O}(f)$; we write $f = \Theta(g)$ iff $f = \Omega(g)$ and $g = \mathrm{O}(f)$.

#### An Observation

Comparing problems of Type I and Type II, we observe that for any given $f_1$ and $q$, $(1 - f_1 h^q) \leq \mathrm{e}^{-f_1 h^q}$ for all $h \in \mathcal{H}$. Therefore, it follows (see Section 5.1.4) that for the same set of parameters a problem of Type I is "harder" than that of Type II; that is

$$\mathcal{P}_{\mathrm{II}}(f_1, f_2, f_3, q, r, s, a_0) \subset \mathcal{P}_{\mathrm{I}}(f_1, f_2, f_3, q, r, s, a_0),$$

$$G_{\mathrm{I}}(x^*_1, x^*) \subset G_{\mathrm{II}}(x^*_1, x^*), \quad \text{and} \quad \mathsf{C}^*_{\mathrm{I}}(\epsilon) \geq \mathsf{C}^*_{\mathrm{II}}(\epsilon).$$

So an algorithm for a Type I problem is also an algorithm for the corresponding Type II problem. Furthermore, since Type III problems are special cases of Type I problems, we only need to develop algorithms for Type I problems.

In the next two subsections, we will look at some approximation algorithms for problems of Type I; we consider a single-grid algorithm and a multigrid algorithm, and analyze the complexities of these algorithms. We show that the multigrid algorithm has a better complexity than single-grid. Furthermore, in Section 5.2.4, we show that the multigrid algorithm is in a certain sense optimal.

### 5.2.2 The Single-Grid Algorithm

We now consider a single-grid algorithm and analyze its complexity for a Type I problem. This algorithm is called the *single-grid algorithm* and is the traditional algorithm of choice. The algorithm is based on the following observation.

#### An Observation

Let $\gamma \in G^*(x^*_1, x^*)$. Given an $\epsilon > 0$, suppose that we use $\gamma$ to compute an $\epsilon$-approximation of $x^*$. Let the final grid-level $\gamma(\epsilon) = h_\ell$. We observe that there are two sources of errors:

1. The successive approximation error $[\mathsf{d}(x^*_{h_\ell}, A^{\gamma(\epsilon)}_{\mathcal{H}} x^*_1)]$ which we can bound by the successive approximation error bound $[\mathsf{p}(x^*_{h_\ell}, A^{\gamma(\epsilon)}_{\mathcal{H}} x^*_1)]$.
2. The discretization error $[\mathsf{d}(x^*, x^*_{h_\ell})]$ which we can bound by the discretization error bound $[\mathsf{D}(h_\ell) = \mathsf{B}(0, h_\ell)]$.

## The Main Idea

The main idea of the single-grid algorithm is as follows. First choose $h_\ell$ so that the discretization error bound is at most $\epsilon/2$. Second, apply $A^t_{h_\ell}$ on $x^*_1$, for some $t \in \mathbf{Z}^+$, until the successive approximation error bound is also at most $\epsilon/2$. Thus, by the triangle inequality the distance between $x^*$ and the final estimate $A^t_{h_\ell} x^*_1$ is given by

$$\begin{aligned}\mathsf{d}(x^*, A^t_{h_\ell} x^*_1) &\le \mathsf{p}(x^*, A^t_{h_\ell} x^*_1)\\ &= \mathsf{p}(x^*, x^*_{h_\ell}) + \mathsf{p}(x^*_{h_\ell}, A^t_{h_\ell} x^*_1)\\ &\le \epsilon/2 + \epsilon/2 = \epsilon.\end{aligned}$$

We now show that this algorithm "works": First, by Assumptions S.2 and S.5, we can always pick an $h_\ell > 0$ that satisfies the requirement $\mathsf{D}(h_\ell) \le \epsilon/2$. It follows from Assumption S.3 that $\alpha_{h_\ell} < 1$; therefore, the algorithm must terminate for some finite $t$.

## The Algorithm

We now give a more precise description of the algorithm. Given any $\epsilon > 0$, the algorithm picks the *largest* $h_\ell \in \mathcal{H}$ for which $\mathsf{D}(h_\ell) \le \epsilon/2$. Therefore, by Assumption S.5,

$$f_3 h^s_\ell \le \epsilon/2, \tag{5.2.1}$$

and

$$f_3 (2h_\ell)^s > \epsilon/2. \tag{5.2.2}$$

It follows from Eq. (5.2.2) that

$$\frac{1}{h_\ell} \le 2\left[\frac{2f_3}{\epsilon}\right]^{1/s}. \tag{5.2.3}$$

The algorithm applies $A_{h_\ell}$ on $x^*_1$ for $t$ iterations. From Assumption S.5, we have $\mathsf{D}(h_\ell) \le \mathsf{D}(1) = f_3$. Using Assumption S.4, we have

$$\begin{aligned}\mathsf{p}(x^*_{h_\ell}, A^t_{h_\ell} x^*_1) &= \alpha^t_{h_\ell} a_0 \mathsf{p}(x^*_{h_\ell}, x^*_1)\\ &\le \alpha^t_{h_\ell} a_0 \left[\mathsf{D}(h_\ell) + \mathsf{D}(1)\right]\\ &\le \alpha^t_{h_\ell} \left[2a_0 f_3\right].\end{aligned}$$

We want to choose $t \in \mathbf{Z}^+$ so that $\mathsf{p}(x^*_{h_\ell}, A^t_{h_\ell} x^*_1) \le \epsilon/2$; hence, it suffices if $\alpha^t_{h_\ell}[2a_0 f_3] \le \epsilon/2$. Using Assumption S.3, we have

$$\begin{aligned} t &\le \frac{\ln[4a_0 f_3/\epsilon]}{\left|\ln \alpha^t_{h_\ell}\right|} + 1 \\ &= \frac{\ln[4a_0 f_3/\epsilon]}{f_1 h^q_\ell} + 1. \end{aligned} \tag{5.2.4}$$

**Complexity Analysis**

We now proceed to analyze the complexity of the single-grid algorithm. We will use the symbol SG to denote the algorithm.

From Assumption S.6, the iteration cost of $A_{h_\ell}$ is

$$\mathsf{C}(h_\ell) = f_2 \left[\frac{1}{h_\ell}\right]^r. \tag{5.2.5}$$

Using Eqs. (5.2.4)–(5.2.5) and Eq. (5.2.3), we obtain the total computational cost

$$\begin{aligned} \mathsf{C}^{\mathrm{SG}}_{\mathrm{I}}(\epsilon \mid f_1, f_2, f_3) &= t\mathsf{C}(h_\ell) \\ &\le \left(\frac{\ln[4a_0 f_3/\epsilon]}{f_1 h^q_\ell} + 1\right) f_2 \left[\frac{1}{h_\ell}\right]^r \\ &\le f_2 \left(\frac{\ln[4a_0 f_3/\epsilon]}{f_1} \left[\frac{2f_3}{\epsilon}\right]^{q/s} 2^q + 1\right) \left[\frac{2f_3}{\epsilon}\right]^{r/s} 2^r \\ &= \mathsf{O}\left(\frac{f_2 \ln[f_3/\epsilon]}{f_1} \left[\frac{f_3}{\epsilon}\right]^{(q+r)/s}\right). \end{aligned}$$

As we have discussed earlier, the above upper bound applies to $\mathsf{C}^{\mathrm{SG}}_{\mathrm{II}}(\epsilon \mid f_1, f_2, f_3)$ as well. And to specialize the results for problems of Type III, we let $q = 0$ and $f_1 = |\ln \alpha|$. Therefore, we have proved the following Theorem.

**Theorem 5.2.1** *There holds*

$$\mathsf{C}^{\mathrm{SG}}_{\mathrm{I}}(\epsilon \mid f_1, f_2, f_3) = \mathsf{O}\left(\frac{f_2 \ln[f_3/\epsilon]}{f_1} \left[\frac{f_3}{\epsilon}\right]^{(q+r)/s}\right);$$

$$\mathsf{C}^{\mathrm{SG}}_{\mathrm{II}}(\epsilon \mid f_1, f_2, f_3) = \mathsf{O}\left(\frac{f_2 \ln[f_3/\epsilon]}{f_1} \left[\frac{f_3}{\epsilon}\right]^{(q+r)/s}\right);$$

$$\mathsf{C}^{\mathrm{SG}}_{\alpha}(\epsilon \mid f_2, f_3) = \mathsf{O}\left(\frac{f_2 \ln[f_3/\epsilon]}{|\ln \alpha|} \left[\frac{f_3}{\epsilon}\right]^{r/s}\right).$$

### 5.2.3 The Basic Multigrid Algorithm

We now look at a one-way multigrid algorithm and analyze its complexity. This algorithm is called the *basic multigrid algorithm.* This algorithm has better complexity than the single-grid version and is based on the following observation.

## An Observation

As the grid-level $h \searrow 0$ both the contraction factor and the iteration cost become worse. But since the initial error is large, we can save computational cost by starting the initial iterations on a coarse grid and use finer and finer grids as the estimate becomes more accurate. Based on this observation we have the following *multigrid principle.*

> **Multigrid Principle:** To save computations, start at a coarse grid-level and iterate on that grid-level until the successive approximation error (bound) is comparable to the discretization error (bound) before changing to a finer grid-level.

We note that once the discretization and approximation errors are comparable, there is no reason to iterate any more on that grid-level, since the grid-level cannot provide more information for approximating $x^*$.

## The Algorithm

The basic multigrid algorithm uses the following implementation of the multigrid principle. The algorithm chooses the finest grid-level $h_\ell \in \mathcal{H}, \ell \in \mathbf{Z}^+$ (as in SG) so that $h_\ell$ is the coarsest grid-level that satisfies $f_3 h_\ell^s \leq \epsilon/2$. Therefore, $h_\ell$ satisfies Eq. (5.2.3). The algorithm uses the following grid-levels

$$h_i = 1/2^i, \quad i = 1, 2, \ldots, \ell.$$

The algorithm starts at the initial grid-level $h_1$, and does $t_1 \in \mathbf{Z}^+$ iterations of $A_{h_1}$ until $\mathsf{p}(x^*_{h_1}, A^{t_1}_{h_1} x^*_1) \leq \mathsf{D}(h_1)$. [Namely, the successive approximation error bound is not more than the discretization error bound]. This is the *grid-refinement criterion.* Once this criterion is met, the algorithm changes to grid-size $h_2$ using $A^{t_1}_{h_1} x^*_1$ as the initial estimate on the grid-level. The algorithm applies $A_{h_2}$ on the estimate for $t_2$ iterations until the grid-refinement criterion for grid-level $h_2$ is met; that is, $\mathsf{p}(x^*_{h_2}, A^{t_2}_{h_2} A^{t_1}_{h_1} x^*_1) \leq \mathsf{D}(h_2)$

More generally, let $\eta \in \mathcal{A}$ denote the trajectory associated with the basic multigrid algorithm when the grid-refinement criterion on grid-level $h_{i-1}$ is met. Therefore, the final grid-level of $\eta$ is $h_{i-1}$ and

$$\mathsf{p}(x^*_{h_{i-1}}, A^{\eta}_{\mathcal{H}} x^*_1) \leq \mathsf{D}(h_{i-1}). \tag{5.2.6}$$

The algorithm now proceeds to grid-level $h_i$ and applies $t_i$ iterations of $A_{h_i}$ to $A^{\eta}_{\mathcal{H}} x^*_1$ until the grid-refinement criterion on grid-level $h_i$ is met; that is,

$$\mathsf{p}(x^*_{h_i}, A^{t_i}_{h_i} A^{\eta}_{\mathcal{H}} x^*_1) \leq \mathsf{D}(h_i). \tag{5.2.7}$$

Eventually the algorithm reaches grid-level $h_\ell$ and does $t_\ell$ iterations of $A_{h_\ell}$ until the grid-refinement criterion is met. The algorithm terminates and returns the final estimate $A^{\eta'}_{\mathcal{H}} x^*_1$, where

$$\eta' = ((t_\ell, h_\ell), \ldots, (t_2, h_2), (t_1, h_1))$$

denotes the trajectory of the algorithm.

Therefore, using the definition of $\mathsf{p}$, the grid-refinement criterion [cf. Eq. (5.2.7)], and Eq. (5.2.1) we obtain

$$\begin{aligned}\mathsf{d}(x^*, A^{\eta'}_{\mathcal{H}} x^*_1) &\le \mathsf{p}(x^*, A^{\eta'}_{\mathcal{H}} x^*_1)\\ &= \mathsf{p}(x^*, x^*_{h_\ell}) + \mathsf{p}(x^*_{h_\ell}, A^{\eta'}_{\mathcal{H}} x^*_1)\\ &= \mathsf{D}(h_\ell) + \mathsf{p}(x^*_{h_\ell}, A^{\eta'}_{\mathcal{H}} x^*_1)\\ &\le 2\mathsf{D}(h_\ell) \le \epsilon.\end{aligned}$$

Thus, $A^{\eta'}_{\mathcal{H}} x^*_1$ is an $\epsilon$-approximation of $x^*$ as required.

The convergence of the basic multigrid algorithms follows simply from the fact that the grid-level is always halved each time the algorithm moves to the next grid-level; after $\ell - 1$ grid-refinements the algorithm reaches grid-level $h_\ell$. Moreover, at each grid-level, the grid-refinement criterion is met after a finite number of iteration because $\alpha_h < 1, \forall h \in \mathcal{H}$ (by Assumption S.3).

We now upper bound the number of iterations on each grid-level. Suppose that the algorithm has just moved into grid-level $h_i$ and let $\eta$ be its trajectory so far. The algorithm does $t_i$ iteration on this grid-level until Eq. (5.2.7) is satisfied. Using the definition of $\mathsf{p}$, Eq. (5.2.6), Assumption S.3 we obtain

$$\begin{aligned}\mathsf{p}(x^*_{h_i}, A^{t_i}_{h_i} A^{\eta}_{\mathcal{H}} x^*_1) &= \alpha^{t_i}_{h_i} a_0 \left[\mathsf{B}(h_i, h_{i-1}) + \mathsf{p}(x^*_{h_{i-1}}, A^{\eta}_{\mathcal{H}} x^*_1)\right]\\ &\le \alpha^{t_i}_{h_i} a_0 \left[\mathsf{D}(h_i) + 2\mathsf{D}(h_{i-1})\right]\\ &= \alpha^{t_i}_{h_i} \left[a_0(1 + 2^{s+1})\right] \mathsf{D}(h_i).\end{aligned}$$

[We let $h_0 = 1$, so the above bound holds for $i = 1$.]

To satisfy the grid-refinement criterion on grid-level $h_i$ [Eq. (5.2.7)], it suffices to have $\alpha^{t_i}_{h_i} [a_0(1 + 2^{s+1})] \le 1$. Thus,

$$\begin{aligned}t_i &\le \frac{\ln\left[a_0(1+2^{s+1})\right]}{-\ln \alpha_{h_i}} + 1\\ &= \frac{\ln\left[a_0(1+2^{s+1})\right]}{f_1 h_i^q} + 1, \quad \forall i = 1, 2, \ldots, \ell.\end{aligned} \tag{5.2.8}$$

## Complexity Analysis

We now analyze the complexity of the basic multigrid algorithm. We will use the symbol MG to denote the algorithm.

First, by our choice of grid-levels, we have

$$\begin{aligned}\sum_{i=1}^{\ell} \left[\frac{1}{h_i}\right]^x &= \sum_{i=1}^{\ell} \left[\frac{1}{2^{i-1} h_\ell}\right]^x\\ &= \left[\frac{1}{h_\ell}\right]^x \sum_{i=0}^{\ell-1} 2^{-xi}\end{aligned}$$

$$\leq \left[\frac{1}{h_\ell}\right]^x \frac{1}{1-2^{-x}}, \quad \forall x > 0. \tag{5.2.9}$$

Using Eqs. (5.2.8), (5.2.9), and (5.2.3), we obtain the following bound on the total computational cost of using MG

$$\begin{aligned}
\mathsf{C}_{\mathrm{I}}^{\mathrm{MG}}(\epsilon \mid f_1, f_2, f_3) &= \sum_{i=1}^{\ell} t_i \mathsf{C}(h_i) \\
&= \sum_{i=1}^{\ell} \left[\frac{\ln\left[a_0(1+2^{s+1})\right]}{f_1 h_i^q} + 1\right] f_2 \left[\frac{1}{h_i}\right]^r \\
&= f_2 \left( \frac{\ln\left[a_0(1+2^{s+1})\right]}{f_1} \sum_{i=1}^{\ell} \left[\frac{1}{h_i}\right]^{(q+r)} + \sum_{i=1}^{\ell} \left[\frac{1}{h_i}\right]^r \right) \\
&\leq f_2 \left( \frac{\ln\left[a_0(1+2^{s+1})\right]}{f_1} \left[\frac{1}{h_\ell}\right]^{(q+r)} \frac{1}{1-2^{-(q+r)}} + \left[\frac{1}{h_\ell}\right]^r \frac{1}{1-2^{-r}} \right) \\
&\leq f_2 \left( \frac{\ln\left[a_0(1+2^{s+1})\right]}{f_1} \left[\frac{2f_3}{\epsilon}\right]^{(q+r)/s} \frac{2^{(q+r)}}{1-2^{-(q+r)}} + \left[\frac{2f_3}{\epsilon}\right]^{r/s} \frac{2^r}{1-2^{-r}} \right) \\
&= \mathrm{O}\left( \frac{f_2}{f_1} \left[\frac{f_3}{\epsilon}\right]^{(q+r)/s} \right).
\end{aligned}$$

Hence, we have obtained an upper bound on $\mathsf{C}_{\mathrm{I}}^{\mathrm{MG}}(\epsilon \mid f_1, f_2, f_3)$ and, consequently, an upper bound on $\mathsf{C}_{\mathrm{I}}^*(\epsilon \mid f_1, f_2, f_3)$. By the same argument given earlier for SG, the bound also applies to problems of Types II and III. The results are summarized below.

