D.P. Heyman, M.J. Sobel, Eds., *Handbooks in OR & MS, Vol. 2*

Chapter 8

Markov Decision Processes

*Martin L. Puterman**

Faculty of Commerce, The University of British Columbia, Vancouver, B.C., Canada V6T 1Y8

1. Introduction

This chapter presents theory, applications and computational methods for Markov Decision Processes (MDP's). MDP's are a class of stochastic sequential decision processes in which the cost and transition functions depend only on the current state of the system and the current action. These models have been applied in a wide range of subject areas, most notably in queueing and inventory control.

A sequential decision process is a model for dynamic system under the control of a decision maker. At each point of time at which a decision can be made, the decision maker, who will be referred to as he with no sexist connotations intended, observes the state of the system. Based on the information from this observation, he chooses an action from a set of available alternatives. The consequences of choosing this action are twofold; the decision maker receives an immediate reward, and specifies a probability distribution on the subsequent system state. If the probability distribution is degenerate, the problem is deterministic. The decision maker's objective is to choose a sequence of actions called a policy, that will optimize the performance of the system over the decision making horizon. Since the action selected at present affects the future evolution of the system, the decision maker cannot choose his action without taking into account future consequences.

Sequential decision processes are classified according to the times (epochs) at which decisions are made, the length of the decision making horizon, the mathematical properties of the state and action spaces and the optimality criteria. The primary focus of this chapter is problems in which decisions are made periodically at discrete time points (e.g. minutes, hours, days, weeks, months or years). The state and action sets are either finite, countable, compact or Borel; their characteristics determine the form of the reward and

* This research has been supported by Natural Sciences and Engineering Research Council (Canada) Grant A-5527.

transition probability functions. The optimality criteria considered include finite and infinite horizon expected total reward, infinite horizon expected total discounted reward and average expected reward.

The main objectives in analyzing sequential decision processes in general and MDP's in particular include:

(a) providing an optimality equation which characterizes the supremal value of the objective function,

(b) characterizing the form of an optimal policy if it exists,

(c) developing efficient computational procedures for finding policies which are optimal or close to optimal.

The optimality or Bellman equation is the basic entity in MDP theory and almost all existence, characterization and computational results are based on analysis of it. The form of this equation depends on the optimality criteria and the nature of the states, actions, rewards and transition probability functions. Because of this, the analysis in this chapter is divided according to optimality criterion. Section 4 treats the finite horizon total expected reward case, Section 6 the expected total discounted reward criteria, Section 7 the expected total reward criteria and Section 8 the average expected and sensitive optimality criteria.

In almost any problem, the class of policies can be made sufficiently general so that an optimal or nearly optimal policy exists. An important theoretical and practical consideration is under what conditions does there exist a policy in a specified class that is optimal or nearly optimal among the class of all policies. The class of stationary policies, each of which uses a decision rule depending only on the current state of the system, independent of its history or stage, receives special attention in this chapter. In many applications, an important further consideration is whether an optimal policy has some special form or structure. If this is the case, optimal policies can be found by restricting search to policies of this form.

For each optimality criterion, results are presented in as much generality as possible. Vector space notation is used to simplify presentation and unify results. Numerical results are necessarily for finite state and action problems and presented almost exclusively for the finite horizon expected total reward and infinite horizon expected total discounted reward criteria. In the finite horizon case, calculations are based on backward induction which is often referred to as dynamic programming, while for infinite horizon problems, calculations are based on value iteration, policy iteration and their variants.

Awareness of the importance and potential applicability of Markov decision processes began with the appearance of the books of Bellman (1957) and Howard (1960). Many precursors to their work abound; the interested reader is referred to Heyman and Sobel (1984) and Denardo (1982) for historical synopses. From a theoretical prospective, important early work includes Blackwell (1962), Dubins and Savage (1965), Strauch (1966), Denardo (1967) and Veinott (1967). In addition to those referred to above, noteworthy books include Hinderer (1970), Derman (1970), Whittle (1983), Ross (1985), Bertsekas (1987), Hernández-Lerma (1989) and Puterman (1991).

This chapter is broad in scope. In Section 2, Markov decision processes are introduced and formal notation is presented. Section 3 gives some simple examples of MDP's. Section 4 is concerned with finite horizon problems while section 5 provides an introduction to the infinite horizon setting which is presented in considerable detail in Sections 6–8. Semi-Markov decision problems are treated in Section 9. An extensive bibliography is included.

Related topics not covered in this chapter include partially observable Markov decision processes (Monahan, 1982 and Bertsekas, 1987) and Markov decision processes with estimated parameters (Mandl, 1974 and Hubner, 1988).

