1

Introduction

Contents

1.1 STRUCTURE OF DYNAMIC PROGRAMMING PROBLEMS

Dynamic programming (DP for short) is the principal method for analysis of a large and diverse class of sequential decision problems. Examples are deterministic and stochastic optimal control problems with a continuous state space, Markov and semi-Markov decision problems with a discrete state space, minimax problems, and sequential zero-sum games. While the nature of these problems may vary widely, their underlying structures turn out to be very similar. In all cases there is an underlying mapping that depends on an associated controlled dynamic system and corresponding cost per stage. This mapping, the DP operator, provides a compact "mathematical signature" of the problem. It defines the cost function of policies and the optimal cost function, and it provides a convenient shorthand notation for algorithmic description and analysis.

More importantly, the structure of the DP operator defines the mathematical character of the associated problem. The purpose of this book is to provide an analysis of this structure, centering on two fundamental properties: *monotonicity* and (weighted sup-norm) *contraction*. It turns out that the nature of the analytical and algorithmic DP theory is determined primarily by the presence or absence of one or both of these two properties, and the rest of the problem's structure is largely inconsequential.

A Deterministic Optimal Control Example

To illustrate our viewpoint, let us consider a discrete-time deterministic optimal control problem described by a system equation

$$x_{k+1} = f(x_k, u_k), \qquad k = 0, 1, \ldots. \tag{1.1}$$

Here x_k is the state of the system taking values in a set X (the state space), and u_k is the control taking values in a set U (the control space). † At stage k, there is a cost

$$\alpha^k g(x_k, u_k)$$

incurred when u_k is applied at state x_k, where α is a scalar in $(0, 1]$ that has the interpretation of a discount factor when $\alpha < 1$. The controls are chosen as a function of the current state, subject to a constraint that depends on that state. In particular, at state x the control is constrained to take values in a given set $U(x) \subset U$. Thus we are interested in optimization over the set of (nonstationary) policies

$$\Pi = \big\{\{\mu_0, \mu_1, \ldots\} \mid \mu_k \in \mathcal{M},\ k = 0, 1, \ldots\big\},$$

† Our discussion of this section is somewhat informal, without strict adherence to mathematical notation and rigor. We will introduce a rigorous mathematical framework later.

where $\mathcal{M}$ is the set of functions $\mu : X \mapsto U$ defined by

$$\mathcal{M} = \big\{\mu \mid \mu(x) \in U(x),\ \forall\ x \in X\big\}.$$

The total cost of a policy $\pi = \{\mu_0, \mu_1, \ldots\}$ over an infinite number of stages (an infinite horizon) and starting at an initial state x_0 is the limit superior of the N-step costs

$$J_\pi(x_0) = \limsup_{N\to\infty} \sum_{k=0}^{N-1} \alpha^k g\big(x_k, \mu_k(x_k)\big), \tag{1.2}$$

where the state sequence $\{x_k\}$ is generated by the deterministic system (1.1) under the policy π:

$$x_{k+1} = f\big(x_k, \mu_k(x_k)\big), \quad k = 0, 1, \ldots.$$

(We use limit superior rather than limit to cover the case where the limit does not exist.) The optimal cost function is

$$J^*(x) = \inf_{\pi\in\Pi} J_\pi(x), \qquad x \in X.$$

For any policy $\pi = \{\mu_0, \mu_1, \ldots\}$, consider the policy $\pi_1 = \{\mu_1, \mu_2, \ldots\}$ and write by using Eq. (1.2),

$$J_\pi(x) = g\big(x, \mu_0(x)\big) + \alpha J_{\pi_1}\big(f(x, \mu_0(x))\big).$$

We have for all $x \in X$

$$\begin{aligned} J^*(x) &= \inf_{\pi=\{\mu_0,\pi_1\}\in\Pi} \Big\{g\big(x, \mu_0(x)\big) + \alpha J_{\pi_1}\big(f(x, \mu_0(x))\big)\Big\} \\ &= \inf_{\mu_0\in\mathcal{M}} \Big\{g\big(x, \mu_0(x)\big) + \alpha \inf_{\pi_1\in\Pi} J_{\pi_1}\big(f(x, \mu_0(x))\big)\Big\} \\ &= \inf_{\mu_0\in\mathcal{M}} \Big\{g\big(x, \mu_0(x)\big) + \alpha J^*\big(f(x, \mu_0(x))\big)\Big\}. \end{aligned}$$

The minimization over $\mu_0 \in \mathcal{M}$ can be written as minimization over all $u \in U(x)$, so we can write the preceding equation as

$$J^*(x) = \inf_{u\in U(x)} \Big\{g(x, u) + \alpha J^*\big(f(x, u)\big)\Big\}, \qquad \forall\ x \in X. \tag{1.3}$$

This equation is an example of *Bellman's equation*, which plays a central role in DP analysis and algorithms. If it can be solved for J^*, an optimal stationary policy $\{\mu^*, \mu^*, \ldots\}$ may typically be obtained by minimization of the right-hand side for each x, i.e.,

$$\mu^*(x) \in \arg\min_{u\in U(x)} \Big\{g(x, u) + \alpha J^*\big(f(x, u)\big)\Big\}, \qquad \forall\ x \in X. \tag{1.4}$$

We now note that both Eqs. (1.3) and (1.4) can be stated in terms of the expression

$$H(x,u,J) = g(x,u) + \alpha J\big(f(x,u)\big), \qquad x \in X,\ u \in U(x).$$

Defining

$$(T_\mu J)(x) = H\big(x,\mu(x),J\big), \qquad x \in X,$$

and

$$(TJ)(x) = \inf_{u\in U(x)} H(x,u,J) = \inf_{\mu\in\mathcal{M}} (T_\mu J)(x), \qquad x \in X,$$

we see that Bellman's equation (1.3) can be written compactly as

$$J^* = TJ^*,$$

i.e., J^* is the fixed point of T, viewed as a mapping from the set of functions on X into itself. Moreover, it can be similarly seen that J_μ, the cost function of the stationary policy $\{\mu,\mu,\ldots\}$, is a fixed point of T_μ. In addition, the optimality condition (1.4) can be stated compactly as

$$T_{\mu^*} J^* = TJ^*.$$

We will see later that additional properties, as well as a variety of algorithms for finding J^* can be stated and analyzed using the mappings T and T_μ.

The mappings T_μ can also be used in the context of DP problems with a finite number of stages (a finite horizon). In particular, for a given policy $\pi = \{\mu_0,\mu_1,\ldots\}$ and a terminal cost $\alpha^N \bar{J}(x_N)$ for the state x_N at the end of N stages, consider the N-stage cost function

$$J_{\pi,N}(x_0) = \alpha^N \bar{J}(x_N) + \sum_{k=0}^{N-1} \alpha^k g\big(x_k,\mu_k(x_k)\big). \tag{1.5}$$

Then it can be verified by induction that for all initial states x_0, we have

$$J_{\pi,N}(x_0) = (T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} \bar{J})(x_0). \tag{1.6}$$

Here $T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}}$ is the composition of the mappings $T_{\mu_0}, T_{\mu_1}, \ldots T_{\mu_{N-1}}$, i.e., for all J,

$$(T_{\mu_0} T_{\mu_1} J)(x) = \big(T_{\mu_0}(T_{\mu_1} J)\big)(x), \qquad x \in X,$$

and more generally

$$(T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} J)(x) = \big(T_{\mu_0}(T_{\mu_1}(\cdots(T_{\mu_{N-1}} J)))\big)(x), \qquad x \in X,$$

(our notational conventions are summarized in Appendix A). Thus the finite horizon cost functions $J_{\pi,N}$ of π can be defined in terms of the mappings T_μ [cf. Eq. (1.6)], and so can the infinite horizon cost function J_π:

$$J_\pi(x) = \limsup_{N\to\infty} (T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} \bar{J})(x), \qquad x \in X, \tag{1.7}$$

where $\bar{J}$ is the zero function, $\bar{J}(x) = 0$ for all $x \in X$.

Connection with Fixed Point Methodology

The Bellman equation (1.3) and the optimality condition (1.4), stated in terms of the mappings T_μ and T, highlight a central theme of this book, which is that DP theory is intimately connected with the theory of abstract mappings and their fixed points. Analogs of the Bellman equation, $J^* = TJ^*$, optimality conditions, and other results and computational methods hold for a great variety of DP models, and can be stated compactly as described above in terms of the corresponding mappings T_μ and T. The gain from this abstraction is greater generality and mathematical insight, as well as a more unified, economical, and streamlined analysis.

1.2 ABSTRACT DYNAMIC PROGRAMMING MODELS

In this section we formally introduce and illustrate with examples an abstract DP model, which embodies the ideas just discussed.

1.2.1 Problem Formulation

Let X and U be two sets, which we loosely refer to as a set of "states" and a set of "controls," respectively. For each $x \in X$, let $U(x) \subset U$ be a nonempty subset of controls that are feasible at state x. We denote by $\mathcal{M}$ the set of all functions $\mu : X \mapsto U$ with $\mu(x) \in U(x)$, for all $x \in X$.

In analogy with DP, we refer to sequences $\pi = \{\mu_0, \mu_1, \ldots\}$, with $\mu_k \in \mathcal{M}$ for all k, as "nonstationary policies," and we refer to a sequence $\{\mu, \mu, \ldots\}$, with $\mu \in \mathcal{M}$, as a "stationary policy." In our development, stationary policies will play a dominant role, and with slight abuse of terminology, we will also refer to any $\mu \in \mathcal{M}$ as a "policy" when confusion cannot arise.

Let $\mathcal{R}(X)$ be the set of real-valued functions $J : X \mapsto \Re$, and let $H : X \times U \times \mathcal{R}(X) \mapsto \Re$ be a given mapping.† For each policy $\mu \in \mathcal{M}$, we consider the mapping $T_\mu : \mathcal{R}(X) \mapsto \mathcal{R}(X)$ defined by

$$(T_\mu J)(x) = H\big(x, \mu(x), J\big), \qquad \forall\ x \in X,\ J \in \mathcal{R}(X),$$

and we also consider the mapping T defined by‡

$$(TJ)(x) = \inf_{u \in U(x)} H(x, u, J), \qquad \forall\ x \in X,\ J \in \mathcal{R}(X).$$

† Our notation and mathematical conventions are outlined in Appendix A. In particular, we denote by $\Re$ the set of real numbers, and by $\Re^n$ the space of n-dimensional vectors with real components.

‡ We assume that H, $T_\mu J$, and TJ are real-valued for $J \in \mathcal{R}(X)$ in the present chapter and in Chapter 2. In Chapters 3 and 4 we will allow $H(x, u, J)$, and hence also $(T_\mu J)(x)$ and $(TJ)(x)$, to take the values ∞ and $-\infty$.

We will generally refer to T and T_μ as the (abstract) *DP mappings* or *DP operators* or *Bellman operators* (the latter name is common in the artificial intelligence and reinforcement learning literature).

Similar to the deterministic optimal control problem of the preceding section, the mappings T_μ and T serve to define a multistage optimization problem and a DP-like methodology for its solution. In particular, for some function $\bar{J} \in \mathcal{R}(X)$, and nonstationary policy $\pi = \{\mu_0, \mu_1, \ldots\}$, we define for each integer $N \geq 1$ the functions

$$J_{\pi,N}(x) = (T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} \bar{J})(x), \qquad x \in X,$$

where $T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}}$ denotes the composition of the mappings T_{μ_0}, T_{μ_1}, $\ldots$, $T_{\mu_{N-1}}$, i.e.,

$$T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} J = \big(T_{\mu_0}(T_{\mu_1}(\cdots(T_{\mu_{N-2}}(T_{\mu_{N-1}} J)))\cdots)\big), \quad J \in \mathcal{R}(X).$$

We view $J_{\pi,N}$ as the "N-stage cost function" of π [cf. Eq. (1.5)]. Consider also the function

$$J_\pi(x) = \limsup_{N\to\infty} J_{\pi,N}(x) = \limsup_{N\to\infty} (T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} \bar{J})(x), \qquad x \in X,$$

which we view as the "infinite horizon cost function" of π [cf. Eq. (1.7); we use lim sup for generality, since we are not assured that the limit exists]. We want to minimize J_π over π, i.e., to find

$$J^*(x) = \inf_\pi J_\pi(x), \qquad x \in X,$$

and a policy π^* that attains the infimum, if one exists.