**Theorem 5.2.2** *There holds*

$$\mathsf{C}_{\mathrm{I}}^*(\epsilon \mid f_1, f_2, f_3) = \mathrm{O}\left( \frac{f_2}{f_1} \left[\frac{f_3}{\epsilon}\right]^{(q+r)/s} \right);$$

$$\mathsf{C}_{\mathrm{II}}^*(\epsilon \mid f_1, f_2, f_3) = \mathrm{O}\left( \frac{f_2}{f_1} \left[\frac{f_3}{\epsilon}\right]^{(q+r)/s} \right);$$

$$\mathsf{C}_{\alpha}^*(\epsilon \mid f_2, f_3) = \mathrm{O}\left( \frac{f_2}{|\ln \alpha|} \left[\frac{f_3}{\epsilon}\right]^{r/s} \right).$$

*Furthermore, the above complexity bounds are attained by* MG, *the basic multigrid-algorithm.*

[Note that in MG we only need $a_{h/2}^h \leq a_0$ for all $h \in \mathcal{H}$ instead of $a_{h'}^h \leq a_0$ for all $h, h' \in \mathcal{H}$.]

## Discussion

In our analysis, we have assumed that $\mathcal{H} = \{1, 1/2, 1/4, \ldots\}$. In general, we could have assumed $\mathcal{H} = \{1/z^i \mid i \in \mathbf{Z}^0\}$, for any $z > 1$. The algorithms, SG and MG are easily adapted to this choice of grid-levels; moreover, the complexities of the adapted algorithms (Theorems 5.2.1 and 5.2.2) remain unchanged. (Note that using a "larger" $\mathcal{H}$ only makes the problem "easier".) In practice, there is a preference for choosing $z$ an integer (more generally, $h_i/h_{i-1}$ an integer). The is because it is often preferable to use a "uniform" grid; for example, in the finite difference discretization scheme, a uniform grid is usually assumed. [By choosing $z$ (depending on $q$, $r$, and $s$) appropriately, it is possible to obtain a multigrid algorithm (for continuous-time MCCP's) that is also optimal in the dependence on $q$, $r$, and $s$.]

Comparing the results of Theorems 5.2.1 and 5.2.2, we note that MG improves on the complexity of SG by a factor of $\ln[f_3/\epsilon]$. The natural question is this: Is there is an algorithm with a better complexity than MG? The answer is no. (At least not with respect to the dependence on $\alpha, f_1, f_2, f_3, \epsilon$.) We will show this in the next subsection.

### 5.2.4 Lower Bounds

In this subsection we show, for the problems of Types I–III, that the basic multigrid algorithm, MG, is optimal in its dependence on $\alpha, f_1, f_2, f_3, \epsilon$; no algorithm can improve on the complexity of MG. We first show the result for problems of Type I and then apply the result to problems of Types II and III. For the rest of this subsection we will focus on showing the result for a Type I problem.

An outline of this subsection is as follows. We first introduce some simplifying assumptions and present the main ideas of the proof. We then construct an instance whose minimum computational cost function $\mathsf{C}_{\mathrm{I}}^*$ is a solution to a certain optimal control problem. Next, we solve the optimal control problem and obtain a lower bound on $\mathsf{C}_{\mathrm{I}}^*$. Finally we apply the results to problems of Types II and III.

#### Simplifying Assumptions

Since we prove lower bound on complexity, the result is "strongest" if we prove this lower bound for an "easiest" problem. For this reason we replace Assumptions S.1, S.2, S.4, and S.5 by the following assumptions.

**S.1′** $\mathcal{T} = \mathbf{R}^0$.

**S.2′** $\mathcal{H} = (0, 1]$.

**S.4′** $a_h^{h'} = 1$ for all $h', h \in \mathcal{H}$.

**S.5′** $\mathsf{B}(h, h') = |f_3 h^s - f_3 h'^s|, \forall h, h' \in \mathcal{H}_0$.

It is clear that letting $\mathcal{T} = \mathbf{R}^0$, $\mathcal{H} = (0, 1]$, and $a_h^{h'} = 1$ only makes the problem easier. And it is clear that the grid-transfer error bound in Assumption S.5′ (is a metric on $\mathcal{H}_0$ and) is smaller than the bound in Assumption S.5; therefore, this also makes the problem easier. For the rest of this subsection, Assumptions S.1′, S.2′, S.4′, and S.5′ are in effect; furthermore, let us fix a Type I problem $\mathcal{P}_{\mathrm{I}}(f_1, f_2, f_3, q, r, s)$ for our discussion.

## Main Ideas

We now give the main ideas of the lower bound construction. We first construct an instance of $\mathcal{P}_{\mathrm{I}}(f_1, f_2, f_3, q, r, s)$ and show that its minimum cost corresponds to the solution of a certain continuous-time optimal control problem. After relaxing certain constraints for that problem, we get a new optimal control problem. We then obtain a closed-form solution to the new optimal control problem by solving a certain Hamilton-Jacobi-Bellman (HJB) equation. As a result we obtain a lower bound on the minimum computational cost for the instance, thereby, obtain a lower bound on the problem's complexity $\mathsf{C}_{\mathrm{I}}^*$. Finally, by comparing the lower bound with the upper bound of Theorem 5.1.2, we prove the optimality of MG.

## The Instance

We construct the instance by first constructing the metric space. Let $\mathsf{M} = [0, f_3]$ and the metric $\mathsf{d}(x, y) = |x - y|, \forall x, y \in \mathsf{M}$.

We now construct the family of contraction processes. Let $x_h^* = f_3 h^s, \forall h \in [0, 1]$, and let

$$A_h^t x = \alpha_h^t (x - f_3 h^s) + f_3 h^s, \quad t \geq 0, h \in (0, 1], x \in \mathsf{M}, \tag{5.2.10}$$

where $\alpha_h = \mathrm{e}^{-f_1 h^q}$. It follows that

$$\mathsf{d}(A_h^t x, A_h^t y) = \alpha_h^t \mathsf{d}(x, y), \quad \forall t \geq 0, h \in \mathcal{H}, x, y \in \mathsf{M}.$$

Hence, $A_h$ is a (continuous-time) semi-group of contraction mappings on $\mathsf{M}$ with contraction factor $\alpha_h$ and fixed-point $x_h^*$; therefore, $\{A_h\}$ is the required family of contraction processes.

The minimum cost function for computing an $\epsilon$-approximation to $x^* = 0$, starting from the initial point $x_1^* = f_3$, is given by

$$\mathsf{C}_{\mathrm{I}}^*(\epsilon) = \inf_{\eta \in \mathcal{A}_0} \mathsf{C}_{\mathcal{H}}^{\eta} \tag{5.2.11}$$

subject to $\mathsf{d}(x^*, A_{\mathcal{H}}^{\eta} x_1^*) = A_{\mathcal{H}}^{\eta} f_3 \leq \epsilon$.

## A Continuous-Time Optimal Control Problem

We can view Eq. (5.2.10) as a description of the law of motion of a certain dynamical system. Let $x(t) \in \mathsf{M}$ denote the "state" of the system at time $t$. Fixing the grid-level $h$, we can rewrite Eq. (5.2.10) as a differential equation as follows:

$$\begin{aligned}
\dot{x}(t) &= \lim_{\Delta t \searrow 0} \left( A_h^{\Delta t} x(t) - x(t) \right) / \Delta t \\
&= \lim_{\Delta t \searrow 0} \frac{\left( \alpha_h^{\Delta t} - \alpha_h^0 \right)}{\Delta t} \cdot [x(t) - f_3 h^s] \\
&= \ln \alpha_h \cdot [x(t) - f_3 h^s] \\
&= -f_1 h^q x(t) + f_1 f_3 h^{q+s}.
\end{aligned} \tag{5.2.12}$$

[Note that $\lim_{\Delta t \searrow 0}(A_h^{\Delta t} - I)/\Delta t$ is the infinitesimal generator of $A_h$.] It is clear that Eq (5.2.10) is the solution to the dynamical system described by Eq. (5.2.12).

Let $G \overset{\text{def}}{=} \{u : [0,\infty) \mapsto \mathcal{H} \mid \text{such that } u \text{ is a Borel measurable function}\}$. By letting $h = u(t)$ be a control variable, we obtain the following control equation

$$\dot{x}(t) = -f_1 u^q(t) x(t) + f_1 f_3 u^{q+s}(t), \quad u \in G. \tag{5.2.13}$$

Eq. (5.2.13) can be viewed as the state equation of a dynamical system controlled by $u$.

For any control function $u \in G$, let $x^u(t), t \in [0,\infty)$ denote the *state trajectory* when the system [Eq. (5.2.13)] is operated with $u$. We now formulate an optimal control problem. Given the initial state of the system $x(0) = f_3 h_0^s$ and a final state $f_3 h_\ell^s$, the problem is to choose some control function $u \in G$ so that at some time $t \in [0,\infty)$, $x^u(t) = f_3 h_\ell^s$. The time when the state *first* reaches $f_3 h_\ell^s$ is called the *terminal time* of $u$ and is denoted by $T$. More formally,

$$T \overset{\text{def}}{=} \inf_{t \in [0,\infty)} \{t \mid x^u(t) = f_3 h_\ell^s\}.$$

[The terminal time of $u$ is infinite if the final state is never reached.] The objective is to minimize the cost needed (as a result of using $u$) to drive the system from $f_3 h_0^s$ to $f_3 h_\ell^s$. More precisely, we are interested in

$$\mathsf{C}^{**}(f_3 h_0^s, f_3 h_\ell^s) \overset{\text{def}}{=} \inf_{\substack{T \geq 0 \\ u \in G}} \left\{ \int_0^T f_2 \left[\frac{1}{u(\tau)}\right]^r d\tau \;\middle|\; x^u(0) = f_3 h_0^r, x^u(T) = f_3 h_\ell^r \right\},$$

where $T$ is the terminal time of $u$. For later reference, the above optimal control problem is denoted by *OCP*.

We now return to the original minimum cost problem [Eq. (5.2.11)]. For any control trajectory $\eta \in \mathcal{A}_0$, there exists a control function $u \in G$ and some terminal time $T$ such that $x^u(0) = f_3$, $x^u(T) = A_{\mathcal{H}}^{\eta} f_3$, and

$$\mathsf{C}_{\mathcal{H}}^{\eta} = \int_0^T f_2 \left[\frac{1}{u(\tau)}\right]^r d\tau.$$

[Namely, for any $\eta$ we choose $u$ to be a piecewise constant function with grid-levels and durations matching $\eta$.] Therefore

$$\mathsf{C}_{\mathrm{I}}^{*}(f_3, 0; \epsilon) \geq \mathsf{C}^{**}(f_3, \epsilon), \quad \forall \epsilon > 0.$$

The main reason for considering *OCP*, is that we can obtain a closed form solution to $\mathsf{C}^{**}$, by solving the following Hamilton-Jacobi-Bellman (HJB) equation.

$$\begin{aligned} 0 &= \inf_{u \in \mathcal{H}} \left\{ f_2 \left[\frac{1}{u}\right]^r + \frac{\partial V(t,y)}{\partial t} + \frac{\partial V(t,y)}{\partial y} \cdot \dot{x}(t) \right\} \\ &= \inf_{u \in (0,1]} \left\{ f_2 \left[\frac{1}{u}\right]^r + \frac{\partial V(t,y)}{\partial t} + \frac{\partial V(t,y)}{\partial y} \left[-f_1 u^q \cdot y + f_1 f_3 u^{q+r}\right] \right\}, \end{aligned} \tag{5.2.14}$$

where $V(t,y)$ is the *cost-to-go function* when the system, at time $t$, is in state $y$. [See, for example, Bryson and Ho (1961) and Flemming (1975) for a discussion of such problems.] The boundary condition for the HJB equation is $V(t, f_3 h_\ell^s) = 0, \forall t \geq 0$; that is, the cost-to-go is 0 once the system reaches the final state $f_3 h_\ell^s$.

Instead of solving the HJB equation directly, we will use a heuristic derivation to obtain a closed form solution to C**. We then verify that it is indeed the solution by showing that it satisfies the HJB equation.

## A Heuristic Derivation

We now give a non-rigorous derivation of the solution to *OCP*. In the next subsection we will verify that we have indeed obtained the solution.

To simplify the algebra, we make a change of variable by letting $y = f_3 h^s$; in particular, $y_0 = f_3 h_0^s$, $y_\ell = f_3 h_\ell^s$. The *OCP* can be rewritten as

$$\mathsf{C}^{**}(y_0, y_\ell) = \inf_{\substack{T\geq 0\\ u\in G}} \left\{ \int_0^T f_2 \left[\frac{1}{u(\tau)}\right]^r d\tau \;\middle|\; x^u(0) = y_0, x^u(T) = y_\ell \right\}.$$

Let $\bar{u} \in G$ denote the optimal control function, which we assume to exist. Let $\bar{x}$ denote the corresponding optimal state trajectory. [That is, $\bar{x}(\cdot) = x^{\bar{u}}(\cdot)$.] Suppose that at some time $t \in [0,T]$ the system is at grid-level $h$ and that the optimal state and control are $\bar{x}(t) = f_3 h^s$ and $\bar{u}(t) = \bar{u}_y \in \mathcal{H}$, respectively.