For a quick overview of the subject the reader should focus on Sections 2, 3, 4.4, 6.1–6.6 and 8.1–8.8.

2. Problem formulation

This section defines the basic elements of a Markov decision process and presents notation.

2.1. *Problem definition and notation*

A system under control of a decision maker is observed as it evolves through time. The set T consists of all time points at which the system is observed and decisions can be made. These time points are denoted by t and are referred to as *decision epochs* or *stages*. The set T can be classified in two ways; T is either finite or infinite and either a discrete set or a continuum. *The primary focus of the chapter is when T is discrete*; a special case of the continuous time model will be discussed in Section 9 of this chapter; the continuous time model will be the focus of Chapter 9.

Discrete time problems are classified as either *finite horizon* or *infinite horizon* according to whether the set T is finite or infinite. The problem formulation in these two cases will be almost identical; however, theory and computational methods will differ considerably. For finite horizon problems, $T = \{1, 2, \ldots, N\}$ and for infinite horizon problems, $T = \{1, 2, \ldots\}$.

The set of possible states of the system at time t is denoted by S_t. In finite horizon problems, S_t is defined for $t = 1, 2, \ldots, N+1$ although decisions are only made at times $t = 1, 2, \ldots, N$. This is because the decision at time N often has future consequences which can be summarized by evaluating the state of the system at time $N+1$.

If at time t, the decision maker observes the system in state $s \in S_t$, he must choose an action, a, from the set of allowable actions at time t, $A_{s,t}$. In most of the literature, S_t and $A_{s,t}$ are either finite, countably infinite or compact subsets of the real line. The most general setting is when S_t and $A_{s,t}$ are non-empty Borel subsets of complete, seperable metric spaces. *To avoid measurability and integrability issues, all of the analysis in this chapter will assume that S_t is discrete.*

There are two consequences of choosing action a when the system is in state s at time t; the decision maker receives an immediate reward and the probability distribution for the state of the system at the next stage is determined. The reward is denoted by the real valued function $r_t(s, a)$; when it is positive it can be thought of as income and when negative as cost. In some applications, it is convenient to think of $r_t(s, a)$ as the *expected* reward received at time t. This will be the case if the reward for the current period depends on the state of the system at the next decision epoch. In such situations $r_t(s, a, j)$ is the reward received in period t if the state of the system at time t is s, action $a \in A_s$ is selected and the system is in state j at time $t+1$. Then the expected reward in period t is

$$r_t(s, a) = \sum_{j \in S_{t+1}} r_t(s, a, j) p_t(j \mid s, a)$$

where $p_t(j \mid s, a)$ is defined below. The example in Section 3.2 illustrates this alternative.

The function $p_t(j \mid s, a)$ denotes the probability that the system is in state $j \in S_{t+1}$ if action $a \in A_{s,t}$ is chosen in state s at time t; $p_t(j \mid s, a)$ is called the *transition probability function*. When S_t is not discrete, $p_t(j \mid s, a)$ is a density, if it exists; otherwise the problem formulation is in terms of a distribution function. In most applications it is convenient to assume that

$$\sum_{j \in S_{t+1}} p_t(j \mid s, a) = 1 . \tag{2.1}$$

The collection of objects $(T, S_t, A_{s,t}, p_t(j \mid s, a), r_t(s, a))$ is a *Markov decision process*. Its distinguishing feature is that the transition probability function and reward function depend only on the current state of the system and the current action selected. Some authors refer to the above collection of objects as a *Markov decision problem*; here that terminology is reserved for a Markov decision process together with an optimality criterion. Generalizations allow the rewards and transition probability functions to depend on some or all previous states and actions (cf. Hinderer, 1970).

A *decision rule* is a function $d_t : S_t \to A_{s,t}$ that specifies the action the decision maker chooses when the system is in state s at time t, that is, $d_t(s) \in A_{s,t}$ for each $s \in S_t$. A decision rule of this type is said to be *Markovian* because it depends only on the current state and stage of the system and not on its past. It is referred to as *deterministic* because it selects an action with certainty. The set of allowable decision rules at time t is denoted by D_t and is called the *decision set*.