The key connection with fixed point methodology is that J^* "typically" (under mild assumptions) can be shown to satisfy

$$J^*(x) = \inf_{u\in U(x)} H(x, u, J^*), \qquad \forall\ x \in X,$$

i.e., it is a fixed point of T. We refer to this as *Bellman's equation* [cf. Eq. (1.3)]. Another fact is that if an optimal policy π^* exists, it "typically" can be selected to be stationary, $\pi^* = \{\mu^*, \mu^*, \ldots\}$, with $\mu^* \in \mathcal{M}$ satisfying an optimality condition, such as for example

$$(T_{\mu^*} J^*)(x) = (T J^*)(x), \qquad x \in X,$$

[cf. Eq. (1.4)]. Several other results of an analytical or algorithmic nature also hold under appropriate conditions, which will be discussed in detail later.

However, Bellman's equation and other related results may not hold without T_μ and T having some special structural properties. Prominent among these are a monotonicity assumption that typically holds in DP problems, and a contraction assumption that holds for some important classes of problems. We describe these assumptions next.

1.2.2 Monotonicity and Contraction Properties

Let us now formalize the monotonicity and contraction assumptions. We will require that both of these assumptions hold for most of the next chapter, and we will gradually relax the contraction assumption in Chapters 3 and 4. Recall also our assumption that T_μ and T map $\mathcal{R}(X)$ (the space of real-valued functions over X) into $\mathcal{R}(X)$. In Chapters 3 and 4 we will relax this assumption as well.

Assumption 1.2.1: (Monotonicity) If $J, J' \in \mathcal{R}(X)$ and $J \leq J'$, then

$$H(x,u,J) \leq H(x,u,J'), \qquad \forall\ x \in X,\ u \in U(x).$$

Note that by taking infimum over $u \in U(x)$, we have

$$J(x) \leq J'(x),\ \ \forall\ x \in X \quad \Rightarrow \quad \inf_{u \in U(x)} H(x,u,J) \leq \inf_{u \in U(x)} H(x,u,J'),\ \ \forall\ x \in X,$$

or equivalently, †

$$J \leq J' \qquad \Rightarrow \qquad TJ \leq TJ'.$$

Another way to arrive at this relation, is to note that the monotonicity assumption is equivalent to

$$J \leq J' \quad \Rightarrow \quad T_\mu J \leq T_\mu J', \qquad \forall\ \mu \in \mathcal{M},$$

and to use the simple but important fact

$$\inf_{u \in U(x)} H(x,u,J) = \inf_{\mu \in \mathcal{M}} (T_\mu J)(x), \qquad \forall\ x \in X,\ J \in \mathcal{R}(X),$$

i.e., for a fixed $x \in X$, *infimum over u is equivalent to infimum over μ*, which holds because of the definition $\mathcal{M} = \big\{\mu \mid \mu(x) \in U(x),\ \forall\ x \in X\big\}$, so that $\mathcal{M}$ can be viewed as the Cartesian product $\Pi_{x \in X} U(x)$. We will be writing this relation as $TJ = \inf_{\mu \in \mathcal{M}} T_\mu J$.

For the contraction assumption, we introduce a function $v : X \mapsto \Re$ with

$$v(x) > 0, \qquad \forall\ x \in X.$$

Let us denote by $\mathcal{B}(X)$ the space of real-valued functions J on X such that $J(x)/v(x)$ is bounded as x ranges over X, and consider the weighted sup-norm

$$\|J\| = \sup_{x \in X} \frac{\big|J(x)\big|}{v(x)}$$

† Unless otherwise stated, in this book, inequalities involving functions, minima and infima of a collection of functions, and limits of function sequences are meant to be pointwise; see Appendix A for our notational conventions.

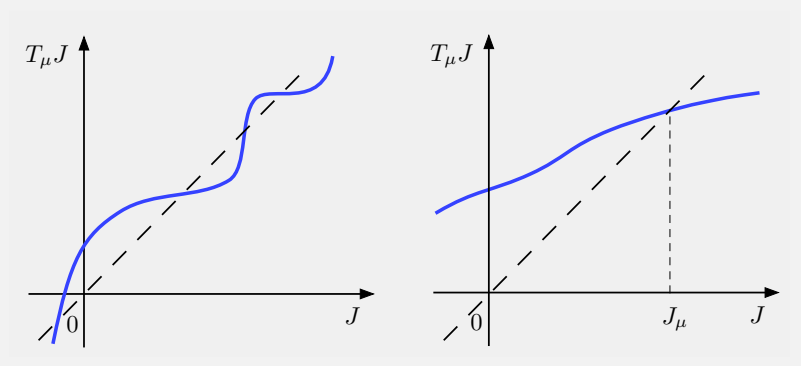

Figure 1.2.1. Illustration of the monotonicity and the contraction assumptions in one dimension. The mapping T_μ on the left is monotone but is not a contraction. The mapping T_μ on the right is both monotone and a contraction. It has a unique fixed point at J_μ.

on $\mathcal{B}(X)$. The properties of $\mathcal{B}(X)$ and some of the associated fixed point theory are discussed in Appendix B. In particular, as shown there, $\mathcal{B}(X)$ is a complete normed space, so any mapping from $\mathcal{B}(X)$ to $\mathcal{B}(X)$ that is a contraction or an m-stage contraction for some integer $m > 1$, with respect to $\|\cdot\|$, has a unique fixed point (cf. Props. B.1 and B.2).

Assumption 1.2.2: (Contraction) For all $J \in \mathcal{B}(X)$ and $\mu \in \mathcal{M}$, the functions $T_\mu J$ and TJ belong to $\mathcal{B}(X)$. Furthermore, for some $\alpha \in (0,1)$, we have

$$\|T_\mu J - T_\mu J'\| \leq \alpha \|J - J'\|, \qquad \forall\ J, J' \in \mathcal{B}(X),\ \mu \in \mathcal{M}. \tag{1.8}$$

Figure 1.2.1 illustrates the monotonicity and the contraction assumptions. It can be shown that the contraction condition (1.8) implies that

$$\|TJ - TJ'\| \leq \alpha \|J - J'\|, \qquad \forall\ J, J' \in \mathcal{B}(X), \tag{1.9}$$

so that T is also a contraction with modulus α. To see this we use Eq. (1.8) to write

$$(T_\mu J)(x) \leq (T_\mu J')(x) + \alpha \|J - J'\|\, v(x), \qquad \forall\ x \in X,$$

from which, by taking infimum of both sides over $\mu \in \mathcal{M}$, we have

$$\frac{(TJ)(x) - (TJ')(x)}{v(x)} \leq \alpha \|J - J'\|, \qquad \forall\ x \in X.$$

Reversing the roles of J and J', we also have

$$\frac{(TJ')(x) - (TJ)(x)}{v(x)} \leq \alpha \|J - J'\|, \qquad \forall\ x \in X,$$

and combining the preceding two relations, and taking the supremum of the left side over $x \in X$, we obtain Eq. (1.9).

Nearly all mappings related to DP satisfy the monotonicity assumption, and many important ones satisfy the weighted sup-norm contraction assumption as well. When both assumptions hold, the most powerful analytical and computational results can be obtained, as we will show in Chapter 2. These are:

(a) Bellman's equation has a unique solution, i.e., T and T_μ have unique fixed points, which are the optimal cost function J^* and the cost functions J_μ of the stationary policies $\{\mu, \mu, \ldots\}$, respectively [cf. Eq. (1.3)].

(b) A stationary policy $\{\mu^*, \mu^*, \ldots\}$ is optimal if and only if

$$T_{\mu^*} J^* = TJ^*,$$

[cf. Eq. (1.4)].

(c) J^* and J_μ can be computed by the *value iteration* method,

$$J^* = \lim_{k \to \infty} T^k J, \qquad J_\mu = \lim_{k \to \infty} T_\mu^k J,$$

starting with any $J \in \mathcal{B}(X)$.

(d) J^* can be computed by the *policy iteration* method, whereby we generate a sequence of stationary policies via

$$T_{\mu^{k+1}} J_{\mu^k} = TJ_{\mu^k},$$

starting from some initial policy μ^0 [here J_{μ^k} is obtained as the fixed point of T_{μ^k} by several possible methods, including value iteration as in (c) above].

These are the most favorable types of results one can hope for in the DP context, and they are supplemented by a host of other results, involving approximate and/or asynchronous implementations of the value and policy iteration methods, and other related methods that combine features of both. As the contraction property is relaxed and is replaced by various weaker assumptions, some of the preceding results may hold in weaker form. For example J^* turns out to be a solution of Bellman's equation in most of the models to be discussed, but it may not be the unique solution. The interplay between the monotonicity and contraction-like properties, and the associated results of the form (a)-(d) described above is a recurring analytical theme in this book.

1.2.3 Some Examples

In what follows in this section, we describe a few special cases, which indicate the connections of appropriate forms of the mapping H with the most popular total cost DP models. In all these models the monotonicity Assumption 1.2.1 (or some closely related version) holds, but the contraction Assumption 1.2.2 may not hold, as we will indicate later. Our descriptions are by necessity brief, and the reader is referred to the relevant textbook literature for more detailed discussion.

Example 1.2.1 (Stochastic Optimal Control - Markovian Decision Problems)

Consider the stationary discrete-time dynamic system

$$x_{k+1} = f(x_k, u_k, w_k), \qquad k = 0, 1, \ldots, \tag{1.10}$$

where for all k, the state x_k is an element of a space X, the control u_k is an element of a space U, and w_k is a random "disturbance," an element of a space W. We consider problems with infinite state and control spaces, as well as problems with discrete (finite or countable) state space (in which case the underlying system is a Markov chain). However, for technical reasons that relate to measure-theoretic issues, *we assume that W is a countable set*.

The control u_k is constrained to take values in a given nonempty subset $U(x_k)$ of U, which depends on the current state x_k [$u_k \in U(x_k)$, for all $x_k \in X$]. The random disturbances w_k, $k = 0, 1, \ldots$, are characterized by probability distributions $P(\cdot \mid x_k, u_k)$ that are identical for all k, where $P(w_k \mid x_k, u_k)$ is the probability of occurrence of w_k, when the current state and control are x_k and u_k, respectively. Thus the probability of w_k may depend explicitly on x_k and u_k, but not on values of prior disturbances $w_{k-1}, \ldots, w_0$.

Given an initial state x_0, we want to find a policy $\pi = \{\mu_0, \mu_1, \ldots\}$, where $\mu_k : X \mapsto U$, $\mu_k(x_k) \in U(x_k)$, for all $x_k \in X$, $k = 0, 1, \ldots$, that minimizes the cost function

$$J_\pi(x_0) = \limsup_{N \to \infty} \mathop{E}_{\substack{w_k \\ k=0,1,\ldots}} \left\{ \sum_{k=0}^{N-1} \alpha^k g\big(x_k, \mu_k(x_k), w_k\big) \right\}, \tag{1.11}$$

subject to the system equation constraint

$$x_{k+1} = f\big(x_k, \mu_k(x_k), w_k\big), \qquad k = 0, 1, \ldots.$$

This is a classical problem, which is discussed extensively in various sources, including the author's text [Ber12a]. It is usually referred to as the *stochastic optimal control problem* or the *Markovian Decision Problem* (MDP for short).

Note that the expected value of the N-stage cost of π,

$$\mathop{E}_{\substack{w_k \\ k=0,1,\ldots}} \left\{ \sum_{k=0}^{N-1} \alpha^k g\big(x_k, \mu_k(x_k), w_k\big) \right\},$$

is defined as a (possibly countably infinite) sum, since the disturbances w_k, $k = 0, 1, \ldots$, take values in a countable set. Indeed, the reader may verify that all the subsequent mathematical expressions that involve an expected value can be written as summations over a finite or a countable set, so they make sense without resort to measure-theoretic integration concepts. †

In what follows we will often impose appropriate assumptions on the cost per stage g and the scalar α, which guarantee that the infinite horizon cost $J_\pi(x_0)$ is defined as a limit (rather than as a lim sup):

$$J_\pi(x_0) = \lim_{N\to\infty} \mathop{E}_{\substack{w_k \\ k=0,1,\ldots}} \left\{ \sum_{k=0}^{N-1} \alpha^k g\big(x_k, \mu_k(x_k), w_k\big) \right\}.$$

In particular, it can be shown that the limit exists if $\alpha < 1$ and the expected value of $|g|$ is uniformly bounded, i.e., for some $B > 0$,

$$E\big\{\big|g(x,u,w)\big|\big\} \le B, \qquad \forall\ x \in X,\ u \in U(x). \tag{1.12}$$

In this case, we obtain the classical discounted infinite horizon DP problem, which generally has the most favorable structure of all infinite horizon stochastic DP models (see [Ber12a], Chapters 1 and 2).