Consider the changes in the state and in the cost from time $t$ to $t + \Delta t$, for some $\Delta t > 0$ very small. The "distance moved"

$$\begin{aligned}\Delta y &= x(t+\Delta t) - x(t)\\ &= \dot{x}(t)\cdot \Delta t\\ &= f_1\left[-\bar{u}_y^q f_3 h^s + f_3 \bar{u}_y^{q+s}\right]\cdot \Delta t\\ &= f_1\left[-\bar{u}_y^q y + f_3 \bar{u}_y^{q+s}\right]\cdot \Delta t\end{aligned}$$

and the increase in cost

$$\begin{aligned}\Delta\mathsf{C}^{**} &= \mathsf{C}^{**}(y_0, y+\Delta y) - \mathsf{C}^{**}(y_0, y)\\ &= f_2\left[\frac{1}{\bar{u}_y}\right]^r \cdot \Delta t.\end{aligned}$$

Therefore, the "increase in cost/distance moved" is given by

$$\begin{aligned}\frac{d\mathsf{C}^{**}(y_0,y)}{dy} &= \lim_{\Delta y\searrow 0} \frac{\Delta \mathsf{C}^{**}}{\Delta y}\\ &= \frac{f_2}{f_1}\left[\frac{1}{-\bar{u}_y^{q+r} y + f_3 \bar{u}_y^{q+r+s}}\right]. \qquad (5.2.15)\end{aligned}$$

If $\bar{u}_y$ is the optimal control at time $t$, then it should minimize $d\mathsf{C}^{**}/dy$ or, equivalently, maximize the expression $[-u^{q+r}y + f_3 u^{q+r+s}]$. (This is actually the "differential form" of the *optimality principle* in dynamic programming; we will say more in Section 5.2.5.) To get the optimum control $\bar{u}_y$, we let

$$\begin{aligned} 0 &= \frac{d(-u^{q+r}y + f_3 u^{q+r+s})}{du} \\ &= -(q+r)u^{q+r-1}y + f_3(q+r+s)u^{q+r+s-1}. \end{aligned}$$

It follows that

$$\bar{u}_y = \left[\frac{\sigma y}{(\sigma+1)f_3}\right]^{1/s}, \tag{5.2.16}$$

where $\sigma \overset{\text{def}}{=} (q+r)/s$. Substituting Eq. (5.2.16) into Eq. (5.2.15) we obtain

$$\begin{aligned} \frac{d\mathsf{C}^{**}(y_0,y)}{dy} &= \frac{f_2}{f_1 f_3}\left[-\left(\frac{\sigma}{\sigma+1}\right)^{\sigma} + \left(\frac{\sigma}{\sigma+1}\right)^{\sigma+1}\right]^{-1}\left[\frac{f_3}{y}\right]^{\sigma+1} \\ &= \frac{f_2}{f_1 f_3}\left[-1 + \frac{\sigma}{\sigma+1}\right]^{-1}\left(\frac{\sigma+1}{\sigma}\right)^{\sigma}\left[\frac{f_3}{y}\right]^{\sigma+1} \\ &= \frac{f_2(\sigma+1)}{-f_1 f_3}\left(\frac{\sigma+1}{\sigma}\right)^{\sigma}\left[\frac{f_3}{y}\right]^{\sigma+1}. \end{aligned} \tag{5.2.17}$$

Integrating Eq. (5.2.17) with respect to $y$ and using the fact that $\mathsf{C}^{**}(y_0,y_0) = 0$, we obtain

$$\mathsf{C}^{**}(y_0,y) = \frac{f_2}{f_1}\left(\frac{\sigma+1}{\sigma}\right)^{\sigma+1}\left[\left(\frac{f_3}{y}\right)^{\sigma} - \left(\frac{f_3}{y_0}\right)^{\sigma}\right]. \tag{5.2.18}$$

## Optimal State and Control Trajectories

To complete the derivation, we will obtain the optimal state and control trajectories as functions of $t$. Let $\bar{u}(t) = \bar{u}_y$ and $\bar{x}(t) = y$. From Eq. (5.2.16), we have

$$\bar{u}(t) = \left(\frac{\sigma}{\sigma+1}\right)^{1/s}\left[\frac{\bar{x}(t)}{f_3}\right]^{1/s}. \tag{5.2.19}$$

Substituting Eq. (5.2.19) into Eq. (5.2.13), we obtain,

$$\begin{aligned} \dot{\bar{x}}(t) &= -f_1\left(\frac{\sigma}{\sigma+1}\right)^{q/s}\left[\frac{\bar{x}(t)}{f_3}\right]^{q/s}\bar{x}(t) + f_1 f_3\left(\frac{\sigma}{\sigma+1}\right)^{q/s+1}\left[\frac{\bar{x}(t)}{f_3}\right]^{q/s+1}. \\ &= f_1 f_3^{-q/s}\left(\frac{\sigma}{\sigma+1}\right)^{q/s}\left[-1 + \frac{\sigma}{\sigma+1}\right]\bar{x}^{q/s+1}(t) \\ &= -\frac{f_1 f_3^{-q/s}}{(\sigma+1)}\left(\frac{\sigma}{\sigma+1}\right)^{q/s}\bar{x}^{q/s+1}(t) \\ \Rightarrow \frac{d\bar{x}}{\bar{x}^{q/s+1}} &= -\frac{f_1 f_3^{-q/s}}{(\sigma+1)}\left(\frac{\sigma}{\sigma+1}\right)^{q/s}dt. \end{aligned} \tag{5.2.20}$$

We consider two cases: First, suppose that $q \neq 0$. Then integrating Eq. (5.2.20) from $t = 0$ to $t$, we obtain

$$\frac{s}{q}\left[\bar{x}^{-q/s}(t) - \bar{x}^{-q/s}(0)\right] = \frac{f_3^{-q/s} f_1}{(\sigma+1)} \left(\frac{\sigma}{\sigma+1}\right)^{q/s} t.$$

Since the initial condition is $\bar{x}(0) = f_3 h_0^s$, we have

$$\bar{x}(t) = f_3 \left[h_0 + \frac{q f_1}{s(\sigma+1)} \left(\frac{\sigma}{\sigma+1}\right)^{q/s} t\right]^{-s/q}, \quad t \in [0, T]. \tag{5.2.21}$$

Substituting Eq. (5.2.21) into Eq. (5.2.19), we obtain

$$\bar{u}(t) = \left(\frac{\sigma}{\sigma+1}\right)^{1/s} \left[h_0 + \frac{q f_1}{s(\sigma+1)} \left(\frac{\sigma}{\sigma+1}\right)^{q/s} t\right]^{-1/q}, \quad t \in [0, T] \tag{5.2.22}$$

where $T$ is the terminal time of $\bar{u}$. [It is clear that $\bar{u} \in G$.]

Second, if $q = 0$ (which corresponds to a problem of Type III), from Eq. (5.2.20) we have

$$\frac{d\bar{x}}{\bar{x}} = -\frac{f_1 s}{r+s},$$
$$\Rightarrow \bar{x}(t) = f_3 h_0^s e^{-\frac{f_1 s}{r+s} t}, \quad t \in [0, T]. \tag{5.2.23}$$

Substituting Eq. (5.2.23) into Eq. (5.2.19), we have

$$\bar{u}(t) = \left(\frac{\sigma}{\sigma+1}\right)^{1/s} h_0 e^{-\frac{f_1}{r+s} t}, \quad t \in [0, T]. \tag{5.2.24}$$

Finally, the terminal time $T$ is given by $\bar{x}(T) = f_3 h_\ell^s$.

### 5.2.5 Verification

We now show that $\mathsf{C}^{**}(y_0, y_\ell)$ is indeed the optimal cost function for *OCP*.

It suffices to show that $V(t, y) = \mathsf{C}^{**}(y, y_\ell)$ satisfies the HJB equation [Eq. (5.2.14)]. And to verify that $\bar{u}_y$ [Eq. (5.2.16)] is the optimal control, it suffices to show that the infimum in Eq. (5.2.14) is attained by $\bar{u}_y$.

First, we check that $V(t, y_\ell) = \mathsf{C}^{**}(y_\ell, y_\ell) = 0, \forall t$. Therefore, $V(t, y_\ell)$ satisfies the boundary condition. Second, since $\mathsf{C}^{**}(y, y_\ell)$ is independent of $t$, we have

$$\frac{\partial V(t, y)}{\partial t} = 0.$$

And from Eq. (5.2.18), we have

$$\frac{\partial V(t, y)}{\partial y} = \frac{d\mathsf{C}^{**}(y, y_1)}{dy} = -\frac{d\mathsf{C}^{**}(y_0, y)}{dy}.$$

Therefore, the HJB equation becomes

$$0 = \inf_{u\in(0,1]} \left\{ f_2 \left[\frac{1}{u}\right]^r - \frac{d\mathsf{C}^{**}(y_0, y)}{dy} \left[-f_1 u^q \cdot y + f_1 f_3 u^{q+r}\right] \right\} \tag{5.2.25}$$

and we are required to show that the infimum in Eq. (5.2.25) is attained by

$$\bar{u}_y = \left[\frac{\sigma y}{(\sigma+1) f_3}\right]^{1/s}.$$

The verification that $\mathsf{C}^{**}$ and $\bar{u}_y$ are the optimal cost function and the optimal control, respectively, is reduced to showing that

$$\frac{d\mathsf{C}^{**}(y_0, y)}{dy} = \frac{f_2}{f_1} \left[\frac{1}{-u^{q+r} \cdot y + f_3 u^{q+r+s}}\right]_{u=\bar{u}_y}. \tag{5.2.26}$$

Comparing with Eqs. (5.2.15)-(5.2.16), we see that Eq. (5.2.26) holds, as required. We also see that the heuristic derivation of $d\mathsf{C}^{**}(y_0, y)/dy$ actually uses Eq. (5.2.26) as the starting point of the derivation [cf. Eq. (5.2.15)], and that we choose the value of $\bar{u}_y$ to minimize the right-hand side of Eq. (5.2.26).

## 5.2.6 Summary of Results

From Eq. (5.2.18), we have

$$\mathsf{C}^{**}(f_3, \epsilon) = \frac{f_2}{f_1} \left(\frac{\sigma+1}{\sigma}\right)^{\sigma+1} \left[\left(\frac{f_3}{\epsilon}\right)^{\sigma} - 1\right].$$

Therefore,

$$\begin{aligned} \mathsf{C}^*_{\mathrm{I}}(\epsilon \mid f_1, f_2, f_3) &\geq \mathsf{C}^{**}(f_3, \epsilon) \\ &= \frac{f_2}{f_1} \left(\frac{q+r+s}{q+r}\right)^{(q+r+s)/s} \left[\left(\frac{f_3}{\epsilon}\right)^{(q+r)/s} - 1\right]. \\ &= \Omega\left(\frac{f_2}{f_1} \left[\frac{f_3}{\epsilon}\right]^{(q+r)/s}\right) \end{aligned}$$

Specializing our results to problems of Type III (by letting $q = 0$ and $f_1 = -\ln \alpha$) we obtain

$$\mathsf{C}^*_{\alpha}(\epsilon \mid f_2, f_3) = \Omega\left(\frac{f_2}{|\ln \alpha|} \left[\frac{f_3}{\epsilon}\right]^{r/s}\right).$$

To obtain the results for problems of Type II, we note that there exists some $h_1 \in (0,1)$ such that

$$e^{-2f_1 h^q} \leq 1 - f_1 h^q, \quad \forall h \in [0, h_1].$$

Therefore, if we consider a problem of Type I $\mathcal{P}_{\rm I}(2f_1, f_2, f_3, q, r, s)$ and a problem of Type II $\mathcal{P}_{\rm II}(f_1, f_2, f_3, q, r, s)$, we have

$$\begin{aligned} \mathrm{C}^*_{\rm II}(x^*_{h_1}, x^*; \epsilon \mid f_1, f_2, f_3) &\geq \mathrm{C}^*_{\rm I}(x^*_{h_1}, x^*; \epsilon \mid f_1, f_2, f_3) \\ &\geq \mathrm{C}^{**}(f_3 h_1^s, \epsilon) \\ &= \frac{f_2}{2f_1}\left(\frac{q+r+s}{q+r}\right)^{(q+r+s)/s}\left[\left(\frac{f_3}{\epsilon}\right)^{(q+r)/s} - \left(\frac{1}{h_1}\right)^{(q+r)}\right]. \\ &= \Omega\left(\frac{f_2}{f_1}\left[\frac{f_3}{\epsilon}\right]^{(q+r)/s}\right). \end{aligned}$$

Since the asymptotic dependence on $\epsilon$ is independent of the initial point, we have

$$\mathrm{C}^*_{\rm II}(\epsilon \mid f_1, f_2, f_3) = \Omega\left(\frac{f_2}{f_1}\left[\frac{f_3}{\epsilon}\right]^{(q+r)/s}\right).$$

We summarize the results of this subsection by the following theorem:

**Theorem 5.2.3** *There holds*

$$\mathrm{C}^*_{\rm I}(\epsilon \mid f_1, f_2, f_3) = \Omega\left(\frac{f_2}{f_1}\left[\frac{f_3}{\epsilon}\right]^{(q+r)/s}\right);$$

$$\mathrm{C}^*_{\rm I}(\epsilon \mid f_1, f_2, f_3) = \Omega\left(\frac{f_2}{f_1}\left[\frac{f_3}{\epsilon}\right]^{(q+r)/s}\right);$$

$$\mathrm{C}^*_{\alpha}(\epsilon \mid f_2, f_3) = \Omega\left(\frac{f_2}{|\ln \alpha|}\left[\frac{f_3}{\epsilon}\right]^{r/s}\right).$$

*It follows that* MG, *the basic multigrid algorithm, attains the above complexity bounds; hence, it is optimal with respect to the dependence on $\alpha, f_1, f_2, f_3, \epsilon$.*

Note that for those cases where the metric space is specified by the problem, we can still use the same lower bound construction, provided we can isometrically imbed the interval $[0, f_3]$ into the specified metric space. It is always possible to do the imbedding for any normed vector space (of non-zero dimension).

### 5.2.7 Discussion

In this subsection we discuss some implications of the results. Our heuristic derivation of the minimum cost function of $\mathrm{C}^{**}$ is based on the optimality principle in dynamic programming. [See Bertsekas (1987) for more details.] The optimality principle is best illustrated by the following example: Let $u \in G$ be the optimal control function that drives the system from the initial state $x_0$ to the final state $x_T$. Let $x^u(\cdot)$ be state trajectory corresponding to $u$. Therefore $x^u(0) = x_0$ and $x^u(T) = x_T$, where $T$

is the terminal time of $u$. Then the optimality principle says that for all $t \in (0,T)$, $u$ must also be the optimal control function that drives the system from $x_0$ to the intermediate state $x_t = x^u(t)$.[†]

The reason is that if there is a better control function $v \in G$ that drives the system from state $x_0$ to $x_t$, then one can improve on $u$ by first using $v(t), t \in [0,T']$, where $T'$ is the terminal time of $v$, to drive the system to $x_t$; then use $u(t), t \in [t,T]$ to drive the system from $x_t$ to $x_T$. This contradicts the optimality of $u$. (This argument is possible because the iteration cost $f_2[1/u]^r$ and the cost-to-go function are time independent.)

It is clear from the optimality principle that the optimal state and control trajectories, $\bar{x}(t)$ and $\bar{u}(t)$ [Eqs. (5.2.21)–(5.2.24)], are independent of the final state. Moreover, if we express $\bar{u}$ as a function of only the state [cf. Eq. (5.2.16)], then $\bar{u}$ is independent of both the initial and the final states.

The heuristic derivation of $dC^{**}(y_0,y)/dy$ in based on the "differential form" of the optimality principle. That is if $u \in G$ is the optimal control function, then the derivative $dC^{**}(y_0,y)/dy$ evaluated at $u$ must be minimized for all $t \in [0,T]$. Assuming the contrary and supposing that the minimum in $dC^{**}(y_0,y)/dy$ is attained by $u_0 \neq u(t)$ at some $t$, then we can reduce the cost by operating the system with $u$ for $t \in [0, t-\Delta t] \cup [t+\Delta t, T]$ and use $u_0$ during $[t-\Delta t, t+\Delta t]$ thereby reducing the cost. This contradicts the optimality of $u$.

The multigrid principle can be viewed as an imprecise statement of the optimality principle. At each grid-level the algorithm decides the duration on the grid-level before grid-refinement and chooses the next grid-level. The choice of the duration and the next grid-level is based on minimizing $\Delta C/\Delta h$, that is, minimizing the cost incurred per improvement in approximation error. It turns out that for problems of Types I–III, iterating until the successive approximation error bound is no more than the discretization error bound, and choosing the next grid-size 1/2 of the current grid-size is sufficient to guarantee the *asymptotic* optimality of the algorithm with respect to $\epsilon$ (and $\alpha, f_1, f_2, f_3$).

## 5.3 Applications

In this section, we look at some applications of the general framework introduced in Section 5.1 and of the complexity results in Section 5.2. We consider three areas of applications: (i) the discrete-time stochastic control problem addressed in the first part of this thesis, (ii) simulated annealing, and (iii) boundary value problems. We will only present the main ideas rather than showing every step.

[†] Here, we are talking about the "forward" optimality principle, which is applicable to certain special cases like ours. The usual, more generally applicable, "backward" optimality says something weaker—that for all $t \in (0,T)$, $u$ must be the optimal control that drives the system from $x_t$ at time $t$ to $x_T$.

### 5.3.1 Discrete-Time Stochastic Control

The problem is to compute an $\epsilon$-approximation to the solution of Bellman's equation

$$TJ = J,$$

where $T$ is the dynamic programming operator introduced in Chapter 2. This problem is the focus of the first part of this report. The application of the general framework to this problem is of special significance for the following reason: It is this problem that motivated our development of the general framework. One of the original motivations is to separate the complexity analysis of the algorithms from the specific details of the problems, such as measurability issues and discretization methods. The general framework allows us to do that.

The basic ideas of discrete-time stochastic control problems are discussed in Chapter 2. We now show how this problem is modeled by a Type III problem. The notation used here is the same as in Part I of this report.

#### The Metric Space

The metric space $\mathsf{M} = \mathcal{B}(S)$, where $\mathcal{B}(S)$ is the space of all bounded Borel-measurable functions on the state space $S$. (For more details see Chapters 2 and 3.) The metric $\mathsf{d}(x, y) = \|x - y\|_\infty, x, y \in \mathcal{B}(S)$, where $\|\cdot\|_\infty$ is the sup-norm on $\mathcal{B}(S)$. We have shown (in Section 2.2) that $T$ is a contraction operator on $\mathcal{B}(S)$ with contraction factor $\alpha \in (0, 1)$ and has a unique fixed point $J^*$. Here, $\alpha$ is the discount factor and $J^*$ is the optimal cost function.