Generalizations of deterministic Markovian decision rules play an important role in MDP theory. A decision rule is said to be *history dependent* if it is a function of the entire past history of the system as summarized in the sequence of previous states and actions, $(s_1, a_1, \ldots, s_{t-1}, a_{t-1}, s_t)$. A decision rule is said to be *randomized* if in each state it specifies a probability distribution on

the set of allowable actions. A deterministic decision rule can be regarded as a special case of a randomized rule in which the probability distribution is degenerate. Some authors distinguish deterministic rules from randomized rules by referring to them as *pure*. If a decision maker uses a randomized decision rule, the action that is actually implemented is a random event. An important theoretical issue is when is it optimal to use a deterministic Markovian decision rule at each stage.

A *policy* specifies the decision rule to be used by the decision maker at each $t \in T$. It tells the decision maker which decision to choose under any possible future state the system might occupy. A policy π is a sequence of decision rules, i.e. $\pi = \{d_1, d_2, \ldots, d_N\}$ where $d_t \in D_t$ for $t = 1, 2, \ldots, N$ for $N \leq \infty$. It will be referred to as randomized, history remembering, deterministic or pure if each decision rule it contains has that property. Let Π denote the set of all history remembering randomized policies; $\Pi = D_1 \times D_2 \times \cdots \times D_N$, $N \leq \infty$. A policy is said to be *stationary* if it uses the identical decision rule in each period i.e. $\pi = (d, d, \ldots)$. Stationary policies play an important role in the theory of infinite horizon Markov decision processes and are discussed in more detail in the next subsection.

The following subsets of Π are distinguished; Π_{M}, the set of all Markovian policies, Π_{P}, the set of all non-randomized (pure) history dependent policies, Π_{S}, the set of all stationary (possibly randomized) Markovian policies and Π_{D}, the set of all deterministic stationary Markovian policies. Observe that $\Pi_{\mathrm{D}} \subset \Pi_{\mathrm{S}} \subset \Pi_{\mathrm{M}} \subset \Pi$ and $\Pi_{\mathrm{P}} \subset \Pi_{\mathrm{D}} \subset \Pi$.

Specifying a policy $\pi = \{d_1, d_2, \ldots, d_N\}$ induces a stochastic process, $\{X_t^{\pi}; t \in T\}$, with time varying state space S_t. It represents the state of the system at each decision epoch. When the policy π is Markov, that is each decision rule is Markov, then the stochastic process $\{X_t^{\pi}; t \in T\}$ is a discrete time Markov chain. If the policy is Markov and stationary, then the Markov chain is also stationary. This Markov chain together with the sequence of rewards $\{r_t(s, d_t(s)),\ s \in S_t,\ t \in T\}$ is often referred to as a *Markov reward process*. The stochastic process $\{r_t(X_t^{\pi}, d_t(X_t^{\pi}));\ t \in T\}$ is the stream of rewards received by the decision maker if he adopts policy π.

2.2. *Stationary Markov decision problems*

Frequently in infinite horizon problems, the data is *stationary*. This means that the set of states, the set of allowable actions in each state, the rewards, the transition or transfer functions and the decision sets are the same at every stage. When this is the case, the time subscript t is deleted and notation S, A_s, $r(s, a)$, $p(j|s, a)$ and D is used. Under most optimality criteria, stationary policies are optimal when the data of the problem is stationary. The stationary policy which uses the decision rule d at each stage will be often denoted by d.

When all the data is stationary, the following additional notation will be useful. Let $d \in D$ be an arbitrary decision rule and define $r_d(s) = r(s, d(s))$ and $p_d(j|s) = p(j|s, d(s))$. These quantities are the one period reward and transi-

tion probability function if the system is in state s and the action corresponding to decision rule system is in state s and the action corresponding to decision rule $d(s)$ is used. Note that if d is a randomized decision rule, then

$$r_d(s) = \sum_{a \in A_s} r(s, a) P\{d(s) = a\}$$

and

$$p_d(j \mid s) = \sum_{a \in A_s} p(j \mid s, a) P\{d(s) = a\} .$$

It is convenient to use vector space notation for analysis and presentation of results. Define V to be family of bounded real valued functions on S, that is, $v \in V$ if $v : S \to R$. For each $v \in V$ define the *norm* of v by $\|v\| = \sup_{s \in S} |v(s)|$. When S is finite, 'sup' can be replaced by 'max' in this definition. The vector space V together with the norm $\|\cdot\|$ is a complete, normed linear space or Banach space (cf. Liusternik and Sobolev, 1961).