To make the connection with abstract DP, let us define

$$H(x,u,J) = E\big\{g(x,u,w) + \alpha J\big(f(x,u,w)\big)\big\},$$

so that

$$(T_\mu J)(x) = E\big\{g\big(x,\mu(x),w\big) + \alpha J\big(f(x,\mu(x),w)\big)\big\},$$

and

$$(TJ)(x) = \inf_{u\in U(x)} E\big\{g(x,u,w) + \alpha J\big(f(x,u,w)\big)\big\}.$$

Similar to the deterministic optimal control problem of Section 1.1, the N-stage cost of π, can be expressed in terms of T_μ:

$$(T_{\mu_0}\cdots T_{\mu_{N-1}}\bar{J})(x_0) = \mathop{E}_{\substack{w_k \\ k=0,1,\ldots}} \left\{ \sum_{k=0}^{N-1} \alpha^k g\big(x_k, \mu_k(x_k), w_k\big) \right\},$$

† As noted in Appendix A, the formula for the expected value of a random variable w defined over a space Ω is

$$E\{w\} = E\{w^+\} + E\{w^-\},$$

where w^+ and w^- are the positive and negative parts of w,

$$w^+(\omega) = \max\big\{0, w(\omega)\big\}, \qquad w^-(\omega) = \min\big\{0, w(\omega)\big\}, \qquad \forall\ \omega \in \Omega.$$

In this way, taking also into account the rule $\infty - \infty = \infty$ (see Appendix A), $E\{w\}$ is well-defined as an extended real number if Ω is finite or countably infinite.

where $\bar{J}$ is the zero function, $\bar{J}(x) = 0$ for all $x \in X$. The same is true for the infinite-stage cost [cf. Eq. (1.11)]:

$$J_\pi(x_0) = \limsup_{N\to\infty} (T_{\mu_0} \cdots T_{\mu_{N-1}} \bar{J})(x_0).$$

It can be seen that the mappings T_μ and T are monotone, and it is well-known that if $\alpha < 1$ and the boundedness condition (1.12) holds, they are contractive as well (under the unweighted sup-norm); see e.g., [Ber12a], Chapter 1. In this case, the model has the powerful analytical and algorithmic properties (a)-(d) mentioned at the end of the preceding subsection. In particular, the optimal cost function J^* [i.e., $J^*(x) = \inf_\pi J_\pi(x)$ for all $x \in X$] can be shown to be the unique solution of the fixed point equation $J^* = TJ^*$, also known as Bellman's equation, which has the form

$$J^*(x) = \inf_{u\in U(x)} E\Big\{g(x,u,w) + \alpha J^*\big(f(x,u,w)\big)\Big\}, \qquad x \in X,$$

and parallels the one given for deterministic optimal control problems [cf. Eq. (1.3)].

These properties can be expressed and analyzed in an abstract setting by using just the mappings T_μ and T, both when T_μ and T are contractive (see Chapter 2), and when they are only monotone and not contractive while either $g \geq 0$ or $g \leq 0$ (see Chapter 4). Moreover, under some conditions, it is possible to analyze these properties in cases where T_μ is contractive for some but not all μ (see Chapter 3, and Section 4.4).

Example 1.2.2 (Finite-State Discounted Markovian Decision Problems)

In the special case of the preceding example where the number of states is finite, the system equation (1.10) may be defined in terms of the transition probabilities

$$p_{xy}(u) = \text{Prob}\big(y = f(x,u,w) \mid x\big), \qquad x, y \in X,\ u \in U(x),$$

so H takes the form

$$H(x,u,J) = \sum_{y\in X} p_{xy}(u)\big(g(x,u,y) + \alpha J(y)\big).$$

When $\alpha < 1$ and the boundedness condition

$$\big|g(x,u,y)\big| \leq B, \qquad \forall\ x, y \in X,\ u \in U(x),$$

[cf. Eq. (1.12)] holds (or more simply, when U is a finite set), the mappings T_μ and T are contraction mappings with respect to the standard (unweighted) sup-norm. This is a classical model, referred to as *discounted finite-state MDP*, which has a favorable theory and has found extensive applications (cf. [Ber12a], Chapters 1 and 2). The model is additionally important, because it is often used for computational solution of continuous state space problems via discretization.

Example 1.2.3 (Discounted Semi-Markov Problems)

With x, y, and u as in Example 1.2.2, consider a mapping of the form

$$H(x,u,J) = G(x,u) + \sum_{y \in X} m_{xy}(u) J(y),$$

where G is some function representing expected cost per stage, and $m_{xy}(u)$ are nonnegative scalars with

$$\sum_{y \in X} m_{xy}(u) < 1, \qquad \forall\, x \in X,\, u \in U(x).$$

The equation $J^* = TJ^*$ is Bellman's equation for a finite-state continuous-time semi-Markov decision problem, after it is converted into an equivalent discrete-time problem (cf. [Ber12a], Section 1.4). Again, the mappings T_μ and T are monotone and can be shown to be contraction mappings with respect to the unweighted sup-norm.

Example 1.2.4 (Discounted Zero-Sum Dynamic Games)

Let us consider a zero-sum game analog of the finite-state MDP Example 1.2.2. Here there are two players that choose actions at each stage: the first (called the *minimizer*) may choose a move i out of n moves and the second (called the *maximizer*) may choose a move j out of m moves. Then the minimizer gives a specified amount a_{ij} to the maximizer, called a *payoff*. The minimizer wishes to minimize a_{ij}, and the maximizer wishes to maximize a_{ij}.

The players use mixed strategies, whereby the minimizer selects a probability distribution $u = (u_1, \ldots, u_n)$ over his n possible moves and the maximizer selects a probability distribution $v = (v_1, \ldots, v_m)$ over his m possible moves. Thus the probability of selecting i and j is $u_i v_j$, and the expected payoff for this stage is $\sum_{i,j} a_{ij} u_i v_j$ or $u'Av$, where A is the $n \times m$ matrix with components a_{ij}.

In a single-stage version of the game, the minimizer must minimize $\max_{v \in V} u'Av$ and the maximizer must maximize $\min_{u \in U} u'Av$, where U and V are the sets of probability distributions over $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$, respectively. A fundamental result (which will not be proved here) is that these two values are equal:

$$\min_{u \in U} \max_{v \in V} u'Av = \max_{v \in V} \min_{u \in U} u'Av. \tag{1.13}$$

Let us consider the situation where a separate game of the type just described is played at each stage. The game played at a given stage is represented by a "state" x that takes values in a finite set X. The state evolves according to transition probabilities $q_{xy}(i,j)$ where i and j are the moves selected by the minimizer and the maximizer, respectively (here y represents

the next game to be played after moves i and j are chosen at the game represented by x). When the state is x, under $u \in U$ and $v \in V$, the one-stage expected payoff is $u'A(x)v$, where $A(x)$ is the $n \times m$ payoff matrix, and the state transition probabilities are

$$p_{xy}(u,v) = \sum_{i=1}^{n} \sum_{j=1}^{m} u_i v_j q_{xy}(i,j) = u' Q_{xy} v,$$

where Q_{xy} is the $n \times m$ matrix that has components $q_{xy}(i,j)$. Payoffs are discounted by $\alpha \in (0,1)$, and the objectives of the minimizer and maximizer, roughly speaking, are to minimize and to maximize the total discounted expected payoff. This requires selections of u and v to strike a balance between obtaining favorable current stage payoffs and playing favorable games in future stages.

We now introduce an abstract DP framework related to the sequential move selection process just described. We consider the mapping G given by

$$\begin{aligned} G(x,u,v,J) &= u'A(x)v + \alpha \sum_{y \in X} p_{xy}(u,v) J(y) \\ &= u' \left(A(x) + \alpha \sum_{y \in X} Q_{xy} J(y) \right) v, \end{aligned} \tag{1.14}$$

where $\alpha \in (0,1)$ is discount factor, and the mapping H given by

$$H(x,u,J) = \max_{v \in V} G(x,u,v,J).$$

The corresponding mappings T_μ and T are

$$(T_\mu J)(x) = \max_{v \in V} G\big(x, \mu(x), v, J\big), \qquad x \in X,$$

and

$$(TJ)(x) = \min_{u \in U} \max_{v \in V} G(x,u,v,J).$$

It can be shown that T_μ and T are monotone and (unweighted) sup-norm contractions. Moreover, the unique fixed point J^* of T satisfies

$$J^*(x) = \min_{u \in U} \max_{v \in V} G(x,u,v,J^*), \qquad \forall\ x \in X,$$

(see [Ber12a], Section 1.6.2).

We now note that since

$$A(x) + \alpha \sum_{y \in X} Q_{xy} J(y)$$

[cf. Eq. (1.14)] is a matrix that is independent of u and v, we may view $J^*(x)$ as the value of a static game (which depends on the state x). In particular, from the fundamental minimax equality (1.13), we have

$$\min_{u \in U} \max_{v \in V} G(x,u,v,J^*) = \max_{v \in V} \min_{u \in U} G(x,u,v,J^*), \qquad \forall\ x \in X.$$

This implies that J^* is also the unique fixed point of the mapping

$$(\overline{T}J)(x) = \max_{v \in V} \overline{H}(x, v, J),$$

where

$$\overline{H}(x, v, J) = \min_{u \in U} G(x, u, v, J),$$

i.e., J^* is the fixed point regardless of the order in which minimizer and maximizer select mixed strategies at each stage.

In the preceding development, we have introduced J^* as the unique fixed point of the mappings T and $\overline{T}$. However, J^* also has an interpretation in game theoretic terms. In particular, it can be shown that $J^*(x)$ is the value of a dynamic game, whereby at state x the two opponents choose multistage (possibly nonstationary) policies that consist of functions of the current state, and continue to select moves using these policies over an infinite horizon. For further discussion of this interpretation, we refer to [Ber12a] and to books on dynamic games such as [FiV96]; see also [PaB99] and [Yu11] for an analysis of the undiscounted case ($\alpha = 1$) where there is a termination state, as in the stochastic shortest path problems of the subsequent Example 1.2.6.

Example 1.2.5 (Minimax Problems)

Consider a minimax version of Example 1.2.1, where w is not random but is rather chosen by an antagonistic player from a set $W(x, u)$. Let

$$H(x, u, J) = \sup_{w \in W(x,u)} \Big[g(x, u, w) + \alpha J\big(f(x, u, w)\big) \Big].$$

Then the equation $J^* = TJ^*$ is Bellman's equation for an infinite horizon minimax DP problem. A special case of this mapping arises in zero-sum dynamic games (cf. Example 1.2.4).

Example 1.2.6 (Stochastic Shortest Path Problems)

The stochastic shortest path (SSP for short) problem is the special case of the stochastic optimal control Example 1.2.1 where:

(a) There is no discounting ($\alpha = 1$).

(b) The state space is $X = \{t, 1, \ldots, n\}$ and we are given transition probabilities, denoted by

$$p_{xy}(u) = P(x_{k+1} = y \mid x_k = x, u_k = u), \qquad x, y \in X,\ u \in U(x).$$

(c) The control constraint set $U(x)$ is finite for all $x \in X$.

(d) A cost $g(x, u)$ is incurred when control $u \in U(x)$ is selected at state x.

(e) State t is a special termination state, which is cost-free and absorbing, i.e., for all $u \in U(t)$,

$$g(t,u) = 0, \qquad p_{tt}(u) = 1.$$

To simplify the notation, we have assumed that the cost per stage does not depend on the successor state, which amounts to using expected cost per stage in all calculations.

Since the termination state t is cost-free, the cost starting from t is zero for every policy. Accordingly, for all cost functions, we ignore the component that corresponds to t, and define

$$H(x,u,J) = g(x,u) + \sum_{y=1}^{n} p_{xy}(u)J(y), \quad x = 1,\ldots,n,\ u \in U(x),\ J \in \Re^n.$$

The mappings T_μ and T are defined by

$$(T_\mu J)(x) = g\big(x,\mu(x)\big) + \sum_{y=1}^{n} p_{xy}\big(\mu(x)\big)J(y), \qquad x = 1,\ldots,n,$$

$$(TJ)(x) = \min_{u\in U(x)} \left[g(x,u) + \sum_{y=1}^{n} p_{xy}(u)J(y) \right], \qquad x = 1,\ldots,n.$$

Note that the matrix that has components $p_{xy}(u)$, $x,y = 1,\ldots,n$, is sub-stochastic (some of its row sums may be less than 1) because there may be a positive transition probability from a state x to the termination state t. Consequently T_μ may be a contraction for some μ, but not necessarily for all $\mu \in \mathcal{M}$.