#### The Family of Contraction Processes

In Section 2.4, we constructed a family of contraction operators $\{\tilde{T}_h\}_{h\in\mathcal{H}}$, where $\mathcal{H} = (0, h_a]$ for some $h_a \in (0, 1]$. Each $\tilde{T}_h$ has contraction factor $\alpha$ and can be thought of as a piecewise constant approximation of $T$ [cf. Theorem 2.4.2]. The fixed-point of $\tilde{T}_h$, denoted by $\tilde{J}_h^*$, is a simple function on $S$.

#### Error Bounds

By Theorem 2.4.2, we have $\|J^* - \tilde{J}_h^*\|_\infty \leq K'h/(1-\alpha)^2, \forall h \in \mathcal{H}$, for some constant $K' \geq 1$. By letting $A_h = \tilde{T}_h$ and $x_h^* = \tilde{J}_h^*$, the discretization error bound is $\mathsf{D}(h) = K'h/(1-\alpha)^2$. Therefore,

$$f_3 = \frac{K'}{(1-\alpha)^2} \quad \text{and} \quad s = 1.$$

We let the grid-transfer error bound $\mathsf{B}(h, h') = f_3 h^s + f_3 h'^s$ for $h \neq h'$.

## The Iteration Cost

The cost per iteration of $\tilde{T}_h$ is the number of arithmetic operations needed in one iteration of $\tilde{T}_h$. It is clear (cf. Lemma 3.1.1) that the iteration cost $\mathsf{C}(h) = c_2 h^{-(2n+m)}$, for some constant $c_2 \geq 1$ and where $n$ and $m$ are the dimensions of the state and the control spaces, respectively. Therefore,

$$f_2 = c_2 \quad \text{and} \quad r = 2n + m.$$

## Complexity Analysis

The above problem belongs to $\mathcal{P}_\alpha(f_2, f_3, r, s)$, with $\mathcal{T} = \mathbf{Z}^0$, $\mathcal{H} = (0, h_a]$, and the values of $\alpha, f_2, f_3, r, s$ are given above. We can use the single-grid and the multigrid algorithms of Section 5.2; by Theorems 5.2.1 and 5.2.2, we have (keeping only the dependence on $\alpha$ and $\epsilon$, as $\epsilon \searrow 0$)

$$\mathsf{C}_\alpha^{\mathrm{SG}}(\epsilon) = \mathrm{O}\left(\frac{\ln[1/(1-\alpha)\epsilon]}{|\log \alpha|}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right),$$

$$\mathsf{C}_\alpha^{\mathrm{MG}}(\epsilon) = \mathrm{O}\left(\frac{1}{|\log \alpha|}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right).$$

This is in agreement with the results of Section 3.2.

## Lower Bounds

Since $\mathcal{B}(S)$ is a Banach space, the lower bound construction of Theorem 5.2.3 applies. [We can isometrically imbed the interval $[0, f_3]$ into $\mathcal{B}(S)$.] Therefore, we have

$$\mathsf{C}_\alpha^*(\epsilon) = \Omega\left(\frac{1}{|\log \alpha|}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right).$$

But the information-based lower bound (cf. Theorem 3.3.2) is

$$\mathsf{C}_{\mathrm{gen}}^{\mathrm{info}}(\alpha, \epsilon) = \Omega\left(\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right).$$

Note that even though there is a gap of order $\mathrm{O}(1/|\log \alpha|)$ between the upper and the information-based lower bound, the algorithmic-based lower bound shows that MG is optimal within the class of all algorithms of the form $\tilde{T}_{h_1}^{t_1}\tilde{T}_{h_2}^{t_2}\cdots\tilde{T}_{h_i}^{t_i}J$ using only the operators in $\{\tilde{T}_h\}$. We conclude that no choice of grid-levels and durations can close the $\mathrm{O}(1/(1-\alpha))$ "gap" between the upper bound and the information-based lower.

The general framework can also be used to analyze the complexity of the special case where a mixing condition is in effect.

## 5.3.2 Simulated Annealing

We now look at another area of application—simulated annealing; we will use the framework to solve the optimal cooling schedule problem. This application is interesting because there has been a lot of research on this subject recently. Using the general framework, we do not obtain any new results; but we get another view of the results. Again we will not discuss this application in detail; for more details see Hajek (1988) and Tsitsiklis (1989).

### Basic Ideas

We now give the basic ideas of simulated annealing. Consider a system with $N$ possible physical states $S = \{1, 2, \ldots, N\}$. Associated with the system is an energy function $J : S \mapsto [0, \infty)$ and a non-negative control parameter $T$, which represents the temperature of the system. The system changes state probabilistically, in a manner depending on the temperature. If the temperature is held constant at $T$ for a long time, the system tends to an equilibrium probability distribution $\pi_T$, where the probability that the system is in state $i$ is given by

$$\Pr(\text{state} = i) = \pi_T(i) = \frac{\mathrm{e}^{-J(i)/T}}{Z_T}, \tag{5.3.1}$$

where $Z_T = \sum_{i=1}^{N} \mathrm{e}^{-J(i)/T}$ is a normalizing constant.

Let $S^* \overset{\text{def}}{=} \{i \in S \mid J(i) \leq J(j), \forall j \in S\}$; that is, $S^*$ denotes lowest energy states of the system. For simplicity, we assume that $S^*$ is a singleton. It is easy to see that $\pi_0$, the probability distribution when $T = 0$, is concentrated on $S^*$. However, the speed of convergence to equilibrium becomes very slow as $T \searrow 0$.

Simulated annealing is a probabilistic algorithm for minimizing the cost function $J$ by simulating the physical system. It can be viewed as a descent-type algorithm with probabilistic upward transitions. Its main application is in combinatorial problems [see, for example, Kirkpatrick, et al (1983)].

### Modeling

We now describe how the general framework allows us to analyze a simulated annealing algorithm. The metric space is the set of all probability distributions on $S$; that is,

$$\mathsf{M} = \left\{ (\pi(i))_{i=1}^{N} \mid \pi(i) \geq 0, \sum_{i=1}^{N} \pi(i) = 1 \right\}.$$

The metric on $\mathsf{M}$ is the (discrete) $L_1$-norm; that is

$$\mathsf{d}(\pi, \pi') = \sum_{i=1}^{N} |\pi(i) - \pi'(i)|, \quad \pi, \pi' \in \mathsf{M}.$$

Instead of using $T$, we will use the parameter $h = \mathrm{e}^{-1/T}$ ($h = 0$ when $T = 0$). Furthermore, let $x_h^* = \pi_T$, $\mathcal{H} = (0, 1/e)$, and $A_h^t$ denotes running the simulation algorithm for the duration $t \in \mathbf{R}^+$, at temperature $T$. Under certain irreducibility and aperiodicity assumptions (at all $T > 0$), it can be shown that the convergence to equilibrium is geometric. Therefore, we can assume that $A_h$ is a contraction process on $\mathsf{M}$ with fixed-point $x_h^*$. The assumptions we make are as follows:

1. $\alpha_h = 1 - f_1 h^q$, where $f_1 > 0, q > 0$ are some constants. [Therefore, we are assuming that the rate of convergence (as a function of the temperature) is $\alpha_T = 1 - f_1 \mathrm{e}^{-q/T}$, $T > 0$.]

2. $\mathsf{C}(h) = 1, \forall h$; that is, $f_2 = 1$, $r = 0$. [Letting $\mathsf{C}(h) = 1$ makes the minimum computational cost problem into a minimum time problem; thus, the algorithm with the minimum cost corresponds to the best cooling schedule.]

3. $\mathsf{D}(h) = f_3 h^s$, for some constants $f_3, s > 0$. [We are assuming that $\mathsf{d}(\pi_0, \pi_T) \leq f_3 \mathrm{e}^{-s/T}$; this assumption follows from Eq. (5.3.1), with some additional simplifying assumptions.]

4. $a_h^{h'} \leq a_0$ for some constant $a_0$.[We are assuming that there exists some constant $a_0 \geq 1$ such that if we fix the temperature at $T > 0$, then $\mathsf{d}(\pi_T, \pi(t)) \leq \alpha_T^t a_0 \mathsf{d}(\pi_T, \pi(0))$, for all $t \geq 0$, where $\pi(t)$ denotes the probability distribution at time $t$. This assumption follows from the norm equivalence between finite dimensional norms—that is, if $\pi(t)$ converges to $\pi_T$ with respect to one norm, then it also converges, with the same contraction factor, with respect to any other norms (except that we may have to introduce an appropriate delay factor in the latter convergence). The key assumption here, is that the delay factor is independent of the temperature. This is a simplifying assumption we make.]

Under Assumptions 1–4, the above is a Type II problem,

$$\mathcal{P}_{\mathrm{II}}(\mathcal{T}, \mathcal{H}, f_1, f_2, f_3, q, r, s),$$

where the parameters are described above.

We are interested in the form of the best cooling schedule; namely, the temperature $T$ as a function of time $t$, in the limit as $t \nearrow \infty$. We can either use the lower bound result of Section 5.2.4 or the upper bound result of Section 5.2.3

To use the lower bound result, we note that for $N > 2$ ($N$ is usually a very large number), we can always imbed the interval $[0, f_3]$ into $(\mathsf{M}, \mathsf{d})$; hence, we can construct the lower bound as in Section 5.2.4. According to Eq. (5.2.22), as $t \nearrow \infty$,

$$h \sim t^{-1/q} \quad \Rightarrow \quad T \sim \frac{q}{\ln t}.$$

[The actual meaning of the lower bound to simulated annealing is unclear because of the special structure of the problem (which may make the lower bound inapplicable). The lower bound result makes sense if we do not consider the special structure but consider only the three parameters of the problem.]

Alternatively, by Theorem 5.2.2 the minimum time to reach an $\epsilon$-approximation to $\pi_0$ is

$$\mathsf{C}^*(\epsilon) = \Theta\left(\left[\frac{1}{\epsilon}\right]^{q/s}\right),$$

where the complexity bound is attained by MG. Therefore, using MG, we have $t \sim \epsilon^{-q/s}$. But since $\epsilon \sim h^s$; therefore, $t \sim h^{-q}$, or equivalently, $T \sim q/\ln t$, which agrees with the cooling schedule based on the lower bound.

In simulated annealing the variable $q$ represents the "depth" to be climbed to get to a global minimum. So a larger $q$ should require a slower cooling, which is in agreement with what we have obtained.

Lastly, note that by the optimality principle, the optimal cooling schedule for obtaining an $\epsilon$-approximation to $\pi_0$ must be a "subschedule" of the optimal cooling schedule for obtaining $\pi_0$.

### 5.3.3 Boundary Value Problems

We now use the general framework to analyze boundary value problems. Instead of considering the most general problem, we will consider a 1-dimensional two-point boundary value problem. This model problem captures the main ideas. [See Briggs (1987) for more details.] We will also show how the general framework can be adapted for the more specialized analysis used for such problems.

#### Main Ideas

Consider the domain $\Omega = [0,1]$ and the boundary $\partial\Omega = \{0,1\}$. Let $\mathcal{C}_0(\Omega)$ denote the space of all bounded continuous functions on $\Omega$ with values equal to 0 at $\partial\Omega$. Let $\sigma$ be a known positive constant and $f$ be a known function on $\Omega$ (of sufficient smoothness). We consider the following boundary value problem

$$-u''(x) + \sigma u(x) = f(x), \quad x \in (0,1), \tag{5.3.2}$$

where $u \in \mathcal{C}_0(\Omega)$ and is twice differentiable. It can be shown that this differential equation has a unique solution $u^*$. We are interested in computing an $\epsilon$-approximation to $u^*$.

#### Finite Difference Approximation

There are many ways of discretizing Eq. (5.3.2). We will consider a particular way, using finite differences. Let the discretized domain be $\Omega_h = \{ih \mid i = 1, 2, \ldots N-1\}$, where $N$ is some positive integer. Let $\mathcal{B}(\Omega_h)$ be the set of all (discrete) real-valued functions on $\Omega_h$. [Therefore elements of $\mathcal{B}(\Omega_h)$ are vectors in $\mathbf{R}^{N-1}$.]

For a fixed the grid-size $h = 1/N$, we use the second-order finite difference to approximate $u''$, so that Eq. (5.3.2) becomes

$$[-u(x_{i-1}) + 2u(x_i) - u(x_{i+1})]\, h^{-2} + \sigma u(x_i) = f(x_i), \quad 1 \le i \le N-1,$$

where $x_i = i/N$, $0 \le i \le N$. Ignoring the two end-points [$u(0)$ and $u(1)$], we can rewrite this equation as

$$\tilde{u} = (2+\sigma h^2)^{-1}[\tilde{L}_h + \tilde{U}_h]\tilde{u} + (2+\sigma)^{-1}h^2\tilde{f}_h, \quad \tilde{u} \in \mathcal{B}(\Omega_h), \tag{5.3.3}$$

where $\tilde{L}_h$ (respectively, $\tilde{U}_h$) is an $(N-1)\times(N-1)$ matrix with 1's just below (respectively, just above) the diagonal, and $\tilde{f}_h \in \mathcal{B}_c(\Omega_h)$ is a vector of $f$-values evaluated at $x_i$. Let $u_h^* \in \mathcal{B}_c(\Omega_h)$ denote the solution of Eq. (5.3.3).

Before defining the contraction processes, we first define the grid-transfer (injection and interpolation) operators. The injection operator $I_h^0 : \mathcal{C}_0(\Omega) \mapsto \mathcal{B}(\Omega_h)$ is given by

$$(I_h^0 u)(x_i) = u(x_i), \quad \forall u \in \mathcal{C}_0(\Omega), x_i \in \Omega_h.$$

[In particular, $\tilde{f}_h = I_h^0 f$.] The interpolation operator $I_0^h : \mathcal{B}(\Omega_h) \mapsto \mathcal{C}_0(\Omega)$ is defined by the property that for any $\tilde{u} \in \mathcal{B}(\Omega_h)$, $I_0^h\tilde{u}$ is a linear interpolation of $\tilde{u}$ with value 0 at $\partial\Omega$. Note that $I_h^0 I_0^h$ is the identity operator on $\mathcal{B}(\Omega_h)$.

## A Family of Contraction Operators

Let $\tilde{B}_h = (2+\sigma h^2)^{-1}\left[\tilde{L}_h + \tilde{U}_h\right]$. It is easy to show that the induced norm of $\tilde{B}_h$ is

$$\|\tilde{B}_h\|_\infty = (1+\sigma h^2/2)^{-1} \sim 1 - \frac{\sigma}{2}h^2.$$

Therefore, $\tilde{B}_h$ is a contraction operator on $\mathcal{B}(\Omega_h)$ with respect to the (discrete) sup-norm.

We now define the metric space and the family of contraction processes. We begin by defining the set of grid-levels and the metric space. Let $\mathcal{H} = \{1, 1/2, \ldots, \}$; $\mathsf{M} = \mathcal{C}_0(\Omega)$; $\mathsf{M}_h = \{I_0^h\tilde{u} \mid \tilde{u} \in \mathcal{B}(\Omega_h)\}$ for each $h \in \mathcal{H}$; and the metric $\mathsf{d}(x,y) = \|x-y\|_\infty$ for all $x, y \in \mathsf{M}$.

Furthermore, for each $h \in \mathcal{H}$, we define an operator $\tilde{A}_h : \mathcal{B}(\Omega_h) \mapsto \mathcal{B}(\Omega_h)$,

$$\tilde{A}_h\tilde{u} = \tilde{B}_h\tilde{u} + (2+\sigma)^{-1}h^2\tilde{f}_h, \qquad \tilde{u} \in \mathcal{B}(\Omega_h).$$

It is clear that $\tilde{A}_h$ is a contraction operator on $(\mathcal{B}(\Omega_h), \|\cdot\|_\infty)$ with contraction factor $\alpha_h = \|\tilde{B}_h\|_\infty \sim 1 - \sigma h^2/2$.

Finally, we define the family of contraction processes by letting

$$A_h^t u = I_0^h \tilde{A}_h^t I_h^0, \qquad u \in \mathsf{M}.$$

It is clear that $x_h^* \overset{\text{def}}{=} I_0^h\tilde{u}_h^*$ is the fixed point of $A_h$; moreover it can be shown that

$$\mathsf{d}(x_h^*, A_h^t u) \le \alpha_h^t \mathsf{d}(x_h^*, \tilde{u}), \quad \forall t \in \mathbf{Z}^0, u \in \mathsf{M}.$$

[It is easy to check that $\mathsf{d}(x_h^*, I_0^h I_h^0 u) \le \mathsf{d}(x_h^*, u)$.] Thus, we have constructed a family of contraction processes with contraction factor $\alpha_h \sim 1 - \sigma h^2/2$ and delay factor 1.