The SSP problem has been discussed in many sources, including the books [Pal67], [Der70], [Whi82], [Ber87], [BeT89], [HeL99], [Ber12a], and [Ber17a], where it is sometimes referred to by earlier names such as "first passage problem" and "transient programming problem." In the framework that is most relevant to our purposes, there is a classification of stationary policies for SSP into *proper* and *improper*. We say that $\mu \in \mathcal{M}$ is proper if, when using μ, there is positive probability that termination will be reached after at most n stages, regardless of the initial state; i.e., if

$$\rho_\mu = \max_{x=1,\ldots,n} P\{x_n \neq 0 \mid x_0 = x, \mu\} < 1.$$

Otherwise, we say that μ is improper. It can be seen that μ is proper if and only if in the Markov chain corresponding to μ, each state x is connected to the termination state with a path of positive probability transitions.

For a proper policy μ, it can be shown that T_μ is a weighted sup-norm contraction, as well as an n-stage contraction with respect to the unweighted sup-norm. For an improper policy μ, T_μ is not a contraction with respect to any norm. Moreover, T also need not be a contraction with respect to any norm (think of the case where there is only one policy, which is improper).

However, T is a weighted sup-norm contraction in the important special case where all policies are proper (see [BeT96], Prop. 2.2, or [Ber12a], Chapter 3).

Nonetheless, even in the case where there are improper policies and T is not a contraction, results comparable to the case of discounted finite-state MDP are available for SSP problems assuming that:

(a) There exists at least one proper policy.

(b) For every improper policy there is an initial state that has infinite cost under this policy.

Under the preceding two assumptions, referred to as the *strong SSP conditions* in Section 3.5.1, it was shown in [BeT91] that T has a unique fixed point J^*, the optimal cost function of the SSP problem. Moreover, a policy $\{\mu^*, \mu^*, \ldots\}$ is optimal if and only if

$$T_{\mu^*} J^* = T J^*.$$

In addition, J^* and J_μ can be computed by value iteration,

$$J^* = \lim_{k \to \infty} T^k J, \qquad J_\mu = \lim_{k \to \infty} T_\mu^k J,$$

starting with any $J \in \Re^n$ (see [Ber12a], Chapter 3, for a textbook account). These properties are in analogy with the desirable properties (a)-(c), given at the end of the preceding subsection in connection with contractive models.

Regarding policy iteration, it works in its strongest form when there are no improper policies, in which case the mappings T_μ and T are weighted sup-norm contractions. When there are improper policies, modifications to the policy iteration method are needed; see [Ber12a], [YuB13a], and also Section 3.6.2, where these modifications will be discussed in an abstract setting.

In Section 3.5.1 we will also consider SSP problems where the strong SSP conditions (a) and (b) above are not satisfied. Then we will see that unusual phenomena can occur, including that J^* may not be a solution of Bellman's equation. Still our line of analysis of Chapter 3 will apply to such problems.

Example 1.2.7 (Deterministic Shortest Path Problems)

The special case of the SSP problem where the state transitions are deterministic is the classical shortest path problem. Here, we have a graph of n nodes $x = 1, \ldots, n$, plus a destination t, and an arc length a_{xy} for each directed arc (x, y). At state/node x, a policy μ chooses an outgoing arc from x. Thus the controls available at x can be identified with the outgoing neighbors of x [the nodes u such that (x, u) is an arc]. The corresponding mapping H is

$$H(x, u, J) = \begin{cases} a_{xu} + J(u) & \text{if } u \neq t, \\ a_{xt} & \text{if } u = t, \end{cases} \qquad x = 1, \ldots, n.$$

A stationary policy μ defines a graph whose arcs are $\big(x, \mu(x)\big)$, $x = 1, \ldots, n$. The policy μ is proper if and only if this graph is acyclic (it consists of a tree of directed paths leading from each node to the destination). Thus there

exists a proper policy if and only if each node is connected to the destination with a directed path. Furthermore, an improper policy has finite cost starting from every initial state if and only if all the cycles of the corresponding graph have nonnegative cycle cost. It follows that the favorable analytical and algorithmic results described for SSP in the preceding example hold if the given graph is connected and the costs of all its cycles are positive. We will see later that significant complications result if the cycle costs are allowed to be zero, even though the shortest path problem is still well posed in the sense that shortest paths exist if the given graph is connected (see Section 3.1).

Example 1.2.8 (Multiplicative and Risk-Sensitive Models)

With x, y, u, and transition probabilities $p_{xy}(u)$, as in the finite-state MDP of Example 1.2.2, consider the mapping

$$H(x,u,J) = \sum_{y \in X} p_{xy}(u) g(x,u,y) J(y) = E\Big\{ g(x,u,y) J(y) \mid x, u \Big\}, \tag{1.15}$$

where g is a scalar function satisfying $g(x,u,y) \geq 0$ for all x, y, u (this is necessary for H to be monotone). This mapping corresponds to the multiplicative model of minimizing over all $\pi = \{\mu_0, \mu_1, \ldots\}$ the cost

$$J_\pi(x_0) = \limsup_{N \to \infty} E\Big\{ g\big(x_0, \mu_0(x_0), x_1\big) g\big(x_1, \mu_1(x_1), x_2\big) \cdots g\big(x_{N-1}, \mu_{N-1}(x_{N-1}), x_N\big) \mid x_0 \Big\}, \tag{1.16}$$

where the state sequence $\{x_0, x_1, \ldots\}$ is generated using the transition probabilities $p_{x_k x_{k+1}}\big(\mu_k(x_k)\big)$.

To see that the mapping H of Eq. (1.15) corresponds to the cost function (1.16), let us consider the unit function

$$\bar{J}(x) \equiv 1, \qquad x \in X,$$

and verify that for all $x_0 \in X$, we have

$$(T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} \bar{J})(x_0) = E\Big\{ g\big(x_0, \mu_0(x_0), x_1\big) g\big(x_1, \mu_1(x_1), x_2\big) \cdots g\big(x_{N-1}, \mu_{N-1}(x_{N-1}), x_N\big) \mid x_0 \Big\}, \tag{1.17}$$

so that

$$J_\pi(x) = \limsup_{N \to \infty} (T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} \bar{J})(x), \qquad x \in X.$$

Indeed, taking into account that $\bar{J}(x) \equiv 1$, we have

$$\begin{aligned} (T_{\mu_{N-1}} \bar{J})(x_{N-1}) &= E\Big\{ g\big(x_{N-1}, \mu_{N-1}(x_{N-1}), x_N\big) \bar{J}(x_N) \mid x_{N-1} \Big\} \\ &= E\Big\{ g\big(x_{N-1}, \mu_{N-1}(x_{N-1}), x_N\big) \mid x_{N-1} \Big\}, \end{aligned}$$

$$(T_{\mu_{N-2}}T_{\mu_{N-1}}\bar{J})(x_{N-2}) = \big((T_{\mu_{N-2}}(T_{\mu_{N-1}}\bar{J})\big)(x_{N-2})$$
$$= E\Big\{g\big(x_{N-2},\mu_{N-2}(x_{N-2}),x_{N-1}\big)$$
$$\cdot E\Big\{g\big(x_{N-1},\mu_{N-1}(x_{N-1}),x_N\big) \mid x_{N-1}\Big\} \mid x_{N-2}\Big\},$$

and continuing similarly,

$$(T_{\mu_0}T_{\mu_1}\cdots T_{\mu_{N-1}}\bar{J})(x_0) = E\Big\{g\big(x_0,\mu_0(x_0),x_1\big)E\Big\{g\big(x_1,\mu_1(x_1),x_2\big)\cdots$$
$$E\Big\{g\big(x_{N-1},\mu_{N-1}(x_{N-1}),x_N\big) \mid x_{N-1}\Big\} \mid x_{N-2}\Big\}\cdots\Big\} \mid x_0\Big\},$$

which by using the iterated expectations formula (see e.g., [BeT08]) proves the expression (1.17).

An important special case of a multiplicative model is when g has the form

$$g(x,u,y) = e^{h(x,u,y)}$$

for some one-stage cost function h. We then obtain a finite-state MDP with an exponential cost function,

$$J_\pi(x_0) = \limsup_{N\to\infty} E\Big\{e^{\left(h(x_0,\mu_0(x_0),x_1)+\cdots+h(x_{N-1},\mu_{N-1}(x_{N-1}),x_N)\right)}\Big\},$$

which is often used to introduce risk aversion in the choice of policy through the convexity of the exponential.

There is also a multiplicative version of the infinite state space stochastic optimal control problem of Example 1.2.1. The mapping H takes the form

$$H(x,u,J) = E\big\{g(x,u,w)J\big(f(x,u,w)\big)\big\},$$

where $x_{k+1} = f(x_k,u_k,w_k)$ is the underlying discrete-time dynamic system; cf. Eq. (1.10).

Multiplicative models and related risk-sensitive models are discussed extensively in the literature, mostly for the exponential cost case and under different assumptions than ours; see e.g., [HoM72], [Jac73], [Rot84], [ChS87], [Whi90], [JBE94], [FlM95], [HeM96], [FeM97], [BoM99], [CoM99], [BoM02], [BBB08], [Ber16a]. The works of references [DeR79], [Pat01], and [Pat07] relate to the stochastic shortest path problems of Example 1.2.6, and are the closest to the semicontractive models discussed in Chapters 3 and 4, based on the author's paper [Ber16a]; see the next example and Section 3.5.2.

Example 1.2.9 (Affine Monotonic Models)

Consider a finite state space $X = \{1,\ldots,n\}$ and a (possibly infinite) control constraint set $U(x)$ for each state x. For each policy μ, let the mapping T_μ be given by

$$T_\mu J = b_\mu + A_\mu J, \tag{1.18}$$

where b_μ is a vector of $\Re^n$ with components $b\big(x,\mu(x)\big)$, $x = 1,\ldots,n$, and A_μ is an $n\times n$ matrix with components $A_{xy}\big(\mu(x)\big)$, $x,y = 1,\ldots,n$. We assume that $b(x,u)$ and $A_{xy}(u)$ are nonnegative,

$$b(x,u) \geq 0, \qquad A_{xy}(u) \geq 0, \qquad \forall\ x,y = 1,\ldots,n,\ u\in U(x).$$

Thus T_μ and T map nonnegative functions to nonnegative functions $J : X \mapsto [0, \infty]$.

This model was introduced in the first edition of this book, and was elaborated on in the author's paper [Ber16a]. Special cases of the model include the finite-state Markov and semi-Markov problems of Examples 1.2.1-1.2.3, and the stochastic shortest path problem of Example 1.2.6, with A_μ being the transition probability matrix of μ (perhaps appropriately discounted), and b_μ being the cost per stage vector of μ, which is assumed nonnegative. An interesting affine monotonic model of a different type is the multiplicative cost model of the preceding example, where the initial function is $\bar{J}(x) \equiv 1$ and the cost accumulates multiplicatively up to reaching a termination state t. In the exponential case of this model, the cost of a generated path starting from some initial state accumulates additively as in the SSP case, up to reaching t. However, the cost of the model is the expected value of the *exponentiated* cost of the path up to reaching t. It can be shown then that the mapping T_μ has the form

$$(T_\mu J)(x) = p_{xt}\big(\mu(x)\big)\exp\big(g(x,\mu(x),t)\big) + \sum_{y=1}^{n} p_{xy}(\mu(x))\exp\big(g(x,\mu(x),y)\big)J(y), \qquad x \in X,$$

where $p_{xy}(u)$ is the probability of transition from x to y under u, and $g(x, u, y)$ is the cost of the transition; see Section 3.5.2 for a detailed derivation. Clearly T_μ has the affine monotonic form (1.18).

Example 1.2.10 (Aggregation)

Aggregation is an approximation approach that replaces a large DP problem with a simpler problem obtained by "combining" many of its states together into *aggregate states*. This results in an "aggregate" problem with fewer states, which may be solvable by exact DP methods. The optimal cost-to-go function of this problem is then used to approximate the optimal cost function of the original problem.