Furthermore, we will assume that the discretization error is $\mathrm{O}(h^2)$ and the cost of each iteration of $A_h$ is $\mathrm{O}(1/h)$. (We can use the delay factor to model the computational cost of the injection and interpolation operators, if necessary.) In summary, we have

1. $\alpha_h = 1 - f_1 h^q$, where $f_1 = \sigma/2$ and $q = 2$.
2. $\mathsf{C}(h) = f_2 h^{-r}$, where $f_2$ is some positive constant and $r = 1$.
3. $\mathsf{D}(h) = f_3 h^s$, where $f_3$ is some positive constant and $s = 2$.
4. the delay factor $a_h^{h'}$ is chosen accordingly.

[Note that the above construction is to illustrate how to use the general framework to model the kinds of analyses used in boundary value problems; we are not suggesting any changes in the actual implementation of the algorithm.]

We are only interested in the dependence on $\epsilon$, as $\epsilon \searrow 0$. Under Assumptions 1–4, the problem belongs to Class II; therefore, using MG, the basic multigrid algorithm, we obtain

$$\mathsf{C}^{\mathrm{MG}}(\epsilon) = \mathrm{O}\left(\left[\frac{1}{\epsilon}\right]^{3/2}\right).$$

Since we can construct the lower bound of Section 5.2.4 in $\mathsf{M}$, the algorithmic-based lower bound within the framework is

$$\mathsf{C}^*(\epsilon) = \Omega\left(\left[\frac{1}{\epsilon}\right]^{3/2}\right).$$

However, the complexity can be improved by exploiting the special structure of the problem. We will consider one such improvement.

## The Full Multigrid V-Cycle

The complexity of MG is not good compared with the *Full Multigrid V-cycle (FMV)* algorithm, which is used in practice. [For more details on the FMV algorithm, see Briggs (1987), Douglas (1984), or Hackbusch (1985).] The FMV algorithm uses defect correction to exploit the special structure of the problem; one obtains a contraction factor independent of the discretization—a Type III problem. If we now let each iteration of $A_h$ correspond to a V-cycle then everything in the preceding subsection remains the same, except that the contraction factor $\alpha_h = \alpha$, independent of $h$ (namely, $q = 0$). Thus we can apply the analysis in Section 5.2.3 and show that the complexity is $\mathrm{O}\left([1/\epsilon]^{1/2}\right)$ which agrees with the known result [see Briggs (1987) and Hackbusch (1985).] Thus, we can view the FMV-cycle as a one-way multigrid algorithm but each iteration is now a V-cycle. This analysis also illustrates the limitations of the lower bound result of Theorem 5.2.3, the result only hold with respect to the family of available algorithms and does not rule out using the problems' special structure to get a better complexity.

## Notes

The actual analysis of the FMV-cycle uses the discrete $L_2$-norm to exploit the different rate of convergence for different frequency components. However, in our framework we use the continuous sup-norm which is the essentially the same as the discrete sup-norm. And by norm equivalence the convergence rate with respect to the discrete sup-norm and the discrete $L_2$ is the same; but it may take a few iterations before the convergence with respect to the discrete sup-norm appears. We can handle this in our framework by an appropriate choice of delay factor. (More study is needed to determine the proper choice of delay factor.) Alternatively, we can use the continuous $L_2$-norm instead of the sup-norm.

Finally, the general framework may be used to analyze continuous-time stochastic control problems and model some of the algorithms in Hoppe (1986) and Akian, et al (1988). (Again more analysis is needed.)

### 5.3.4 Summary

We have shown that the general framework can model a wide variety of relaxation problems. However more study is needed to determine how the delay factor should be chosen to model boundary value and continuous-time stochastic control problems. The advantage of using the general framework is that we can simply use the results of Section 5.2 for the complexity analysis, without having to redo the analysis for each problem. Furthermore, the framework also provides a unifying view of these problems.

## 5.4 General Results for MCCP's

In this section we present some general results for Minimum Computational Cost Problems (MCCP's). First, we show that the basic bound is in a certain sense optimal. Second, we show under certain monotonicity assumptions that one-way multigrid algorithms are in a certain sense optimal. Finally, we explore the relationship between the discretization error bound and the grid-transfer error bound, and prove an intuitive result (see Theorem and Corollary 5.4.3).

### 5.4.1 Optimality of The Basic Bound

Recall (see Lemma 5.1.1) that the basic bound $\mathsf{p}(x_h^*, A_{\mathcal{H}}^{\eta} x_{h_0}^*)$ allows us to upper bound $\mathsf{d}(x_h^*, A_{\mathcal{H}}^{\eta} x_{h_0}^*)$ for all instances. Some of the reasons for using the basic bound are: (i) it is simple to use, (ii) it gives good results (cf. the optimality results in Section 5.2), and (iii) it is intuitively clear that it is the best upper bound within the framework.

In this subsection, we reinforce the last reason by showing that the basic bound is indeed a tight upper bound on $\mathsf{d}(x_h^*, A_{\mathcal{H}}^{\eta} x_{h_0}^*)$; thus, providing a mathematical justification for using the basic bound. Moreover, this result allows us to develop further results on the MCCP's, which we will discuss in the later subsections.

### The Main Idea

We show that the basic bound is a tight upper bound by constructing an instance for which $\mathsf{d}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*) = \mathsf{p}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*)$; therefore, in general, there can be no better upper bound on $\mathsf{d}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*)$ than the basic bound (unless more information is provided about the problem). For the rest of this subsection let us fix an MCCP $\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h))$.

### The Construction

We now construct the instance $\iota = \left(\mathsf{M}, \mathsf{d}, \{A_h\}_{h\in\mathcal{H}}, \{x_h^*\}_{h\in\mathcal{H}_0}\right)$ with the desired properties. Let $\mathsf{M}_h = \left\{(h, x_2) \mid x_2 \in [0,\infty)\right\}$, $h \in \mathcal{H}$, $\mathsf{M}_0 = \left\{(0,0)\right\}$, $\mathsf{M} = \bigcup_{h\in\mathcal{H}_0} \mathsf{M}_h$, and $x_h^* = (h, 0)$, $h \in \mathcal{H}_0$.

The metric is defined, for $h, h' \in \mathcal{H}_0$,

$$\mathsf{d}(x_h^*, x_{h'}^*) = \mathsf{B}(h,h').$$

Since $\mathsf{B}$ is a metric, we have

$$\mathsf{d}(x_h^*, x_{h'}^*) \le \mathsf{d}(x_h^*, x_{h''}^*) + \mathsf{d}(x_{h''}^*, x_{h'}^*), \quad \forall h, h', h'' \in \mathcal{H}_0.$$

More generally, for $x = (h, x_2)$ and $y = (h', y_2)$,

$$\mathsf{d}(x,y) = \begin{cases} |x_2 - y_2| & \text{if } h = h'; \\ x_2 + y_2 + \mathsf{d}(x_h^*, x_{h'}^*) & \text{otherwise.} \end{cases} \tag{5.4.1}$$

To verify that $\mathsf{d}$ is a metric on $\mathsf{M}$, we only need to verify the triangle inequality. Let $z = (h'', z_2)$, then

$$\mathsf{d}(x,z) = \begin{cases} |x_2 - z_2| & \text{if } h = h''; \\ x_2 + z_2 + \mathsf{d}(x_h^*, x_{h''}^*) & \text{otherwise.} \end{cases}$$
$$\mathsf{d}(z,y) = \begin{cases} |z_2 - y_2| & \text{if } h'' = h'; \\ z_2 + y_2 + \mathsf{d}(x_{h''}^*, x_{h'}^*) & \text{otherwise.} \end{cases}$$

We would like to show that

$$\mathsf{d}(x,y) \le \mathsf{d}(x,z) + \mathsf{d}(z,y), \quad \forall x, y, z \in \mathsf{M}. \tag{5.4.2}$$

By doing a case by case analysis (when $h = h'$, $h = h''$, $h' = h''$, and all the $h$'s are distinct) it is easy to verify Eq. (5.4.2).

The contraction process $A_h : \mathsf{M} \mapsto \mathsf{M}_h$ is given by

$$A_h^t x = (h, \alpha_h^t a_h^{h'} \mathsf{d}(x_h^*, x)), \quad t \in \mathcal{T}^+, h, h' \in \mathcal{H}, x \in \mathsf{M}_{h'}. \tag{5.4.3}$$

It is clear that $A_h$ is a contraction process on $\mathsf{M}$ with contraction factor $\alpha_h$, delay factor $a_h^{h'}$ and fixed point $x_h^* = (h, 0)$.

It remains to verify that

$$\mathsf{d}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*) = \mathsf{p}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*), \quad \forall h, h_0 \in \mathcal{H}_0, \eta \in \mathcal{A}_0. \tag{5.4.4}$$

For the null trajectory, Eq. (5.4.4) is trivially true. We now assume, as our induction hypothesis, that Eq. (5.4.4) holds for all trajectories with length $i-1$, $i \in \mathbf{Z}^+$, and we show it for trajectories of length $i$.

Let $\eta$ be a trajectory of length $i-1$. If $\eta$ is a null trajectory then let $a_{h_i}^- = a_{h_i}^{h_0}$; otherwise, let $a_{h_i}^- = a_{h_i}^{h_{i-1}}$, where $h_{i-1}$ is the final grid-level of $\eta$. Using Eqs. (5.4.1) and (5.4.3), the induction hypothesis, and the definition of $\mathsf{p}$, we obtain

$$\begin{aligned}
\mathsf{d}(x_h^*, A_{h_i}^{t_i} A_{\mathcal{H}}^\eta x_{h_0}^*) &= \mathsf{d}(x_h^*, x_{h_i}^*) + \mathsf{d}(x_{h_i}^*, A_{h_i}^{t_i} A_{\mathcal{H}}^\eta x_{h_0}^*) \\
&= \mathsf{d}(x_h^*, x_{h_i}^*) + \alpha_{h_i}^{t_i} a_{h_i}^- \mathsf{d}(x_{h_i}^*, A_{\mathcal{H}}^\eta x_{h_0}^*) \\
&= \mathsf{p}(x_h^*, x_{h_i}^*) + \alpha_{h_i}^{t_i} a_{h_i}^- \mathsf{p}(x_{h_i}^*, A_{\mathcal{H}}^\eta x_{h_0}^*) \\
&= \mathsf{p}(x_h^*, A_{h_i}^{t_i} A_{\mathcal{H}}^\eta x_{h_0}^*),
\end{aligned}$$

which completes the induction step. Thus we have proved the following result.

**Proposition 5.4.1** *For any MCCP $\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h))$, there exists an instance $\iota \in \mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h))$ such that there holds,*

$$\mathsf{d}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*) = \mathsf{p}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*), \quad \forall h, h_0 \in \mathcal{H}_0, \eta \in \mathcal{A}_0.$$

## An Extension

If the metric space is specified by the problem, we expect to be able to obtain tighter bounds on $\mathsf{d}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*)$ using our knowledge about the metric space. However, the result of Proposition 5.4.1 can be extended to these cases whenever we can isometrically imbed the metric space constructed here into the specified metric space. Once that is done, the rest of the construction and, thus, the optimality of the basic bound follow.

## Significance of The Basic Bound

Fix some MCCP

$$\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h)).$$

Note that the bound $\mathsf{d}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*) \le \mathsf{p}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*)$ holds for all instances. Moreover, we can always find an instance for which the bound is an equality. Thus, we have $\mathsf{d}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*) \le \epsilon$ for all instances iff $\mathsf{p}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*) \le \epsilon$. Therefore, we can define $A_{\mathcal{H}}^\eta x_{h_0}^*$ to be an $\epsilon$-approximation of $x_h^*$ iff $\mathsf{p}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*) \le \epsilon$. The significance of this result is that we can work with the basic bound instead of the metric.

Finally, we prove a useful lemma.

**Lemma 5.4.1** *There holds,*

$$\mathsf{p}(x_h^*, A_{\mathcal{H}}^\eta x_{h_0}^*) \le \mathsf{p}(x_h^*, x_{h'}^*) + \mathsf{p}(x_{h'}^*, A_{\mathcal{H}}^\eta x_{h_0}^*), \quad \forall h, h', h_0 \in \mathcal{H}_0, \eta \in \mathcal{A}_0.$$

**Proof** For $|\eta| = 0$ the results follows trivially from the triangle inequality. We now prove the lemma by induction on $|\eta|$. Suppose the results hold for all trajectories $\eta$ with $|\eta| \leq i-1$. (Let the final grid-level of $\eta$ be $h_{i-1}$.) We now prove it for trajectories of length $i$.

Using the definition of $\mathsf{p}$ [Eqs. (5.1.16)-(5.1.17)] and the fact that $\mathsf{B}$ is a metric, we have

$$\begin{aligned}\mathsf{p}(x_h^*, A_{h_i}^{t_i} A_{\mathcal{H}}^{\eta} x_{h_0}^*) &= \mathsf{p}(x_h^*, x_{h_i}^*) + \alpha_{h_i}^{t_i} a_{h_i}^{h_{i-1}} \mathsf{p}(x_{h_i}^*, A_{\mathcal{H}}^{\eta} x_{h_0}^*) \\ &\leq \mathsf{p}(x_h^*, x_{h'}^*) + \mathsf{p}(x_{h'}^*, x_{h_i}^*) + \alpha_{h_i}^{t_i} a_{h_i}^{h_{i-1}} \mathsf{p}(x_{h_i}^*, A_{\mathcal{H}}^{\eta} x_{h_0}^*) \\ &\leq \mathsf{p}(x_h^*, x_{h'}^*) + \mathsf{p}(x_{h'}^*, A_{h_i}^{t_i} A_{\mathcal{H}}^{\gamma} x_{h_0}^*),\end{aligned}$$

as required. □

## 5.4.2 The Optimality of One-Way Multigrid Algorithms

In this subsection we show that, under certain monotonicity assumptions on the problem parameter, one-way multigrid algorithms (algorithms that proceed from coarse to fine grid-levels) are in a certain sense optimal; more precisely, one can always replace a non-one-way algorithm by a one-way (in fact, a strictly one-way) algorithm whose complexity is just as good as the original, if not better.

We begin by introducing some terminology and the assumptions.

### Terminology and Assumptions

Let $f$ be a real-valued function on a subset of the real line. We say that $f$ is *monotone increasing* (respectively, *monotone decreasing* ) if $f(h) \leq f(h')$ [respectively, $f(h) \geq f(h')$] for all $h < h'$. It is *strictly* monotone if the inequality is strict.

We now introduce the following assumptions.

**G.0** 0 is a limit point of $\mathcal{H}$ and $\lim_{h \searrow 0} \mathsf{D}(h) = 0$.

**G.1** $\alpha_h$ is a monotone decreasing function of $h$.

**G.2** $\mathsf{C}(h)$ is a monotone decreasing function of $h$.

**G.3** $a_h^{h'} = a_h^{h''}$ for all $h, h', h'' \in \mathcal{H}$.

Note that we can also consider strict versions of Assumptions G.1–G.2, where "monotone" is replaced by "strictly monotone". And Assumption G.3 says that the delay factor may depend on the final grid-level but is independent of the initial grid-level.

The following theorem and corollary show the "optimality" of one-way trajectories and algorithms—they are all that is needed for the minimum computational cost problem.

**Theorem 5.4.1** *Suppose that the problem $\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h))$ satisfies Assumption G.1–G.3. Then for any $\eta \in \mathcal{A}_0$, there exists an $\tilde{\eta} \in \mathcal{A}_0$ such that*

*1. $\tilde{\eta}$ is a strictly one-way trajectory.*

*2. The set of grid-levels of $\tilde{\eta}$ is a subset of that of $\eta$; $\tilde{\eta}$ and $\eta$ have the same final grid-level; and $|\tilde{\eta}| \leq |\eta|$.*

*3. $\mathsf{C}_{\mathcal{H}}^{\tilde{\eta}} \leq \mathsf{C}_{\mathcal{H}}^{\eta}$,*

*4. $\mathsf{p}(x_h^*, A_{\mathcal{H}}^{\tilde{\eta}} x_{h_0}^*) \leq \mathsf{p}(x_h^*, A_{\mathcal{H}}^{\eta} x_{h_0}^*)$ for all $h, h_0 \in \mathcal{H}_0$.*

**Proof** We can construct $\tilde{\eta}$ from $\eta$ as follows. Start by setting the final duration and grid-level pair of $\tilde{\eta}$ to be that of $\eta$. Continue doing this for $\tilde{\eta}$, until a duration grid-level pair that will destroy the monotonicity of $\tilde{\eta}$ is encountered. If so, use the coarsest grid-level of $\tilde{\eta}$, instead. This construction is the basic idea behind the following inductive proof.

We will prove the theorem by induction on the length of $\eta$. The result is trivially true when $|\eta| \leq 1$. Our induction hypothesis is that the result hold for all trajectories of length $i-1$, where $i \geq 2$. We now show it for trajectories of length $i$.

Let $\eta'$ be a trajectory of length $i-1$ and with final grid-level $h'$; let $\eta = ((t_i, h_i)) \cdot \eta'$. Using the definition of $\mathsf{p}$ and the definition of $\mathsf{C}$, we obtain

$$\begin{aligned} \mathsf{p}(x_h^*, A_{\mathcal{H}}^{\eta} x_{h_0}^*) &= \mathsf{p}(x_h^*, A_{h_i}^{t_i} A_{\mathcal{H}}^{\eta'} x_{h_0}^*) \\ &= \mathsf{p}(x_h^*, x_{h_i}^*) + \alpha_{h_i}^{t_i} a_{h_i}^{h'} \mathsf{p}(x_{h_i}^*, A_{\mathcal{H}}^{\eta'} x_{h_0}^*); \qquad (5.4.5) \\ \mathsf{C}_{\mathcal{H}}^{\eta} &= t_i \mathsf{C}(h_i) + \mathsf{C}_{\mathcal{H}}^{\eta'}. \qquad (5.4.6) \end{aligned}$$

From Eqs. (5.4.5) and (5.4.6), we see that if we apply our induction hypothesis to $\eta'$ we can assumed that $\eta'$ is a strictly one-way trajectory. And we are done if $h_i < h'$. Therefore it remains to prove the result when $h_i \geq h'$.

Let us suppose that $\eta' = ((t', h')) \cdot \eta''$. And if $\eta''$ is a null trajectory, then let $a_{h_i}^{-} = a_{h_i}^{h_0}$; otherwise, let $a_{h_i}^{-} = a_{h_i}^{h''}$, where $h''$ is the final grid-level of $\eta''$.

Using Eq. (5.4.5), the definition of $\mathsf{p}$, Lemma 5.4.1, and the facts $a_{h_i}^{-} \geq 1$, $\alpha_{h_i} \leq \alpha_{h'} < 1$, and $a_{h_i}^{h'} = a_{h_i}^{-}$, we have

$$\begin{aligned} \mathsf{p}(x_h^*, A_{\mathcal{H}}^{\eta} x_{h_0}^*) &= \mathsf{p}(x_h^*, x_{h_i}^*) + \alpha_{h_i}^{t_i} a_{h_i}^{h'} \left[\mathsf{p}(x_{h_i}^*, x_{h'}^*) + \alpha_{h'}^{t'} a_{h'}^{-} \mathsf{p}(x_{h'}^*, A_{\mathcal{H}}^{\eta''} x_{h_0}^*)\right] \\ &\geq \mathsf{p}(x_h^*, x_{h_i}^*) + \alpha_{h_i}^{t_i} \alpha_{h'}^{t'} a_{h_i}^{h'} \left[\mathsf{p}(x_{h_i}^*, x_{h'}^*) + \mathsf{p}(x_{h'}^*, A_{\mathcal{H}}^{\eta''} x_{h_0}^*)\right] \\ &\geq \mathsf{p}(x_h^*, x_{h_i}^*) + \alpha_{h_i}^{(t_i+t')} a_{h_i}^{-} \left[\mathsf{p}(x_{h_i}^*, A_{\mathcal{H}}^{\eta''} x_{h_0}^*)\right] \\ &= \mathsf{p}(x_h^*, A_{\mathcal{H}}^{((t_i+t', h_i))\cdot\eta''} x_{h_0}^*). \end{aligned}$$

We now apply the induction hypothesis to $((t_i + t', h_i)) \cdot \eta''$ (since its length is $i-1$) to get the required strictly one-way sub-trajectory $\tilde{\eta}$. It remains to verify that $\tilde{\eta}$ satisfies the cost requirements.

From Eq. (5.4.6) and using the monotonicity of $\mathsf{C}$, we have

$$\begin{aligned} \mathsf{C}_{\mathcal{H}}^{\eta} &= t_i \mathsf{C}(h_i) + t' \mathsf{C}(h') + \mathsf{C}_{\mathcal{H}}^{\eta''} \\ &\geq (t_i + t') \mathsf{C}(h_i) + \mathsf{C}_{\mathcal{H}}^{\eta''} \\ &= \mathsf{C}_{\mathcal{H}}^{((t_i+t', h_i))\cdot\eta''} \\ &\geq \mathsf{C}_{\mathcal{H}}^{\tilde{\eta}}, \end{aligned}$$

where the last inequality follows from the induction hypothesis.

□

Note that under the strict version of Assumptions G.1–G.2, if $\eta$ is non-one-way, then Theorem 5.4.1 can be strengthened, with the inequalities in Parts 2, 3, and 4 replaced by strict inequalities. In this case one-way trajectories are strictly better than non-one-way trajectories.

The following corollary follows immediately.

**Corollary 5.4.1** *Under the hypothesis of Theorem 5.4.1 and Assumption G.0, for all $h_0, h \in \mathcal{H}_0$ and for any algorithm $\gamma \in G(x^*_{h_0}, x^*_h)$, there exists a strictly one-way algorithm $\tilde{\gamma} \in G(x^*_{h_0}, x^*_h)$ such that*

$$\mathsf{C}^{\tilde{\gamma}}(\epsilon) \leq \mathsf{C}^{\gamma}(\epsilon), \quad \forall \epsilon > 0.$$

*In particular,*

$$\mathsf{C}^*(x^*_{h_0}, x^*_h; \epsilon) = \inf_{\gamma \in G(x^*_{h_0}, x^*_h)} \left\{ \mathsf{C}^{\gamma}(\epsilon) \,\middle|\, \gamma \text{ is strictly one-way} \right\}, \quad \forall h_0, h \in \mathcal{H}_0, \epsilon > 0.$$

Theorem 5.4.1 is surprising since it only requires the monotonicity of $\alpha_h$ and $\mathsf{C}(h)$ [Assumptions G.1–G.2] and does not require any monotonicity in the discretization error bound $\mathsf{D}$ or the grid-transfer error bound $\mathsf{B}$. (We "need" Assumption G.3 to rule out cases where the delay factor "prohibits" certain grid-level changes.) Moreover, the "optimality" of one-way algorithm applies to approximating $x^*_h$, for any $h \in \mathcal{H}_0$ and starting from any initial point $x^*_{h_0}$. The next theorem clarifies the latter issue. But we need to introduce the following simplifying restriction on the delay factor.

**G.4** $a^h_{h'} = a^h_{h''}$ for all $h, h', h'' \in \mathcal{H}$.

[The above assumption says that the delay factor may depend on the initial grid-level but is independent of the final grid-level. So, Assumptions G.3–G.4 say that the delay factor is a constant, independent of the grid-levels.]

**Theorem 5.4.2** *Under the hypothesis of Theorem 5.4.1 and Assumption G.4, and for all $h, h_0 \in \mathcal{H}_0$, $\eta \in \mathcal{A}_0$, there exists a strictly one-way trajectory $\tilde{\eta} \in \mathcal{A}_0$ with final grid-level no less than $h$, such that*

$$\mathsf{C}^{\tilde{\eta}}_{\mathcal{H}} \leq \mathsf{C}^{\eta}_{\mathcal{H}} \quad \textit{and} \quad \mathsf{p}(x^*_h, A^{\tilde{\eta}}_{\mathcal{H}} x^*_{h_0}) \leq \mathsf{p}(x^*_h, A^{\eta}_{\mathcal{H}} x^*_{h_0}).$$

**Proof** By Theorem 5.4.1 we can assume that $\eta$ is one-way. Moreover, the result is trivially true if the final grid-level of $\eta$ is no less than $h$; in particular, it is trivially true for the null trajectory.

Let us suppose that $\eta = ((t', h')) \cdot \eta''$, where $h' < h$. And if $\eta''$ is a null trajectory, let $a^-_{h'} = a^{h_0}_{h'}$; otherwise, let $a^-_{h'} = a^{h''}_{h'}$, where $h''$ is the final grid-level of $\eta''$.

Using the definition of $\mathsf{p}$ we obtain

$$\begin{aligned}\mathsf{p}(x_h^*, A_{\mathcal{H}}^{\eta} x_{h_0}^*) &= \mathsf{p}(x_h^*, x_{h'}^*) + \mathsf{p}(x_{h'}^*, A_{h'}^{t'} A_{\mathcal{H}}^{\eta''} x_{h_0}^*) \\ &= \mathsf{p}(x_h^*, x_{h'}^*) + \alpha_{h'}^{t'} a_{h'}^{-} \mathsf{p}(x_{h'}^*, A_{\mathcal{H}}^{\eta''} x_{h_0}^*).\end{aligned}$$

We now consider the following two cases: (i) $\alpha_{h'}^{t'} a_{h'}^{-} \le 1$, (ii) $\alpha_{h'}^{t'} a_{h'}^{-} > 1$.

For Case (i), we have

$$\begin{aligned}\mathsf{p}(x_h^*, A_{\mathcal{H}}^{\eta} x_{h_0}^*) &\ge \alpha_{h'}^{t'} a_{h'}^{-} [\mathsf{p}(x_h^*, x_{h'}^*) + \mathsf{p}(x_{h'}^*, A_{\mathcal{H}}^{\eta''} x_{h_0}^*)] \\ &\ge \alpha_h^{t'} a_h^{-} [\mathsf{p}(x_h^*, A_{\mathcal{H}}^{\eta''} x_{h_0}^*)] \\ &= \mathsf{p}(x_h^*, A_{\mathcal{H}}^{((t',h))\cdot\eta''} x_{h_0}^*),\end{aligned}$$

where the last inequality follows from the monotonicity of $\alpha_h$, Assumption G.4, and Lemma 5.4.1. We now can apply Theorem 5.4.1 to the trajectory $((t', h)) \cdot \eta''$ to obtained the desired one-way trajectory $\tilde{\eta}$ (which has final-grid level $h$).

For Case (ii), we have

$$\begin{aligned}\mathsf{p}(x_h^*, A_{\mathcal{H}}^{\eta} x_{h_0}^*) &\ge \mathsf{p}(x_h^*, x_{h'}^*) + \mathsf{p}(x_{h'}^*, A_{\mathcal{H}}^{\eta''} x_{h_0}^*) \\ &\ge \mathsf{p}(x_h^*, A_{\mathcal{H}}^{\eta''} x_{h_0}^*).\end{aligned}$$

Now if the final grid-level of $\eta''$ is no less than $h$, then we are done; otherwise, we apply the same argument to $\eta''$ until either Case (i) occurs or we get a null trajectory. In both cases the result follows.

□

Note that under the strict versions of Assumption G.1 and G.2 and if the final grid-level of $\eta$ is less than $h$, then we can show that $\tilde{\eta}$ is strictly better than $\eta$.

The following corollary is now immediate:

**Corollary 5.4.2** *Under the hypothesis of Corollary 5.4.1 and Assumption G.4. For any $h_0, h \in \mathcal{H}_0$, $\gamma \in G(x_{h_0}^*, x_h^*)$, there exists a strictly one-way algorithm $\tilde{\gamma} \in G(x_{h_0}^*, x_h^*)$ such that for all $\epsilon > 0$, the final grid-level of $\tilde{\gamma}(\epsilon)$ is no less than $h$ and $\mathsf{C}^{\tilde{\gamma}}(\epsilon) \le \mathsf{C}^{\gamma}(\epsilon)$. In particular,*

$$\mathsf{C}^*(x_{h_0}^*, x_h^*; \epsilon) = \inf_{\gamma \in G(x_{h_0}^*, x_h^*)} \Big\{ \mathsf{C}^{\gamma}(\epsilon) \,\Big|\, \gamma \text{ is strictly one-way and the final grid-level of } \gamma(\epsilon) \text{ is no less than } h \Big\}, \quad \forall h_0, h \in \mathcal{H}_0, \epsilon > 0.$$

In summary, to compute $x_h^*$, one can always use a one-way algorithm and without using any grid-level less than $h$.

**Some Observations**

We now discuss some implications of Theorem and Corollary 5.4.1. This result is especially instructive when applied to the following situation:

Suppose that $\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h,h'), \mathsf{C}(h))$ is an MCCP where the discretization error bound $\mathsf{D}(h) = \mathsf{B}(0,h)$ decreases to 0 monotonically as $h \downarrow 0$. We assume that $\alpha_h$ and $\mathsf{C}(h)$ are some arbitrary functions—but such that we can reorder $\mathcal{H}$ to give a new problem $\mathcal{P}(\mathcal{T}, \mathcal{H}, \hat{\alpha}_h, \{\hat{a}_h^{h'}\}, \hat{\mathsf{B}}(h,h'), \hat{\mathsf{C}}(h))$ where $\hat{\alpha}_h$ and $\hat{\mathsf{C}}(h)$ now satisfy Assumptions G.1–G.2. [For example, this can always be done when $\mathsf{C}(h)$ is a constant.] We now apply Theorem and Corollary 5.4.1 to the new problem and conclude that one-way trajectories and algorithms are "optimal" for approximating $x_h^*$. But with respect to the original problem, we make the following conclusion: The choice of grid-level should always proceed from those with good contraction factor and, by assumption, low cost to those with worse contraction factor and high cost.

There is a simple explanation for the above conclusion. We observe that in computing an $\epsilon$-approximation to $x_h^*$ one need not go below [with respect to the monotonicity of $\alpha_h$ and $\mathsf{C}(h)$] grid-level $h$ [Cf. Theorem and Corollary 5.4.2.] This is because, for $h \neq 0$, if we take the single-grid algorithm at grid-level $h$ as the "standard algorithm" there is no reason for going to a grid-level with a worse contraction and a higher cost; since it will only cost more and introduce an additional grid-transfer error. However, one may choose a grid-level, say $h'$, with a better contraction factor and, hence, a lower cost than that of $h$, provided the improvement is worth the price in the grid-transfer error. [There is always a grid-transfer error $\mathsf{p}(x_h^*, x_{h'}^*)$ because the final result must be compared at grid-level $h$—a peculiarity of the basic bound.] Applying the same reasoning to the grid-level $h'$, one may choose a grid-level with better contraction factor and lower cost, say $h''$; again provided the savings is worth the cost due to the grid-transfer error $\mathsf{p}(x_{h'}^*, x_{h''}^*)$. Using this chain of reasoning we conclude that the best algorithm must be one-way with respect to the monotonicity in $\alpha_h$ and $\mathsf{C}(h)$, and the form of $\mathsf{D}(h)$ or $\mathsf{B}(h,h')$ is irrelevant.

Finally, note that in order for Theorems and Corollaries 5.4.1–5.4.2 to be true, we need assumptions such as G.3 and G.4 to rule out cases where the delay factor "prohibits" certain grid-level changes. However, we should be able to prove these results under weaker conditions (such as monotonicity), and understand better the effects of the delay factors. This is a topic for future research.

## 5.4.3 Discretization and Grid-Transfer Error Bounds

In this subsection we explore the connection between the discretization error bound $\mathsf{D}(h)$ and the grid-transfer error bound $\mathsf{B}(h,h')$. This will help us better understand the structural properties of $\mathsf{B}(h,h')$ and how they relate to the monotonicity of $\mathsf{D}(h)$. We will also prove an intuitive theorem and corollary at the end of the subsection.

Suppose that we are given a discretization error bound $\mathsf{D}(h)$. We can form the following grid-transfer error bound $\mathsf{B}_1 : \mathcal{H}_0 \times \mathcal{H}_0 \mapsto \mathbf{R}^0$,

$$\mathsf{B}_1(h,h') \stackrel{\text{def}}{=} \begin{cases} \mathsf{D}(h) + \mathsf{D}(h') & \text{if } h \neq h'; \\ 0 & \text{otherwise.} \end{cases} \tag{5.4.7}$$

It follows from the triangle inequality [cf. Eq. (5.1.6)] that

$$\mathsf{d}(x_h^*, x_{h'}^*) \leq \mathsf{B}_1(h,h'), \qquad \forall h, h' \in \mathcal{H}_0.$$

Thus, $\mathsf{B}_1$ is indeed a grid-transfer error bound. (It is easy to see that $\mathsf{B}_1$ is a metric on $\mathcal{H}_0$.)