Consider an n-state Markovian decision problem with transition probabilities $p_{ij}(u)$. To construct an aggregation framework, we introduce a finite set $\mathcal{A}$ of aggregate states. We generically denote the aggregate states by letters such as x and y, and the original system states by letters such as i and j. The approximation framework is specified by combining in various ways the aggregate states and the original system states to form a larger system (see Fig. 1.2.2). To specify the probabilistic structure of this system, we introduce two (somewhat arbitrary) choices of probability distributions, which relate the original system states with the aggregate states:

(1) For each aggregate state x and original system state i, we specify the *disaggregation probability* d_{xi}. We assume that $d_{xi} \geq 0$ and

$$\sum_{i=1}^{n} d_{xi} = 1, \qquad \forall\ x \in \mathcal{A}.$$

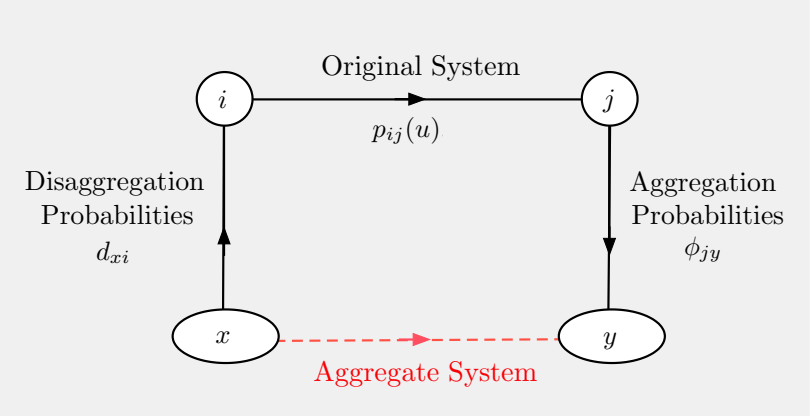

Figure 1.2.2 Illustration of the relation between aggregate and original system states.

Roughly, d_{xi} may be interpreted as the "degree to which x is represented by i."

(2) For each aggregate state y and original system state j, we specify the *aggregation probability* ϕ_{jy}. We assume that $\phi_{jy} \geq 0$ and

$$\sum_{y \in \mathcal{A}} \phi_{jy} = 1, \qquad \forall\, j = 1, \ldots, n.$$

Roughly, ϕ_{jy} may be interpreted as the "degree of membership of j in the aggregate state y."

The aggregation and disaggregation probabilities specify a dynamic system involving both aggregate and original system states (cf. Fig. 1.2.2). In this system:

(i) From aggregate state x, we generate original system state i according to d_{xi}.

(ii) We generate transitions from original system state i to original system state j according to $p_{ij}(u)$, with cost $g(i, u, j)$.

(iii) From original system state j, we generate aggregate state y according to ϕ_{jy}.

Illustrative examples of aggregation frameworks are given in the books [Ber12a] and [Ber17a]. One possibility is *hard aggregation*, where aggregate states are identified with the sets of a partition of the state space. For another type of common scheme, think of the case where the original system states form a fine grid in some space, which is "aggregated" into a much coarser grid. In particular let us choose a collection of "representative" original system states, and associate each one of them with an aggregate state. Thus, each aggregate state x is associated with a unique representative state i_x, and the

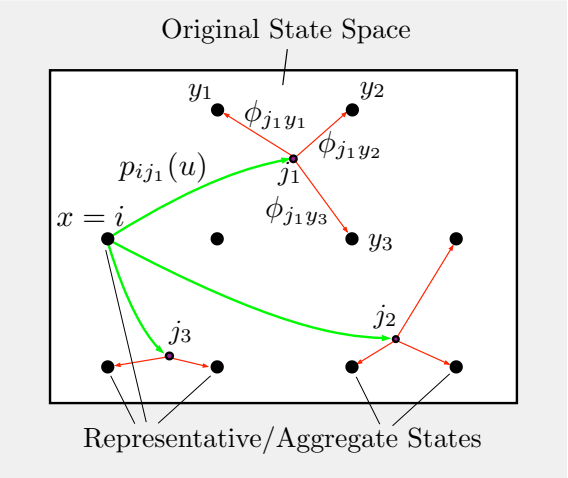

Figure 1.2.3 Aggregation based on a small subset of representative states (these are shown with larger dark circles, while the other (nonrepresentative) states are shown with smaller dark circles). In this figure, from representative state $x = i$, there are three possible transitions, to states j_1, j_2, and j_3, according to $p_{ij_1}(u), p_{ij_2}(u), p_{ij_3}(u)$, and each of these states is associated with a convex combination of representative states using the aggregation probabilities. For example, j_1 is associated with $\phi_{j_1 y_1} y_1 + \phi_{j_1 y_2} y_2 + \phi_{j_1 y_3} y_3$.

disaggregation probabilities are

$$d_{xi} = \begin{cases} 1 & \text{if } i = i_x, \\ 0 & \text{if } i \neq i_x. \end{cases} \tag{1.19}$$

The aggregation probabilities are chosen to represent each original system state j with a convex combination of aggregate/representative states; see Fig. 1.2.3. It is also natural to assume that the aggregation probabilities map representative states to themselves, i.e.,

$$\phi_{jy} = \begin{cases} 1 & \text{if } j = j_y, \\ 0 & \text{if } j \neq j_y. \end{cases}$$

This scheme makes intuitive geometrical sense as an interpolation scheme in the special case where both the original and the aggregate states are associated with points in a Euclidean space. The scheme may also be extended to problems with a continuous state space. In this case, the state space is discretized with a finite grid, and the states of the grid are viewed as the aggregate states. The disaggregation probabilities are still given by Eq. (1.19), while the aggregation probabilities may be arbitrarily chosen to represent each original system state with a convex combination of representative states.

As an extension of the preceding schemes, suppose that through some special insight into the problem's structure or some preliminary calculation, we know some features of the system's state that can "predict well" its cost. Then it seems reasonable to form the aggregate states by grouping together

states with "similar features," or to form aggregate states by using "representative features" instead of representative states. This is called "feature-based aggregation;" see the books [BeT96] (Section 3.1) and [Ber12a] (Section 6.5) for a description and analysis.

Given aggregation and disaggregation probabilities, one may define an *aggregate problem* whose states are the aggregate states. This problem involves an aggregate discrete-time system, which we will describe shortly. We require that the control is applied with knowledge of the current aggregate state only (rather than the original system state).† To this end, we assume that the control constraint set $U(i)$ is independent of the state i, and we denote it by U. Then, by adding the probabilities of all the relevant paths in Fig. 1.2.2, it can be seen that the transition probability from aggregate state x to aggregate state y under control $u \in U$ is

$$\hat{p}_{xy}(u) = \sum_{i=1}^{n} d_{xi} \sum_{j=1}^{n} p_{ij}(u)\phi_{jy}.$$

The corresponding expected transition cost is given by

$$\hat{g}(x,u) = \sum_{i=1}^{n} d_{xi} \sum_{j=1}^{n} p_{ij}(u)g(i,u,j).$$

These transition probabilities and costs define the aggregate problem.

We may compute the optimal costs-to-go $\hat{J}(x)$, $x \in \mathcal{A}$, of this problem by using some exact DP method. Then, the costs-to-go of each state j of the original problem are usually approximated by

$$\tilde{J}(j) = \sum_{y \in \mathcal{A}} \phi_{jy}\hat{J}(y).$$

Example 1.2.11 (Distributed Aggregation)

The abstract DP framework is useful not only in modeling DP problems, but also in modeling algorithms arising in DP and even other contexts. We illustrate this with an example from [BeY10] that relates to the distributed solution of large-scale discounted finite-state MDP using cost function approximation based on aggregation. ‡ It involves a partition of the n states into m subsets for the purposes of distributed computation, and yields a corresponding approximation $(V_1, \ldots, V_m)$ to the cost vector J^*.

In particular, we have a discounted n-state MDP (cf. Example 1.2.2), and we introduce aggregate states $S_1, \ldots, S_m$, which are disjoint subsets of

† An alternative form of aggregate problem, where the control may depend on the original system state is discussed in Section 6.5.2 of the book [Ber12a].

‡ See [Ber12a], Section 6.5.2, for a more detailed discussion. Other examples of algorithmic mappings that come under our framework arise in asynchronous policy iteration (see Sections 2.6.3, 3.6.2, and [BeY10], [BeY12], [YuB13a]), and in constrained forms of policy iteration (see [Ber11c], or [Ber12a], Exercise 2.7).

the original state space with $S_1 \cup \cdots \cup S_n = \{1, \ldots, n\}$. We envision a network of processors $\ell = 1, \ldots, m$, each assigned to the computation of a local cost function V_ℓ, defined on the corresponding aggregate state/subset S_ℓ:

$$V_\ell = \{V_{\ell y} \mid y \in S_\ell\}.$$

Processor ℓ also maintains a scalar aggregate cost R_ℓ for its aggregate state, which is a weighted average of the detailed cost values $V_{\ell x}$ within S_ℓ:

$$R_\ell = \sum_{x \in S_\ell} d_{\ell x} V_{\ell x},$$

where $d_{\ell x}$ are given probabilities with $d_{\ell x} \geq 0$ and $\sum_{x \in S_\ell} d_{\ell x} = 1$. The aggregate costs R_ℓ are communicated between processors and are used to perform the computation of the local cost functions V_ℓ (we will discuss computation models of this type in Section 2.6).

We denote $J = (V_1, \ldots, V_m, R_1, \ldots, R_m)$. We introduce the mapping $H(x, u, J)$ defined for each of the n states x by

$$H(x, u, J) = W_\ell(x, u, V_\ell, R_1, \ldots, R_m), \qquad \text{if } x \in S_\ell,$$

where for $x \in S_\ell$

$$W_\ell(x, u, V_\ell, R_1, \ldots, R_m) = \sum_{y=1}^{n} p_{xy}(u) g(x, u, y) + \alpha \sum_{y \in S_\ell} p_{xy}(u) V_{\ell y} + \alpha \sum_{y \notin S_\ell} p_{xy}(u) R_{s(y)},$$

and for each original system state y, we denote by $s(y)$ the index of the subset to which y belongs [i.e., $y \in S_{s(y)}$].

We may view H as an abstract mapping on the space of J, and aim to find its fixed point $J^* = (V_1^*, \ldots, V_m^*, R_1^*, \ldots, R_m^*)$. Then, for $\ell = 1, \ldots, m$, we may view V_ℓ^* as an approximation to the optimal cost vector of the original MDP starting at states $x \in S_\ell$, and we may view R_ℓ^* as a form of aggregate cost for S_ℓ. The advantage of this formulation is that it involves significant decomposition and parallelization of the computations among the processors, when performing various DP algorithms. In particular, the computation of $W_\ell(x, u, V_\ell, R_1, \ldots, R_m)$ depends on just the local vector V_ℓ, whose dimension may be potentially much smaller than n.

1.2.4 Approximation Models - Projected and Aggregation Bellman Equations

Given an abstract DP model described by a mapping H, we may be interested in fixed points of related mappings other than T and T_μ. Such mappings may arise in various contexts, such as for example distributed

asynchronous aggregation in Example 1.2.11. An important context is *subspace approximation*, whereby T_μ and T are restricted onto a subspace of functions for the purpose of approximating their fixed points. Much of the theory of approximate DP, neuro-dynamic programming, and reinforcement learning relies on such approximations (there are quite a few books, which collectively contain extensive accounts these subjects, such as Bertsekas and Tsitsiklis [BeT96], Sutton and Barto [SuB98], Gosavi [Gos03], Cao [Cao07], Chang, Fu, Hu, and Marcus [CFH07], Meyn [Mey07], Powell [Pow07], Borkar [Bor08], Haykin [Hay08], Busoniu, Babuska, De Schutter, and Ernst [BBD10], Szepesvari [Sze10], Bertsekas [Ber12a], [Ber17a], and Vrabie, Vamvoudakis, and Lewis [VVL13]).