Even though the above bound is, in general, the "best bound" on $\mathsf{d}(x_h^*, x_{h'}^*)$, it is too "pessimistic", in practice. A more "optimistic" grid-transfer error bound is

$$\mathsf{B}_0(h,h') \stackrel{\text{def}}{=} |\mathsf{D}(h) - \mathsf{D}(h')|, \quad \forall h, h' \in \mathcal{H}_0. \tag{5.4.8}$$

[It is also easy to check that $\mathsf{B}_0$ is a metric on $\mathcal{H}_0$.] This bound is not always valid, but it is motivated by the observation (from the triangle inequality) that

$$\mathsf{d}(x_h^*, x_{h'}^*) \geq |\mathsf{d}(x_h^*, x^*) - \mathsf{d}(x^*, x_{h'}^*)|, \quad \forall h, h' \in \mathcal{H}. \tag{5.4.9}$$

And if the discretization error is "tight" [that is, $\mathsf{d}(x^*, x_h^*) \approx \mathsf{D}(h)$], then the grid-transfer error bound of Eq. (5.4.8) is indeed "optimistic. [It is also clear from Eq. (5.4.9) that if $\mathsf{d}(x_h^*, x_{h'}^*) = 0$, then $\mathsf{d}(x^*, x_h^*) = \mathsf{d}(x^*, x_{h'}^*)$—so, the optimistic bound may be "too optimistic".]

To model the whole spectrum [Eqs. (5.4.7)–(5.4.8)], we introduce the *grid-transfer parameter* $\beta \in [0,1]$ and assume that

$$\mathsf{d}(x_h^*, x_{h'}^*) \leq \mathsf{B}_\beta(h,h'), \quad \forall h, h' \in \mathcal{H}_0,$$

where $\mathsf{B}_\beta : \mathcal{H}_0 \times \mathcal{H}_0 \mapsto \mathbf{R}$,

$$\mathsf{B}_\beta(h,h') \stackrel{\text{def}}{=} \begin{cases} |\mathsf{D}(h) - \mathsf{D}(h')| + 2\beta[\mathsf{D}(h) \wedge \mathsf{D}(h')] & \text{if } h \neq h'; \\ 0 & \text{otherwise.} \end{cases} \tag{5.4.10}$$

The parameter $\beta$ parametrizes "the degree of pessimism" in the grid-transfer error bound. Given the grid-transfer parameter $\beta$ and the discretization error. It clear that

$$\mathsf{B}_0(h,h') \leq \mathsf{B}_\beta(h,h') \leq \mathsf{B}_1(h,h'), \quad \forall h, h' \in \mathcal{H}_0, \beta \in [0,1].$$

Moreover, it is seen that $\mathsf{B}_\beta = (1-\beta)\mathsf{B}_0 + \beta\mathsf{B}_1$ for all $\beta \in [0,1]$; it follows that $\mathsf{B}_\beta$ is a metric on $\mathcal{H}_0$ (since it is a convex combination of two metrics on $\mathcal{H}_0$). Therefore $\mathsf{B}_\beta$ is indeed a grid-transfer error bound.

Before we relate the monotonicity properties of $\mathsf{D}$ and $\mathsf{B}_\beta$, we first introduce some terminology.

**Left and Right Monotonicity**

Let $h_0, h_1, h_2, h_3 \in \mathcal{H}_0$. A function $f : \mathcal{H}_0 \times \mathcal{H}_0 \mapsto \mathbf{R}$ is said to be:

1. *right monotone* if $f(h_0, h_1) \leq f(h_0, h_2)$ for all $h_0 \leq h_1 < h_2$.
2. *left monotone* if $f(h_2, h_3) \leq f(h_1, h_3)$ for all $h_1 < h_2 \leq h_3$.
3. *monotone* if $f(h_1, h_2) \leq f(h_0, h_3)$ for all $h_0 \leq h_1 < h_2 \leq h_3$.

[One can consider strict versions of the above definitions by replace the inequalities by strict inequalities.] It is seen that $f$ is (strictly) monotone iff it is both (strictly) left and right monotone.

We now state two lemmas relating the monotonicity of $\mathsf{B}_\beta$ to the monotonicity of $\mathsf{D}$ and the value of $\beta$.

**Lemma 5.4.2** $\mathsf{B}_\beta$ *is a (strictly) right monotone metric iff* $\mathsf{D}$ *is a (strictly) monotone increasing function.*

**Proof** By the definition of $\mathsf{B}_\beta$ [Eq (5.4.10)] $\mathsf{B}_\beta(0, h) = \mathsf{D}(h)$ for all $h \in \mathcal{H}_0$, the (strict) monotonicity of $\mathsf{D}$ follows. To prove the converse, we assume assume that $\mathsf{D}$ is (strictly) monotone increasing. Then again by the definition of $\mathsf{B}_\beta$, we have

$$\begin{aligned}
\mathsf{B}_\beta(h_0, h_1) &= |\mathsf{D}(h_0) - \mathsf{D}(h_1)| + 2\beta[\mathsf{D}(h_0) \wedge \mathsf{D}(h_1)] \\
&= \mathsf{D}(h_1) - \mathsf{D}(h_0) + 2\beta \mathsf{D}(h_0) \\
&\leq \mathsf{D}(h_2) - \mathsf{D}(h_0) + 2\beta[\mathsf{D}(h_0) \wedge \mathsf{D}(h_2)] \\
&= \mathsf{B}_\beta(h_0, h_2)
\end{aligned}$$

as required. (The last inequality is strict if $\mathsf{D}$ is strictly monotone). □

**Lemma 5.4.3** *Let* $\mathsf{D}$ *be a monotone increasing function. There holds,*

1. *If* $\beta \leq 1/2$ *then* $\mathsf{B}_\beta$ *is a left monotone metric.*
2. *If* $\mathsf{D}$ *is strictly monotone, then*

$$\mathsf{B}_\beta \textit{ is strictly left monotone} \quad \textit{iff} \quad \beta < 1/2.$$

**Proof** To prove Part 1, we have, by definition,

$$\begin{aligned}
\mathsf{B}_\beta(h_2, h_3) &= |\mathsf{D}(h_2) - \mathsf{D}(h_3)| + 2\beta[\mathsf{D}(h_2) \wedge \mathsf{D}(h_3)] \\
&= \mathsf{D}(h_3) - (1 - 2\beta)\mathsf{D}(h_2) \\
&\leq \mathsf{D}(h_3) - (1 - 2\beta)\mathsf{D}(h_1) \\
&= \mathsf{B}_\beta(h_1, h_3),
\end{aligned}$$

as required. By doing a case by case analysis on when the last inequality is strict, Part 2 follows. □

## Theorem and Corollary

We now conclude this section with a final theorem and corollary. And to keep the proofs simple we introduce the following simplifying assumption on the delay factor:

**G.5** $a_h^{h'} = 1$ for all $h, h' \in \mathcal{H}_0$.

The theorem and corollary says two things (which we intuitively know). Under the monotonicity assumptions G.1–G.2 and the appropriate monotonicity assumption on the grid-transfer error bound $\mathsf{B}$ the following can be shown: Let $h_0, h_1, h_2 \in \mathcal{H}_0$ and $h_0 \leq h_1 \leq h_2$.

1. Starting at $x_{h_2}^*$, it is "easier" to compute an $\epsilon$-approximation of $x_{h_1}^*$ than to compute an $\epsilon$-approximation of $x_{h_0}^*$.
2. To compute an $\epsilon$-approximation of $x_{h_0}^*$, it is "easier" to start at $x_{h_1}^*$ than to start $x_{h_2}^*$.

**Theorem 5.4.3** *Given an MCCP $\mathcal{P}(\mathcal{T}, \mathcal{H}, \alpha_h, a_h^{h'}, \mathsf{B}(h, h'), \mathsf{C}(h))$ that satisfies Assumptions G.1–G.2, and G.5. For any $h_0, h_1, h_2, h_3 \in \mathcal{H}_0$ such that $h_0 \leq h_1 \leq h_2 \leq h_3$ and for any $\eta \in \mathcal{A}_0$, there exists a $\tilde{\eta} \in \mathcal{A}_0$ such that $\mathsf{C}_{\mathcal{H}}^{\tilde{\eta}} \leq \mathsf{C}_{\mathcal{H}}^{\eta}$ and there holds*

1. *If $\mathsf{B}$ is left monotone, then $\mathsf{p}(x_{h_2}^*, A_{\mathcal{H}}^{\tilde{\eta}} x_{h_3}^*) \leq \mathsf{p}(x_{h_1}^*, A_{\mathcal{H}}^{\eta} x_{h_3}^*)$.*
2. *If $\mathsf{B}$ is right monotone, then $\mathsf{p}(x_{h_0}^*, A_{\mathcal{H}}^{\tilde{\eta}} x_{h_1}^*) \leq \mathsf{p}(x_{h_0}^*, A_{\mathcal{H}}^{\eta} x_{h_2}^*)$.*
3. *If $\mathsf{B}$ is monotone, then $\mathsf{p}(x_{h_1}^*, A_{\mathcal{H}}^{\tilde{\eta}} x_{h_2}^*) \leq \mathsf{p}(x_{h_0}^*, A_{\mathcal{H}}^{\eta} x_{h_3}^*)$.*

**Proof** By Theorem 5.4.1 we can assume that $\eta$ is one-way; therefore, we will assume throughout the proof that $\eta$ is one-way.

We now prove Part 1. The main idea is that we will split $\eta$ into two parts: The first part with grid-level above $h_2$ and the second part is the remainder. The trajectory $\tilde{\eta}$ is formed by keeping the first part but replacing the second part by a singe-grid trajectory on grid-level $h_2$. (We can think of $\tilde{\eta}$ as the result of "clipping" $\eta$ so that it does not go below $h_2$.)

We begin with the trivial cases. First, the result is trivially true if $|\eta| = 0$. Second, if the final grid-level of $\eta$ is $h' \geq h_2$, then $\tilde{\eta} = \eta$ is the required trajectory. This is because $\mathsf{p}(x_{h_2}^*, x_{h'}^*) \leq \mathsf{p}(x_{h_1}^*, x_{h'}^*)$, by left monotonicity assumption of $\mathsf{B}$.

So let $\eta = \eta_1 \cdot \eta_0$, where the final grid-level of $\eta_0$ is greater than $h_2$ and

$$\eta_1 = ((t_i', h_i'), (t_{i-1}', h_{i-1}'), \ldots, (t_1', h_1')),$$

for some $h_1' \leq h_2$. And let $\tilde{\eta} = ((t_s, h_s)) \cdot \eta_0$, where $t_s = \sum_{j=1}^{i} t_j'$. It is clear from the monotonicity of $\mathsf{C}$ that $\mathsf{C}_{\mathcal{H}}^{\tilde{\eta}} \leq \mathsf{C}_{\mathcal{H}}^{\eta}$. It remains to verify that $\tilde{\eta}$ satisfies the second requirement.

Since the final grid-level of $\eta_0$ is greater than $h_2$, it follows (from the left monotonicity of $\mathsf{B}$) that

$$\mathsf{p}(x_{h_2}^*, A_{\mathcal{H}}^{\eta_0} x_{h_3}^*) \leq \mathsf{p}(x_{h_1}^*, A_{\mathcal{H}}^{\eta_0} x_{h_3}^*). \tag{5.4.11}$$

Using the definition of $\mathsf{p}$, Assumptions G.1–G.2, Eq. (5.4.11) we have

$$\begin{aligned}
\mathsf{p}(x^*_{h_2}, A^{\tilde\eta}_{\mathcal{H}} x^*_{h_3}) &= \alpha^{t_s}_{h_s} \mathsf{p}(x^*_{h_2}, A^{\eta_0}_{\mathcal{H}} x^*_{h_3}) \\
&\leq \alpha^{t_s}_{h_s} \mathsf{p}(x^*_{h_1}, A^{\eta_0}_{\mathcal{H}} x^*_{h_3}) \\
&\leq \alpha^{t_s}_{h_s} \left[\mathsf{p}(x^*_{h_1}, x^*_{h'_i}) + \mathsf{p}(x^*_{h'_i}, x^*_{h'_{i-1}}) + \cdots + \mathsf{p}(x^*_{h'_1}, A^{\eta_0}_{\mathcal{H}} x^*_{h_3})\right] \\
&\leq \mathsf{p}(x^*_{h_1}, x^*_{h'_i}) + \alpha^{t'_i}_{h'_i} \left[\mathsf{p}(x^*_{h'_i}, x^*_{h'_{i-1}}) + \alpha^{t'_{i-1}}_{h'_{i-1}} \left[\cdots + \alpha^{t'_1}_{h'_1} \mathsf{p}(x^*_{h'_1}, A^{\eta_0}_{\mathcal{H}} x^*_{h_3})\right]\right] \\
&= \mathsf{p}(x^*_{h_1}, A^{\eta_1 \cdot \eta_0}_{\mathcal{H}} x^*_{h_3})
\end{aligned}$$

as required.

We now prove Part 2. The basic idea is similar to that used in Part 1 but instead of "clipping" $\tilde\eta$ from below, we only use the part of $\eta$ with grid-levels below $h_1$.

As in Part 1, the result is trivially true if $|\eta| = 0$ and if initial grid-level of $\eta$, $h'_s \leq h_1$. In the latter case $\tilde\eta = \eta$ is the required trajectory. This is because by the right monotonicity of $\mathsf{B}$, we have, $\mathsf{p}(x^*_{h'_s}, x^*_{h_1}) \leq \mathsf{p}(x^*_{h'_s}, x^*_{h_2})$, and the result follows from the definition of $\mathsf{p}$.

So let $\eta = \eta_1 \cdot \eta_0$, where

$$\eta_1 = ((t'_j, h'_j), \ldots, (t'_1, h'_1)),$$

$h'_1 < h_1$, $\eta_0 = ((t'_0, h'_0)) \cdot \eta'_0$, and $h'_0 \geq h_1$.

Let $\tilde\eta = \eta_1$. It is clear that $\mathsf{C}^{\tilde\eta}_{\mathcal{H}} \leq \mathsf{C}^{\eta}_{\mathcal{H}}$; it remains to verify that $\tilde\eta$ satisfies the second requirement. First, note that

$$\begin{aligned}
\mathsf{p}(x^*_{h'_1}, x^*_{h_1}) &\leq \mathsf{p}(x^*_{h'_1}, x^*_{h'_0}) \\
&\leq \mathsf{p}(x^*_{h'_1}, x^*_{h'_0}) + \alpha^{t'_0}_{h'_0} \mathsf{p}(x^*_{h'_1}, A^{\eta'_0}_{\mathcal{H}} x^*_{h_2}) \\
&= \mathsf{p}(x^*_{h'_1}, A^{((t'_0,h'_0))\cdot\eta'_0}_{\mathcal{H}} x^*_{h_2}) \\
&= \mathsf{p}(x^*_{h'_1}, A^{\cdot\eta_0}_{\mathcal{H}} x^*_{h_2}).
\end{aligned}$$

Lastly, using the above inequality, we obtain

$$\begin{aligned}
\mathsf{p}(x^*_{h_0}, A^{\tilde\eta}_{\mathcal{H}} x^*_{h_1}) &= \mathsf{p}(x^*_{h_0}, x^*_{h'_j}) + \alpha^{t'_j}_{h'_j} \left[\cdots + \alpha^{t'_1}_{h'_1} \mathsf{p}(x^*_{h'_1}, x^*_{h_1})\right] \\
&\leq \mathsf{p}(x^*_{h_0}, x^*_{h'_j}) + \alpha^{t'_j}_{h'_j} \left[\cdots + \alpha^{t'_1}_{h'_1} \mathsf{p}(x^*_{h'_1}, A^{\eta_0}_{\mathcal{H}} x^*_{h_2})\right] \\
&= \mathsf{p}(x^*_{h_0}, A^{\eta_1\cdot\eta_0}_{\mathcal{H}} x^*_{h_2}) \\
&= \mathsf{p}(x^*_{h_0}, A^{\eta}_{\mathcal{H}} x^*_{h_2}),
\end{aligned}$$

as required.

Part 3 follows easily from Parts 1 and 2.