For an illustration, consider the approximate evaluation of the cost vector of a discrete-time Markov chain with states $i = 1, \ldots, n$. We assume that state transitions (i, j) occur at time k according to given transition probabilities p_{ij}, and generate a cost $\alpha^k g(i, j)$, where $\alpha \in (0, 1)$ is a discount factor. The cost function over an infinite number of stages can be shown to be the unique fixed point of the Bellman equation mapping $T : \Re^n \mapsto \Re^n$ whose components are given by

$$(TJ)(i) = \sum_{j=1}^{n} p_{ij}(u)\big(g(i, j) + \alpha J(j)\big), \qquad i = 1, \ldots, n, \ J \in \Re^n.$$

This is the same as the mapping T in the discounted finite-state MDP Example 1.2.2, except that we restrict attention to a single policy. Finding the cost function of a fixed policy is the important policy evaluation subproblem that arises prominently within the context of policy iteration. It also arises in the context of a simplified form of policy iteration, the *rollout algorithm*; see e.g., [BeT96], [Ber12a], [Ber17a]. In some artificial intelligence contexts, policy iteration is referred to as *self-learning*, and in these contexts the policy evaluation is almost always done approximately, sometimes with the use of neural networks.

A prominent approach for approximation of the fixed point of T is based on the solution of lower-dimensional equations defined on the subspace $\{\Phi r \mid r \in \Re^s\}$ that is spanned by the columns of a given $n \times s$ matrix Φ. Two such approximating equations have been studied extensively (see [Ber12a], Chapter 6, for a detailed account and references; also [BeY07], [BeY09], [YuB10], [Ber11a] for extensions to abstract contexts beyond approximate DP). These are:

(a) The *projected equation*

$$\Phi r = \Pi_\xi T(\Phi r), \tag{1.20}$$

where Π_ξ denotes projection onto S with respect to a weighted Euclidean norm

$$\|J\|_\xi = \sqrt{\sum_{i=1}^{n} \xi_i \big(J(i)\big)^2} \tag{1.21}$$

with $\xi = (\xi_1, \ldots, \xi_n)$ being a probability distribution with positive components (sometimes a seminorm projection is used, whereby some of the components ξ_i may be zero; see [YuB12]).

(b) The *aggregation equation*

$$\Phi r = \Phi DT(\Phi r), \tag{1.22}$$

with D being an $s \times n$ matrix whose rows are restricted to be probability distributions; these are the disaggregation probabilities of Example 1.2.10. Also, in this approach, the rows of Φ are restricted to be probability distributions; they are the aggregation probabilities of Example 1.2.10.

We now see that solution of the projected equation (1.20) and the aggregation equation (1.22) amounts to finding a fixed point of the mappings $\Pi_\xi T$ and ΦDT, respectively. These mappings derive their structure from the DP operator T, so they have some DP-like properties, which can be exploited for analysis and computation.

An important fact is that the aggregation mapping ΦDT preserves the monotonicity and the sup-norm contraction property of T, while the projected equation mapping $\Pi_\xi T$ does not in general. The reason for preservation of monotonicity is the nonnegativity of the components of the matrices Φ and D (see the author's survey paper [Ber11c] for a discussion of the importance of preservation of monotonicity in various DP operations). The reason for preservation of sup-norm contraction is that the matrices Φ and D are sup-norm nonexpansive, because their rows are probability distributions. In fact, it can be verified that the solution r of Eq. (1.22) can be viewed as the *exact* DP solution of the "aggregate" DP problem that represents a lower-dimensional approximation of the original (see Example 1.2.10). The preceding observations are important for our purposes, as they indicate that much of the theory developed in this book applies to approximation-related mappings based on aggregation.

By contrast, the projected equation mapping $\Pi_\xi T$ need not be monotone, because the components of Π_ξ need not be nonnegative. Moreover while the projection Π_ξ is nonexpansive with respect to the projection norm $\|\cdot\|_\xi$, it need not be nonexpansive with respect to the sup-norm. As a result the projected equation mapping $\Pi_\xi T$ need not be a sup-norm contraction. These facts play a significant role in approximate DP methodology.

1.2.5 Multistep Models - Temporal Difference and Proximal Algorithms

An important possibility for finding a fixed point of T is to replace T with another mapping, say F, such that F and T have the same fixed points. For example, F may offer some advantages in terms of algorithmic convenience or quality of approximation when used in conjunction with projection or

aggregation [cf. Eqs. (1.20) or (1.22)]. Alternatively, F may be the mapping of some iterative method $x_{k+1} = F(x_k)$ that is suitable for computing fixed points of T.

In this book we will not consider in much detail the possibility of using an alternative mapping F to find a fixed point of a mapping T. We will just mention here some multistep versions of T, which have been used widely for approximations, particularly in connection with the projected equation approach. An important example is the mapping $T^{(\lambda)} : \Re^n \mapsto \Re^n$, defined for a given $\lambda \in (0,1)$ as follows: $T^{(\lambda)}$ transforms a vector $J \in \Re^n$ to the vector $T^{(\lambda)}J \in \Re^n$, whose n components are given by

$$\big(T^{(\lambda)}J\big)(i) = (1-\lambda)\sum_{\ell=0}^{\infty}\lambda^\ell (T^{\ell+1}J)(i), \qquad i = 1,\ldots,n,\ J \in \Re^n,$$

for $\lambda \in (0,1)$, where T^ℓ is the ℓ-fold composition of T with itself ℓ times. Here there should be conditions that guarantee the convergence of the infinite series in the preceding definition. The multistep analog of the projected Eq. (1.20) is

$$\Phi r = \Pi_\xi T^{(\lambda)}(\Phi r).$$

The popular temporal difference methods, such as TD(λ), LSTD(λ), and LSPE(λ), aim to solve this equation (see the book references on approximate DP, neuro-dynamic programming, and reinforcement learning cited earlier). The mapping $T^{(\lambda)}$ also forms the basis for the λ-policy iteration method to be discussed in Sections 2.5, 3.2.4, and 4.3.3.

The multistep analog of the aggregation Eq. (1.22) is

$$\Phi r = \Phi D T^{(\lambda)}(\Phi r),$$

and methods that are similar to the temporal difference methods can be used for its solution. In particular, a multistep method based on the mapping $T^{(\lambda)}$ is the, so-called, λ-aggregation method (see [Ber12a], Chapter 6), as well as other forms of aggregation (see [Ber12a], [YuB12]).

In the case where T is a linear mapping of the form

$$TJ = AJ + b,$$

where b is a vector in $\Re^n$, and A is an $n \times n$ matrix with eigenvalues strictly within the unit circle, there is an interesting connection between the multistep mapping $T^{(\lambda)}$ and another mapping of major importance in numerical convex optimization. This is the *proximal mapping*, associated with T and a scalar $c > 0$, and denoted by $P^{(c)}$. In particular, for a given $J \in \Re^n$, the vector $P^{(c)}J$ is defined as the unique vector $Y \in \Re^n$ that solves the equation

$$Y - AY - b = \frac{1}{c}(J - Y).$$

Equivalently,

$$P^{(c)}J = \left(\frac{c+1}{c}I - A\right)^{-1}\left(b + \frac{1}{c}J\right), \tag{1.23}$$

where I is the identity matrix. Then it can be shown (see Exercise 1.2 or the papers [Ber16b], [Ber17e]) that if

$$c = \frac{\lambda}{1-\lambda},$$

we have

$$T^{(\lambda)} = T \cdot P^{(c)} = P^{(c)} \cdot T.$$

Moreover, the vectors J, $P^{(c)}J$, and $T^{(\lambda)}J$ are colinear and satisfy

$$T^{(\lambda)}J = J + \frac{c+1}{c}\big(P^{(c)}J - J\big).$$

The preceding formulas show that $T^{(\lambda)}$ and $P^{(c)}$ are closely related, and that iterating with $T^{(\lambda)}$ is "faster" than iterating with $P^{(c)}$, since the eigenvalues of A are within the unit circle, so that T is a contraction. In addition, methods such as TD(λ), LSTD(λ), LSPE(λ), and their projected versions, which are based on $T^{(\lambda)}$, can be adapted to be used with $P^{(c)}$.

A more general form of multistep approach, introduced and studied in the paper [YuB12], replaces $T^{(\lambda)}$ with a mapping $T^{(w)} : \Re^n \mapsto \Re^n$ that has components

$$\big(T^{(w)}J\big)(i) = \sum_{\ell=1}^{\infty} w_{i\ell}(T^\ell J)(i), \qquad i = 1, \ldots, n, \ J \in \Re^n,$$

where w is a vector sequence whose ith component, $(w_{i1}, w_{i2}, \ldots)$, is a probability distribution over the positive integers. Then the multistep analog of the projected equation (1.20) is

$$\Phi r = \Pi_\xi T^{(w)}(\Phi r), \tag{1.24}$$

while the multistep analog of the aggregation equation (1.22) is

$$\Phi r = \Phi D T^{(w)}(\Phi r). \tag{1.25}$$

The mapping $T^{(\lambda)}$ is obtained for $w_{i\ell} = (1-\lambda)\lambda^{\ell-1}$, independently of the state i. A more general version, where λ depends on the state i, is obtained for $w_{i\ell} = (1-\lambda_i)\lambda_i^{\ell-1}$. The solution of Eqs. (1.24) and (1.25) by simulation-based methods is discussed in the paper [YuB12]; see also Exercise 1.3.

Let us also note that there is a connection between projected equations of the form (1.24) and aggregation equations of the form (1.25). This connection is based on the use of a seminorm [this is given by the same expression as the norm $\|\cdot\|_\xi$ of Eq. (1.21), with some of the components of ξ allowed to be 0]. In particular, the most prominent cases of aggregation equations can be viewed as seminorm projected equations because, for these cases, ΦD is a seminorm projection (see [Ber12a], p. 639, [YuB12], Section 4). Moreover, they can also be viewed as projected equations where the projection is oblique (see [Ber12a], Section 7.3.6).

1.3 ORGANIZATION OF THE BOOK

The examples of the preceding sections have illustrated how the monotonicity assumption is satisfied for most DP models, while the contraction assumption may or may not be satisfied. In particular, the contraction assumption is satisfied for the mapping H in Examples 1.2.1-1.2.5, assuming that there is discounting and that the cost per stage is bounded, but it need not hold in the SSP Example 1.2.6, the multiplicative Example 1.2.8, and the affine monotonic Example 1.2.9.

The main theme of this book is that the presence or absence of monotonicity and contraction is the primary determinant of the analytical and algorithmic theory of a typical total cost DP model. In our development, with few exceptions, we will assume that monotonicity holds. Consequently, the rest of the book is organized around the presence or absence of the contraction property. In the next three chapters we will discuss the following three types of models.

(a) **Contractive models:** These models, discussed in Chapter 2, have the richest and strongest algorithmic theory, and are the benchmark against which the theory of other models is compared. Prominent among them are discounted stochastic optimal control problems (cf. Example 1.2.1), finite-state discounted MDP (cf. Example 1.2.2), and some special types of SSP problems (cf. Example 1.2.6).

(b) **Semicontractive models:** In these models T_μ is monotone but it need not be a contraction for all $\mu \in \mathcal{M}$. Most deterministic, stochastic, and minimax-type shortest path problems of practical interest are of this type. One of the difficulties here is that under certain circumstances, some of the cost functions of the problem may take the values $+\infty$ or $-\infty$, and the mappings T_μ and T must accordingly be allowed to deal with such functions.

The salient characteristic of semicontractive models is that policies are separated into those that "behave well" with respect to our optimization framework and those that do not. It turns out that the notion of contraction is not sufficiently general for our purposes. We will thus introduce a related notion of "regularity," which is based on the idea that a policy μ should be considered "well-behaved" if the dynamic system defined by T_μ has J_μ as an asymptotically stable equilibrium within some domain. Our models and analysis are patterned to a large extent after the SSP problems of Example 1.2.6 (the regular μ correspond to the proper policies). We show that the (restricted) optimal cost function over just the regular policies may have favorable value and policy iteration properties. By contrast, the optimal cost function over all policies J^* may not be obtainable by these algorithms, and indeed J^* may not be a solution of Bellman's equation, as we will show with a simple example in Section 3.1.2.

The key idea is that under certain conditions, the restricted optimization (under the regular policies only) is well behaved, both analytically and algorithmically. Under still stronger conditions, which directly or indirectly guarantee that there exists an optimal regular policy, we prove that semicontractive models have strong properties, sometimes almost as strong as those of the contractive models.

In Chapter 3, we develop the basic theory of semicontractive models for the case where the regular policies are stationary, while in Chapter 4 (Section 4.4), we extend the notion of regularity to nonstationary policies. Moreover, we illustrate the theory with a variety of interesting shortest path-type problems (stochastic, minimax, affine monotonic, and risk sensitive/exponential cost), linear-quadratic optimal control problems, and deterministic and stochastic optimal control problems.