□

The following corollary is now immediate

**Corollary 5.4.3** *Under the hypotheses of Theorem 5.4.3 and Assumption G.0, there hold*

*1. If* B *is left monotone, then*

$$\mathsf{C}^*(x^*_{h_3}, x^*_{h_2}; \epsilon) \leq \mathsf{C}^*(x^*_{h_3}, x^*_{h_1}; \epsilon), \quad \forall \epsilon > 0.$$

*2. If* B *is right monotone, then*

$$\mathsf{C}^*(x^*_{h_1}, x^*_{h_0}; \epsilon) \leq \mathsf{C}^*(x^*_{h_2}, x^*_{h_0}; \epsilon) \quad \forall \epsilon > 0.$$

*3. If* B *is monotone, then*

$$\mathsf{C}^*(x^*_{h_2}, x^*_{h_1}; \epsilon) \leq \mathsf{C}^*(x^*_{h_3}, x^*_{h_0}; \epsilon) \quad \forall \epsilon > 0.$$

## 5.5 Conclusions

The general framework allows us to understand and analyze the behavior of many relaxation algorithms. In this framework, we only need the following three parameters (as a function of the grid-level): the contraction factor, the discretization error (or the grid-transfer error) bound, and the iteration cost. We have analyzed in detail some specific forms of these parameters in Section 5.2, explored various applications of the framework in Section 5.3, and proved some general results in Section 5.4. We now discuss some observations and make some suggestions for future research.

### 5.5.1 Some Observations

#### Relation to Continuation Methods

In *continuation methods* [see, for example, Wacker (1978)] a "hard" problem is solved by imbedding it into a family of problems, parametrized by $s \in [0, 1]$. The family of problems can be viewed as a homotopy of the original problem, where $s = 0$ corresponds to some "easy" problem and $s = 1$ corresponds to the original problem. The original problem is solved by first obtaining a solution to the easy, $s = 0$, problem, and continuously transforming this solution to that of the solution of the $s = 1$ problem.

Our formulation of the multigrid methods can be viewed as a form of continuation method. We approximate the solution to a "hard", computationally expensive ($h = 0$) problem by first approximation the solution of some "easy", computationally cheap ($h = 1$) problem and using that as the initial estimate for approximating the solution to an "intermediate" (for example, $h = 1/2$) problem. We keep doing that until we obtain the required approximation. The only difference (from the traditional continuation method) is that we use a "discrete" sequence of problems (instead of a continuous homotopy). Moreover, there is a notion of computational cost, so our framework is an example of *computational homotopy*, or "compotopy". We found the optimality principle of dynamic programming to be an effective tool for tackling such problems.

### 5.5.2 Future Research

Some suggestions for future research are as follows.

1. One possible area of application of the abstract framework concerns finite-state relaxation problems, where the state space is large. Many such problems can be viewed as finding the fixed point of a discrete-time contraction process, say $x_h^*$ of $A_h$. For such problems an iteration of $A_h$ can very expensive. If we let $1/h$ denote the number of states, then we may be able to approximate $A_h$ by a family of $\{A_{h'}\}_{h' \in \mathcal{H}}$, where $\mathcal{H} \subset [h, 1]$ and each iteration of $A_{h'}$ cost less and that $\mathsf{d}(x_h^*, x_{h'}^*)$ decreases to 0 as $h' \searrow h$. It remains to find the proper domain for the framework. Also note from the general results in Section 5.4 that the algorithms should proceed from coarse to fine grid-levels.

2. More study is needed to determine if the general framework in this chapter provides a fruitful way of viewing problems in Numerical Analysis, and to determine suitable assumptions on the grid-transfer error bounds and the delay factor. In particular, more study is needed to determine realistic (monotonicity) assumptions on the delay factor for which the results of Section 5.4 hold.

3. In our framework, we assume that the contraction process converges geometrically (that is, at a *linear* rate). One can consider other types of convergence such as *superlinear* or *quadratic* (for example, in Newton's method) and perhaps develop a similar framework for these problems. One possible application of such a framework is in the analysis of inexact Newton Methods for solving finite (or perhaps infinite) dimensional problems.

   It is also interesting to know whether the multigrid principle (which suggests iterating on each grid-level until the successive approximation error is comparable to the discretization error) still applies in such cases. We expect that the results of Section 5.4 [in particular, the optimality of one-way multigrid algorithms] still hold for such a framework. More importantly, we expect the optimality principle of dynamic programming to be applicable.

4. Our formulation of the minimum computational cost problem can be viewed as a one-parameter problem (parametrized by $h$). In general, we can consider two (or more) parameter problems; that is, we consider a family of contraction processes $\{A_{hl}\}$, parametrized by some $h$ and $l$, where the contraction factor, grid-transfer error bound, and the iteration cost are now functions of $h$ and $l$. A possible application of such problem is in general continuous-time, continuous-state-space problems, where one parameter describes the spatial discretization and the other parameter describes the temporal discretization. Again we expect the optimality principle in dynamic programming to be effective for handling such problems. However, since the parametric space is not one-dimensional, finding an optimal trajectory may be harder.

# Chapter 6

# Conclusions

We now give a summary of our research and suggest some general areas for future research.

## 6.1 Summary

We have looked at algorithms and lower bounds for discrete-time stochastic control, as well as, for more general fixed-point problems. We looked at a single-grid and a one-way multigrid algorithm and showed that the multigrid algorithm is an improvement of the single-grid algorithm. We showed the optimality of the one-way multigrid algorithm by proving lower bounds. In the first part of this report, we proved information-based lower bounds; in the second part, we proved algorithmic-based lower bounds.

We have seen that the (asymptotic) optimality of multigrid algorithm is the result of its one-way (coarse to fine) nature; going up and down the grid-levels (for example, in the Full Multigrid V-cycle algorithm) only improves the convergence rate. In the context of discrete-time stochastic control, the multigrid algorithm does not have to go up and down the grid levels because the contraction factor is part of the problem specification. It is unclear whether going up and down the grid-levels, in our context, can give a better contraction factor. This is because discrete-time stochastic control problems do not seem to have any special structure.

A novel feature of our analysis is that we simultaneously consider the dependence on the accuracy and the discount parameters. We see that the latter dependence leads to some deeper results. There are three reasons for considering the dependence on the discount factor (as it increases to one): (i) The closer the discount factor is to one, the more and more the problem looks like an average cost problem. (ii) This dependence is sensitive to certain ergodicity conditions on the problem. (iii) In a continuous-time problem, as the time discretization becomes smaller and smaller the contraction factor of the corresponding discretized problem increases to one.

Another novel feature of our work is the concept of algorithmic-based lower bounds. Currently, these lower bounds have rather restricted interpretations. We expect further research in this area should lead to more refined lower bound results.

In summary, we have presented an in-depth study of a narrow aspect of the connection between control and computation. We believe this area of study is both fundamental and enlightening. It also raises an issue concerning research in control. Before computers were widely available and used in control, it was justifiable to develop abstract control models, prove convergence results, but not address the computational aspects of the model. In the light that the use of computers is becoming prevalent in control, the justification has to be reexamined. Perhaps, one should not only derive rate of convergence, but also consider the computational aspects of the model.

## 6.2 Areas of Future Research

We now suggest some general areas for future research. More specific suggestions have been given in the earlier sections.

1. We have only considered "worst case" lower bounds, since they are the simplest to prove. But they are too pessimistic in practice. One area of research is to prove "average case" lower bounds, which may be more of more practical value.

2. We have only discussed the computational aspects of discrete-time stochastic control. Another area of research is to carry our a similar study for continuous-time stochastic control. But there are numerous technical difficulties in the latter problem, one of which is in finding good discretization results.

3. More study is needed to further develop the abstract framework of Chapter 5. We have only studied one type of convergence—linear convergence. A future area of research is to consider other types of convergence, such as superlinear or quadratic convergence. Furthermore, we have only considered a family of "homogeneous" operators. It would be interesting to prove lower bound results for two or more families of "non-homogeneous" operators that can interact in some non-linear fashion.

# References

[1] Akian, M., J. P. Quadrat, and J. P. Chancelier (1988). Dynamic Programming Complexity and Application, *Proceedings of the 27th IEEE Conference on Decision and Control,* Austin, Texas, December, 1551–1558.

[2] Ash, R. B. (1972). *Measure, Integration, and Functional Analysis.* Academic Press, New York.

[3] Bertsekas, D. P. (1975). Convergence of Discretization Procedures in Dynamic Programming, *IEEE Trans. Automatic Control,* **AC-20**, 415–419.

[4] Bertsekas, D. P. (1987). *Dynamic Programming: Deterministic and Stochastic Models.* Prentice-Hall, Englewood Cliffs, New Jersey.

[5] Bertsekas, D. P. and D. Castanon (1986). Adaptive Aggregation Methods for Discounted Dynamic Programming, *Proceedings of the 25th IEEE Conference on Decision and Control,* Athens, Greece, December, 1840–1845.

[6] Bertsekas, D. P. and S. E. Shreve (1978). *Stochastic Optimal Control: The Discrete Time Case.* Academic Press, New York.

[7] Briggs, B. (1987). *A Multigrid Tutorial.* SIAM, Philadelphia.

[8] Böhmer, K. and H. J. Stetter (1984). *Defect Correction Methods: Theory and Applications.* Springer-Verlag, New York.

[9] Brandt, A. (1986). Multi-level Approaches to Large Scale Problems, *Proceedings of International Congress of Mathematicians,* Berkeley, California, August.

[10] Bryson, A. and Y.-C. Ho (1961). *Applied Optimal Control.* Blaisdel Publishing Company.

[11] Chatelin, F. and W. L. Miranker (1980). Acceleration by Aggregation of Successive Approximation Methods, *Linear Algebra Appl.* **43**, 17–47.

[12] Chow, C.-S. and J. N. Tsitsiklis (1988). An Optimal Multigrid Algorithm for Continuous State Discrete Time Stochastic Control, *27th IEEE Conference on Decision and Control,* Austin, Texas, December, 1908–1912.

[13] Chow, C.-S. and J. N. Tsitsiklis (1989a). The Information-Based Complexity of Dynamic Programming, Technical Report LIDS-P1863, Laboratory for Information and Decision Systems, M.I.T., Cambridge, Massachusetts, April. (To appear in the *J. Complexity,* December, 1989.)

[14] Chow, C.-S. and J. N. Tsitsiklis (1989b). An Optimal Multigrid Algorithm for Discrete-Time Stochastic Control, Technical Report LIDS-P1864, Laboratory for Information and Decision Systems, M.I.T., Cambridge, Massachusetts, April. (To appear in the *IEEE Trans. Automatic Control.*)

[15] Conway, J. (1985). *A Course in Functional Analysis.* Springer-Verlag, New York.

[16] Denardo, E. V. (1967). Contraction Mappings in The Theory Underlying Dynamic Programming, *SIAM Review,* **9**, 165 – 177.

[17] Douglas, C. C. (1984). Multi-grid algorithms with applications to elliptic boundary-value problems, *SIAM J. Numer. Anal.* **21**, 236–254.

[18] Federgruen, A., P. J. Schweitzer, and H. C. Tijms (1978). Contraction Mappings Underlying Undiscounted Markov Decision Problems, *J. Math. Anal. Appl.* **65**, 711–730.

[19] Fleming, W. H. and R. W. Rishel (1975). *Deterministic and Stochastic Optimal Control.* Springer-Verlag, New York.

[20] Fox, B. L. (1971). Finite-State Approximations to Denumerable-State Dynamic Programs, *J. Math. Anal. Appl.* **34**, 665–670.

[21] Fox, B. L. (1973). Discretizing Dynamic Programs, *J. Opt. Theory and Appl.* **11**, 228–234.

[22] Goodman, J. and A. D. Sokal (1986). Multigrid Monte Carlo Methods for Lattice Field Theories, *Physical Review Letters,* **56**, 1015–1018.

[23] Hackbusch, W. (1985). *Multi-Grid Methods and Applications.* Springer-Verlag, New York.

[24] Hajek, B. (1988). Cooling Schedules for Optimal Annealing, *Math. of Oper. Res.* **13**, 311–329.

[25] Hartley, R., L. C. Thomas, and D. J. White (eds.) (1980). *Recent Developments in Markov Decision Processes.* Academic Press, New York.

[26] Haurie, A. and P. L'Ecuyer (1986). Approximation and Bounds in Discrete Event Dynamic Programming, *IEEE Trans. Automatic Control,* **AC-31**, 227–235.

[27] Howard, R. (1960). *Dynamic Programming and Markov Processes.* MIT Press, Cambridge, Massachusetts.

[28] Hinderer, K. (1976). Estimates for Finite-State Dynamic Programs, *J. Math. Anal. and Appl.* **55**, 207–238.

[29] Hernández-Lerma, O. (1989). *Adaptive Markov Control Process.* Springer-Verlag, New York.

[30] Hoppe R. H. W. (1986). Multi-Grid Methods for Hamilton-Jacobi-Bellman Equations, *Numerische Mathematik,* **49**, 239–254.

[31] Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). Optimization by Simulated Annealing, *Science,* **220**, 621-680.

[32] Kolmogorov, A. N. and S. V. Fomin (1970). *Introductory Real Analysis.* Dover Publications.

[33] Langen, H.J. (1981). Convergence of Dynamic Programming Models, *Math. Oper. Res.* **6**, 493–512.

[34] L'Ecuyer, P. (1989). Computing Approximate Solutions to Markov Renewal Programs with Continuous State Space, Report no DIUL-RR 8912, Département d'Informatique, Uni. Laval, Canada.

[35] Lewis, H. R. and C. H. Papadimitriou (1981). *Elements of The Theory of Computation.* Prentice-Hall, Englewood Cliffs, New Jersey.

[36] Mayr, O. (1970). *The origins of feedback control.* MIT Press, Cambridge.

[37] Nemirovsky, A. S. and D. B. Yudin (1979). *Problem Complexity and Method Efficiency Optimization* (translated by E. R. Dawson 1983). John Wiley & Sons, New York.

[38] Nummelin, E. (1984). *General irreducible Markov chains and non-negative operators.* Cambridge University Press, Great Britain.

[39] Papadimitriou, C. H., and J.N. Tsitsiklis (1986). Intractable Problems in Control Theory, *SIAM J. Control and Optimization* **24**, 639–654.

[40] Papadimitriou, C. H., and J.N. Tsitsiklis (1987). The Complexity of Markov Decision Processes, *Math. of Oper. Res.* **12**, 441–450.

[41] Puterman, M. L. (editor) (1978). *Dynamic Programming and Its Applications.* Academic Press, New York.

[42] Puterman, M. L. and S. L. Brumelle (1979). On the Convergence of Policy Iteration in Stationary Dynamic Programming, *Math. of Oper. Res.* **4**, 60–69.

[43] Puterman, M. L. and M. C. Shin (1978). Modified Policy Iteration Algorithms for Discounted Markov Decision Problems, *Management Sci.* **24**, 1127–1137.

[44] Royden, W. (1964). *Principles of Mathematical Analysis* (2nd edition). McGraw-Hill Book Co., Inc., New York.

[45] Schippers, H. (1979). Multigrid Techniques for the Solution of Fredholm Integral Equations of the Second Kind, *Proceedings of a Colloquium on Numerical Treatment of Integral Equations*, T. Riele, (editor), Mathematisch Centrum, Amsterdam.

[46] Seneta, E. (1981). *Non-negative Matrices and Markov Chains* (2nd edition). Springer-Verlag, New York.

[47] Schweitzer, P. J., M. L. Puterman, and K. W. Kindle (1985). Iterative Aggregation-Disaggregation Procedures for Discounted Semi-Markov Reward Processes, *Oper. Res.* **33**, 589–605.

[48] Schweitzer, P. J., (1988). Contraction Mappings Underlying Undiscounted Markov Decision Problems—II, *J. Math. Anal. Appl.* **132**, 154–170.

[49] Terzopoulos, D. (1984). Multigrid Relaxation methods and the analysis of lightness, shading, and flow, *MIT AI Laboratory A.I. Memo No. 803.*, MIT, Cambridge, Massachusetts.

[50] Tsitsiklis, J. N. (1989). Markov Chains with Rare Transitions and Simulated Annealing, *Math. Oper. Res.* **14**, 70–90.

[51] Traub, J. F., G. W. Wasilkowski, and H. Wozniakowski (1988). *Information-Based Complexity.* Academic Press, New York.

[52] Wacker, H. (editor) (1978). *Continuation Methods.* Academic Press, New York.

[53] Werschulz, A. G. (1985). What is the Complexity of the Fredholm Problem of the Second Kind? *J. Integral Eq.* **9**, 213-241.

[54] White, D. (1980). Finite State Approximations for Denumerable State Infinite Horizon Discounted Markov Decision Processes, *J. Math Anal. Appl.* **74**, 292–295.

[55] White, D. (1982). Finite State Approximations for Denumerable State Infinite Horizon Discounted Markov Decision Processes with Unbounded Rewards, *J. Math. Anal. Appl.* **86**.

[56] Whitt, W. (1978). Approximations of Dynamic Programs I, *Math. Oper. Res.* **3**, 231–243.

[57] Whitt, W. (1979). Approximations of Dynamic Programs II, *Math. Oper. Res.* **4**, 179–185.