(c) **Noncontractive models:** These models rely on just the monotonicity property of T_μ, and are more complex than the preceding ones. As in semicontractive models, the various cost functions of the problem may take the values $+\infty$ or $-\infty$, and in fact the optimal cost function may take the values ∞ and $-\infty$ as a matter of course (rather than on an exceptional basis, as in semicontractive models). The complications are considerable, and much of the theory of the contractive models generalizes in weaker form, if at all. For example, in general the fixed point equation $J = TJ$ need not have a unique solution, the value iteration method may work starting with some functions but not with others, and the policy iteration method may not work at all. Of course some of these weaknesses may not appear in the presence of additional structure, and we will discuss in Sections 4.4-4.6 noncontractive models that also have some semicontractive structure, and corresponding favorable properties.

Examples of DP problems from each of the above model categories, mostly special cases of the specific DP models discussed in Section 1.2, are scattered throughout the book, both to illustrate the theory and its exceptions, and to illustrate the beneficial role of additional special structure. In some other types of models there are restrictions to the set of policies, so that $\mathcal{M}$ may be a strict subset of the set of functions $\mu : X \mapsto U$ with $\mu(x) \in U(x)$ for all $x \in X$. Such restrictions may include measurability (needed to establish a mathematically rigorous probabilistic framework) or special structure that enhances the characterization of optimal policies and facilitates their computation. These models were treated in Chapter 5 of the first edition of this book, and also in Chapter 6 of [BeS78]. †

† Chapter 5 of the first edition is accessible from the author's web site and the book's web page, and uses terminology and notation that are consistent with the present edition of the book.

Algorithms

Our discussion of algorithms centers on abstract forms of value and policy iteration, and is organized along three characteristics: *exact, approximate, and asynchronous*. The exact algorithms represent idealized versions, the approximate represent implementations that use approximations of various kinds, and the asynchronous involve irregular computation orders, where the costs and controls at different states are updated at different iterations (for example the cost of a single state being iterated at a time, as in Gauss-Seidel and other methods; see [Ber12a] for several examples of distributed asynchronous DP algorithms).

Approximate and asynchronous implementations have been the subject of intensive investigations since the 1980s, in the context of the solution of large-scale problems. Some of this methodology relies on the use of simulation, which is asynchronous by nature and is prominent in approximate DP. Generally, the monotonicity and sup-norm contraction structures of many of the prominent DP models favors the use of asynchronous algorithms in DP, as first shown in the author's paper [Ber82], and discussed at various points, starting with Section 2.6.

1.4 NOTES, SOURCES, AND EXERCISES

This monograph is written in a mathematical style that emphasizes simplicity and abstraction. According to the relevant Wikipedia article:

"Abstraction in mathematics is the process of extracting the underlying essence of a mathematical concept, removing any dependence on real world objects with which it might originally have been connected, and generalizing it so that it has wider applications or matching among other abstract descriptions of equivalent phenomena ... The advantages of abstraction are:

(1) It reveals deep connections between different areas of mathematics.

(2) Known results in one area can suggest conjectures in a related area.

(3) Techniques and methods from one area can be applied to prove results in a related area.

One disadvantage of abstraction is that highly abstract concepts can be difficult to learn. A degree of mathematical maturity and experience may be needed for conceptual assimilation of abstractions."

Consistent with the preceding view of abstraction, our aim has been to construct a minimalist framework, where the important mathematical structures stand out, while the application context is deliberately blurred. Of course, our development has to pass the test of relevance to applications. In this connection, we note that our presentation has integrated the relation of our abstract DP models with the applications of Section

1.2, and particularly discounted stochastic optimal control models (Chapter 2), shortest path-type models (Chapters 3 and 4), and undiscounted deterministic and stochastic optimal control models (Chapter 4). We have given illustrations of the abstract mathematical theory using these models and others throughout the text. A much broader and accessible account of applications is given in the author's two-volume DP textbook.

Section 1.2: The abstract style of mathematical development has a long history in DP. In particular, the connection between DP and fixed point theory may be traced to Shapley [Sha53], who exploited contraction mapping properties in analysis of the two-player dynamic game model of Example 1.2.4. Since that time the underlying contraction properties of discounted DP problems with bounded cost per stage have been explicitly or implicitly used by most authors that have dealt with the subject. Moreover, the value of the abstract viewpoint as the basis for economical and insightful analysis has been widely recognized.

An abstract DP model, based on unweighted sup-norm contraction assumptions, was introduced in the paper by Denardo [Den67]. This model pointed to the fundamental connections between DP and fixed point theory, and provided generality and insight into the principal analytical and algorithmic ideas underlying the discounted DP research up to that time. Abstract DP ideas were also researched earlier, notably in the paper by Mitten (Denardo's Ph.D. thesis advisor) [Mit64]; see also Denardo and Mitten [DeM67]. The properties of monotone contractions were also used in the analysis of sequential games by Zachrisson [Zac64].

Two abstract DP models that rely only on monotonicity properties were given by the author in the papers [Ber75], [Ber77]. They were patterned after the negative cost DP problem of Blackwell [Bla65] and the positive cost DP problem of Strauch [Str66] (see the monotone decreasing and monotone increasing models of Section 4.3). These two abstract DP models, together with the finite horizon models of Section 4.2, were used extensively in the book by Bertsekas and Shreve [BeS78] for the analysis of both discounted and undiscounted DP problems, ranging over MDP, minimax, multiplicative, and Borel space models.

Extensions of the analysis of the author's [Ber77] were given by Verdu and Poor [VeP87], which considered additional structure that allows the development of backward and forward value iterations, and in the thesis by Szepesvari [Sze98a], [Sze98b], which introduced non-Markovian policies into the abstract DP framework. The model of [Ber77] was also used by Bertsekas [Ber82], and Bertsekas and Yu [BeY10], to develop asynchronous value and policy iteration methods for abstract contractive and noncontractive DP models. Another line of related research involving abstract DP mappings that are not necessarily scalar-valued was initiated by Mitten [Mit74], and was followed up by a number of authors, including Sobel [Sob75], Morin [Mor82], and Carraway and Morin [CaM88].

Section 1.3: Generally, noncontractive total cost DP models with some special structure beyond monotonicity, fall in three major categories: monotone increasing models principally represented by positive cost DP, monotone decreasing models principally represented by negative cost DP, and transient models, exemplified by the SSP model of Example 1.2.6, where the decision process terminates after a period that is random and subject to control. Abstract DP models patterned after the first two categories have been known since [Ber77] and are further discussed in Section 4.3. The semicontractive models of Chapter 3 and Sections 4.4-4.6 (introduced and analyzed in the first edition of this book, as well as the subsequent series of papers and reports, [Ber14], [Ber15], [Ber16a], [BeY16], [Ber17b], [Ber17c], [Ber17d]), are patterned after the third category. Their analysis is based on the idea of separating policies into those that are well-behaved (these are called *regular*, and have contraction-like properties) and those that are not (these are called *irregular*). The objective of the analysis is then to explain the detrimental effects of the irregular policies, and to delineate the kind of model structure that can effectively limit these effects. As far as the author knows, this idea is new in the context of abstract DP. One of the aims of the present monograph is to develop this idea and to show that it leads to an important and insightful paradigm for conceptualization and solution of major classes of practical DP problems.

E X E R C I S E S

1.1 (Multistep Contraction Mappings)

This exercise shows how starting with an abstract mapping, we can obtain multistep mappings with the same fixed points and a stronger contraction modulus. Consider a set of mappings $T_\mu : \mathcal{B}(X) \mapsto \mathcal{B}(X)$, $\mu \in \mathcal{M}$, satisfying the contraction Assumption 1.2.2, let m be a positive integer, and let $\mathcal{M}_m$ be the set of m-tuples $\nu = (\mu_0, \ldots, \mu_{m-1})$, where $\mu_k \in \mathcal{M}$, $k = 1, \ldots, m-1$. For each $\nu = (\mu_0, \ldots, \mu_{m-1}) \in \mathcal{M}_m$, define the mapping $\overline{T}_\nu$, by

$$\overline{T}_\nu J = T_{\mu_0} \cdots T_{\mu_{m-1}} J, \qquad \forall\ J \in \mathcal{B}(X).$$

Show the contraction properties

$$\|\overline{T}_\nu J - \overline{T}_\nu J'\| \leq \alpha^m \|J - J'\|, \qquad \forall\ J, J' \in \mathcal{B}(X), \tag{1.26}$$

and

$$\|\overline{T} J - \overline{T} J'\| \leq \alpha^m \|J - J'\|, \qquad \forall\ J, J' \in \mathcal{B}(X), \tag{1.27}$$

where $\overline{T}$ is defined by

$$(\overline{T}J)(x) = \inf_{(\mu_0, \ldots, \mu_{m-1}) \in \mathcal{M}_m} (T_{\mu_0} \cdots T_{\mu_{m-1}} J)(x), \qquad \forall\ J \in \mathcal{B}(X),\ x \in X.$$

Solution: By the contraction property of $T_{\mu_0}, \ldots, T_{\mu_{m-1}}$, we have for all $J, J' \in B(X)$,

$$\begin{aligned}\|\overline{T}_\nu J - \overline{T}_\nu J'\| &= \|T_{\mu_0} \cdots T_{\mu_{m-1}} J - T_{\mu_0} \cdots T_{\mu_{m-1}} J'\| \\ &\leq \alpha \|T_{\mu_1} \cdots T_{\mu_{m-1}} J - T_{\mu_1} \cdots T_{\mu_{m-1}} J'\| \\ &\leq \alpha^2 \|T_{\mu_2} \cdots T_{\mu_{m-1}} J - T_{\mu_2} \cdots T_{\mu_{m-1}} J'\| \\ &\vdots \\ &\leq \alpha^m \|J - J'\|,\end{aligned}$$

thus showing Eq. (1.26).

We have from Eq. (1.26)

$$(T_{\mu_0} \cdots T_{\mu_{m-1}} J)(x) \leq (T_{\mu_0} \cdots T_{\mu_{m-1}} J')(x) + \alpha^m \|J - J'\|\, v(x), \qquad \forall\, x \in X,$$

and by taking infimum of both sides over $(T_{\mu_0} \cdots T_{\mu_{m-1}}) \in \mathcal{M}_m$ and dividing by $v(x)$, we obtain

$$\frac{(\overline{T}J)(x) - (\overline{T}J')(x)}{v(x)} \leq \alpha^m \|J - J'\|, \qquad \forall\, x \in X.$$

Similarly

$$\frac{(\overline{T}J')(x) - (\overline{T}J)(x)}{v(x)} \leq \alpha^m \|J - J'\|, \qquad \forall\, x \in X,$$

and by combining the last two relations and taking supremum over $x \in X$, Eq. (1.27) follows.

1.2 (Relation of Multistep and Proximal Mappings [Ber16b], [Ber17e])

Consider a linear mapping of the form

$$TJ = AJ + b,$$

where b is a vector in $\Re^n$, and A is an $n \times n$ matrix with eigenvalues strictly within the unit circle. Let $\lambda \in (0, 1)$ and $c = \frac{\lambda}{1-\lambda}$, and consider the multistep mapping

$$\left(T^{(\lambda)} J\right)(i) = (1 - \lambda) \sum_{\ell=0}^{\infty} \lambda^\ell (T^{\ell+1} J)(i), \qquad i = 1, \ldots, n,\ J \in \Re^n,$$

and the proximal mapping

$$P^{(c)} J = \left(\frac{c+1}{c} I - A\right)^{-1} \left(b + \frac{1}{c} J\right);$$

cf. Eq. (1.23) [equivalently, for a given J, $P^{(c)}J$ is the unique vector $Y \in \Re^n$ that solves the equation

$$Y - TY = \frac{1}{c}(J - Y),$$

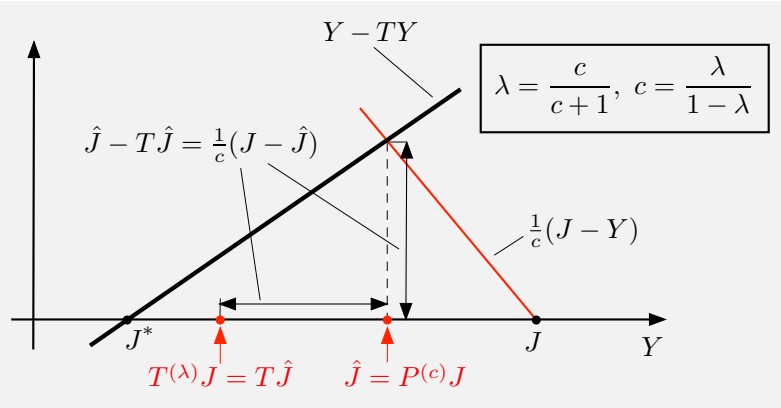

Figure 1.4.1. Illustration of the iterates $T^{(\lambda)}J$ and $P^{(c)}J$ for finding the fixed point J^* of a linear mapping T. Given J, we find the proximal iterate $\hat{J} = P^{(c)}J$ and then add the amount $\frac{1}{c}\big(\hat{J} - J\big)$ to obtain $T^{(\lambda)}J = TP^{(c)}J$. If T is a contraction mapping, $T^{(\lambda)}J$ is closer to J^* than $P^{(c)}J$.

(cf. Fig. 1.4.1)].

(a) Show that $P^{(c)}$ is given by

$$P^{(c)} = (1-\lambda)\sum_{\ell=0}^{\infty}\lambda^\ell T^\ell,$$

and can be written as

$$P^{(c)}J = \overline{A}^{(\lambda)}J + \overline{b}^{(\lambda)},$$

where

$$\overline{A}^{(\lambda)} = (1-\lambda)\sum_{\ell=0}^{\infty}\lambda^\ell A^\ell, \qquad \overline{b}^{(\lambda)} = \sum_{\ell=0}^{\infty}\lambda^{\ell+1}A^\ell b.$$

(b) Verify that

$$T^{(\lambda)}J = A^{(\lambda)}J + b^{(\lambda)},$$

where

$$A^{(\lambda)} = (1-\lambda)\sum_{\ell=0}^{\infty}\lambda^\ell A^{\ell+1}, \qquad b^{(\lambda)} = \sum_{\ell=0}^{\infty}\lambda^\ell A^\ell b,$$

and show that

$$T^{(\lambda)} = TP^{(c)} = P^{(c)}T, \tag{1.28}$$

and that for all $J \in \Re^n$,

$$P^{(c)}J = J + \lambda\big(T^{(\lambda)}J - J\big), \qquad T^{(\lambda)}J = J + \frac{c+1}{c}\big(P^{(c)}J - J\big). \tag{1.29}$$

Thus $T^{(\lambda)}J$ is obtained by extrapolation along the line segment $P^{(c)}J - J$, as illustrated in Fig. 1.4.1. Note that since T is a contraction mapping, $T^{(\lambda)}J$ is closer to J^* than $P^{(c)}J$.

(c) Show that for a given $J \in \Re^n$, the multistep and proximal iterates $T^{(\lambda)}J$ and $P^{(c)}J$ are the unique fixed points of the contraction mappings W_J and $\overline{W}_J$ given by

$$W_J Y = (1-\lambda)TJ + \lambda TY, \qquad \overline{W}_J Y = (1-\lambda)J + \lambda TY, \qquad Y \in \Re^n,$$

respectively.

(d) Show that the fixed point property of part (c) yields the following formula for the multistep mapping $T^{(\lambda)}$:

$$T^{(\lambda)}J = (1-\lambda A)^{-1}\big(b + (1-\lambda)AJ\big). \tag{1.30}$$

(e) (*Multistep Contraction Property for Nonexpansive A [BeY09]*) Instead of assuming that A has eigenvalues strictly within the unit circle, assume that the matrix $I - A$ is invertible and A is nonexpansive [i.e., has all its eigenvalues within the unit circle (possibly on the unit circle)]. Show that $A^{(\lambda)}$ is contractive (i.e., has eigenvalues that lie strictly within the unit circle) and its eigenvalues have the form

$$\theta_i = (1-\lambda)\sum_{\ell=0}^{\infty}\lambda^\ell \zeta_i^{\ell+1} = \frac{\zeta_i(1-\lambda)}{1-\zeta_i\lambda}, \qquad i = 1, \ldots, n, \tag{1.31}$$

where ζ_i, $i = 1, \ldots, n$, are the eigenvalues of A. *Note*: For an intuitive explanation of the result, note that the eigenvalues of $A^{(\lambda)}$ can be viewed as convex combinations of complex numbers from the unit circle at least two of which are different from each other, since $\zeta_i \neq 1$ by assumption (the nonzero corresponding eigenvalues of A and A^2 are different from each other). As a result the eigenvalues of $A^{(\lambda)}$ lie strictly within the unit circle.

(f) (*Contraction Property of Projected Multistep Mappings*) Under the assumptions of part (e), show that $\lim_{\lambda\to 1} A^{(\lambda)} = 0$. Furthermore, for any $n \times n$ matrix W, the matrix $WA^{(\lambda)}$ is contractive for λ sufficiently close to 1. In particular the projected mapping $\Pi A^{(\lambda)}$ and corresponding projected proximal mapping (cf. Section 1.2.5) become contractions as $\lambda \to 1$.

Solution: (a) The inverse in the definition of $P^{(c)}$ is written as

$$\left(\frac{c+1}{c}I - A\right)^{-1} = \left(\frac{1}{\lambda}I - A\right)^{-1} = \lambda(I-\lambda A)^{-1} = \lambda\sum_{\ell=0}^{\infty}(\lambda A)^\ell.$$

Thus, using the equation $\frac{1}{c} = \frac{1-\lambda}{\lambda}$,

$$\begin{aligned} P^{(c)}J &= \left(\frac{c+1}{c}I - A\right)^{-1}\left(b + \frac{1}{c}J\right) \\ &= \lambda\sum_{\ell=0}^{\infty}(\lambda A)^\ell\left(b + \frac{1-\lambda}{\lambda}J\right) \\ &= (1-\lambda)\sum_{\ell=0}^{\infty}(\lambda A)^\ell J + \lambda\sum_{\ell=0}^{\infty}(\lambda A)^\ell b, \end{aligned}$$

which is equal to $\overline{A}^{(\lambda)}J + \overline{b}^{(\lambda)}$. The formula $P^{(c)} = (1-\lambda)\sum_{\ell=0}^{\infty}\lambda^\ell T^\ell$ follows from this expression.

(b) The formula $T^{(\lambda)}J = A^{(\lambda)}J + b^{(\lambda)}$ is verified by straightforward calculation. We have,

$$\begin{aligned} TP^{(c)}J &= A\big(\overline{A}^{(\lambda)}J + \overline{b}^{(\lambda)}\big) + b \\ &= (1-\lambda)\sum_{\ell=0}^{\infty}\lambda^\ell A^{\ell+1}J + \sum_{\ell=0}^{\infty}\lambda^{\ell+1}A^{\ell+1}b + b = A^{(\lambda)}J + b^{(\lambda)} \\ &= T^{(\lambda)}J, \end{aligned}$$

thus proving the left side of Eq. (1.28). The right side is proved similarly. The interpolation/extrapolation formula (1.29) follows by a straightforward calculation from the definition of $T^{(\lambda)}$. As an example, to show the left side of Eq. (1.29), we write

$$\begin{aligned} J + \lambda\big(T^{(\lambda)}J - J\big) &= (1-\lambda)J + \lambda T^{(\lambda)}J \\ &= (1-\lambda)J + \lambda\left((1-\lambda)\sum_{\ell=0}^{\infty}\lambda^\ell A^{\ell+1}J + \sum_{\ell=0}^{\infty}\lambda^\ell A^\ell b\right) \\ &= (1-\lambda)\left(J + \sum_{\ell=1}^{\infty}\lambda^\ell A^\ell J\right) + \sum_{\ell=0}^{\infty}\lambda^{\ell+1}A^\ell b \\ &= \overline{A}^{(\lambda)}J + \overline{b}^{(\lambda)} \\ &= P^{(c)}J. \end{aligned}$$

(c) To show that $T^{(\lambda)}J$ is the fixed point of W_J, we must verify that

$$T^{(\lambda)}J = W_J\big(T^{(\lambda)}J\big),$$

or equivalently that

$$T^{(\lambda)}J = (1-\lambda)TJ + \lambda T\big(T^{(\lambda)}J\big) = (1-\lambda)TJ + \lambda T^{(\lambda)}(TJ).$$

The right-hand side, in view of the interpolation formula

$$(1-\lambda)J + \lambda T^{(\lambda)}J = P^{(c)}J, \qquad \forall\ x \in \Re^n,$$

is equal to $P^{(c)}(TJ)$, which from the formula $T^{(\lambda)} = P^{(c)}T$ [cf. part (b)], is equal to $T^{(\lambda)}J$. The proof is similar for $\overline{W}_J$.

(d) The fixed point property of part (c) states that $T^{(\lambda)}J$ is the unique solution of the following equation in Y:

$$Y = (1-\lambda)TJ + \lambda TY = (1-\lambda)(AJ+b) + \lambda(AY+b),$$

from which the desired relation follows.

(e), (f) The formula (1.31) follows from the expression for $A^{(\lambda)}$ given in part (b). This formula can be used to show that the eigenvalues of $A^{(\lambda)}$ lie strictly within the unit circle, using also the fact that the matrices A^m, $m \geq 1$, and $A^{(\lambda)}$ have the same eigenvectors (see [BeY09] for details). Moreover, the eigenvalue formula shows that all eigenvalues of $A^{(\lambda)}$ converge to 0 as $\lambda \to 1$, so that $\lim_{\lambda\to 1} A^{(\lambda)} = 0$. This also implies that $WA^{(\lambda)}$ is contractive for λ sufficiently close to 1.

1.3 (State-Dependent Weighted Multistep Mappings [YuB12])

Consider a set of mappings $T_\mu : \mathcal{B}(X) \mapsto \mathcal{B}(X)$, $\mu \in \mathcal{M}$, satisfying the contraction Assumption 1.2.2. Consider also the mappings $T_\mu^{(w)} : \mathcal{B}(X) \mapsto \mathcal{B}(X)$ defined by

$$(T_\mu^{(w)} J)(x) = \sum_{\ell=1}^{\infty} w_\ell(x) \big(T_\mu^\ell J\big)(x), \qquad x \in X,\ J \in \mathcal{B}(X),$$

where $w_\ell(x)$ are nonnegative scalars such that for all $x \in X$,

$$\sum_{\ell=1}^{\infty} w_\ell(x) = 1.$$

Show that

$$\frac{\big|(T_\mu^{(w)} J)(x) - (T_\mu^{(w)} J')(x)\big|}{v(x)} \leq \sum_{\ell=1}^{\infty} w_\ell(x)\, \alpha^\ell \|J - J'\|, \qquad \forall\ x \in X,$$

where α is the contraction modulus of T_μ, so that $T_\mu^{(w)}$ is a contraction with modulus

$$\bar{\alpha} = \sup_{x \in X} \sum_{\ell=1}^{\infty} w_\ell(x)\, \alpha^\ell \leq \alpha.$$

Show also that for all $\mu \in \mathcal{M}$, the mappings T_μ and $T_\mu^{(w)}$ have the same fixed point.

Solution: By the contraction property of T_μ, we have for all $J, J' \in B(X)$ and $x \in X$,

$$\begin{aligned}
\frac{\big|(T_\mu^{(w)} J)(x) - (T_\mu^{(w)} J')(x)\big|}{v(x)} &= \frac{\big|\sum_{\ell=1}^{\infty} w_\ell(x)(T_\mu^\ell J)(x) - \sum_{\ell=1}^{\infty} w_\ell(x)(T_\mu^\ell J')(x)\big|}{v(x)} \\
&\leq \sum_{\ell=1}^{\infty} w_\ell(x) \|T_\mu^\ell J - T_\mu^\ell J'\| \\
&\leq \left(\sum_{\ell=1}^{\infty} w_\ell(x) \alpha^\ell\right) \|J - J'\|,
\end{aligned}$$

showing the contraction property of $T_\mu^{(w)}$.

Let J_μ be the fixed point of T_μ. By using the relation $(T_\mu^\ell J_\mu)(x) = J_\mu(x)$, we have for all $x \in X$,

$$(T_\mu^{(w)} J_\mu)(x) = \sum_{\ell=1}^{\infty} w_\ell(x) \big(T_\mu^\ell J_\mu\big)(x) = \left(\sum_{\ell=1}^{\infty} w_\ell(x)\right) J_\mu(x) = J_\mu(x),$$

so J_μ is the fixed point of $T_\mu^{(w)}$ [which is unique since $T_\mu^{(w)}$ is a contraction].