Let r_d be the m-vector, with sth component $r_d(s)$ and let P_d be the $m \times m$ matrix with it (s, j)th entry given by $p_d(j \mid s)$, where m is the number of elements in S. Assume that $r_d \in V$. This means that r_d is bounded. (In Section 6.8 the case of unbounded rewards will be considered.) Since P_d is a probability matrix, it follows that $P_d v \in V$ for all $v \in V$. The quantity r_d is referred to as the *reward vector* and P_d as the *transition probability matrix* corresponding to decision rule d. When policy $\pi = (d_1, d_2, \ldots, d_N)$ is used, the (s, j)th component of the n-step transition probability P_π^n is given by

$$P_\pi^n(s, j) = P_{d_1} P_{d_2} \cdots P_{d_n}(s, j) = P(X_n^\pi = j \mid X_1^\pi = s) . \tag{2.2}$$

3. Examples

This section presents two very simple examples of Markov decision processes. The reader is referred to White (1985b) for a recent survey of applications of MDP's.

3.1. A two state Markov decision process

The following simple example will be useful for illustrating the basic concepts of a Markov decision process and serve as an example of several results in latter sections.

At each t the system can be in either of two states s_1 and s_2. Whenever the system is in state 1 there are two possible actions $a_{1,1}$ and $a_{1,2}$ and in state 2 there is one action $a_{2,1}$. As a consequence of choosing action $a_{1,1}$ in s_1, the

decision maker receives an immediate reward of 5 units and at the next decision epoch the system is in state s_1 with probability 0.5 and state s_2 with probability 0.5. If instead action $a_{1,2}$ is chosen in state s_1, the decision maker receives an immediate reward of 10 units and at the next decision epoch the system is in state s_2 with probability 1. If the system is in state s_2, the decision maker must choose action $a_{2,1}$. As a consequence of this decision, the decision maker incurs a cost of 1 unit and at the next decision epoch, the system is in state s_2 with certainty. Figure 3.1 gives a convenient symbolic representation of this problem.

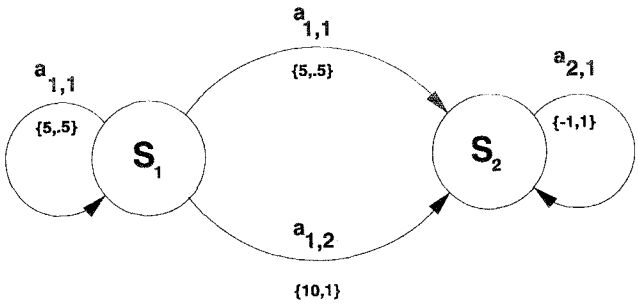

Fig. 3.1. Symbolic representation of two state Markov process.

The above problem is stationary, i.e., the set of states, the sets of actions, the rewards and transition probabilities do not depend on the stage in which the decision is made. Thus, the t subscript on these quantities is unnecessary and will be deleted in subsequent references to the problem. A formal description follows.

Decision epochs:

$$T = \{1, 2, \ldots, N\}, \quad N \leqslant \infty .$$

States:

$$S_t = \{s_1, s_2\}, \quad t \in T .$$

Actions:

$$A_{s_1,t} = \{a_{1,1}, a_{1,2}\}, \quad A_{s_2,t} = \{a_{2,1}\}, \qquad t \in T .$$

Rewards:

$$r_t(s_1, a_{1,1}) = 5, \quad r_t(s_1, a_{1,2}) = 10, \quad r_t(s_2, a_{2,1}) = -1, \qquad t \in T .$$

Transition probabilities:

$$p_t(j|s_1, a_{1,1}) = 0.5\,, \quad j = s_1, s_2\,,$$
$$p_t(s_1|s_1, a_{1,2}) = 0\,, \quad p_t(s_2|s_1, a_{1,2}) = 1\,,$$
$$p_t(s_1|s_2, a_{2,1}) = 0\,, \quad p_t(s_2|s_1, a_{2,1}) = 1\,, \qquad t \in T\,.$$

3.2. *A stochastic inventory control problem*

This section presents a simplified version of an inventory control problem. Some non-standard assumptions have been made to facilitate calculations. The numerical example below will be used to illustrate the algorithmic methods in the subsequent sections. For more detail on inventory theory, the reader is referred to Chapter 12.

The problem is as follows. Each month, the manager of a warehouse determines current inventory of a product. Based on this information, he decides whether or not to order additional product. In doing so, he is faced with a tradeoff between the costs associated with keeping inventory on hand and the lost sales or penalties associated with being unable to satisfy customer demand for the product. The manager's objective is to maximize some measure of profit (sales revenue less inventory holding and ordering costs) over the decision making horizon. The demand for the product throughout the month is random with a known probability distribution.

Several simplifying assumptions enable a concise formulation. They are:

(a) the decision to order additional stock is made at the beginning of each month and delivery occurs instantaneously,

(b) demand for the product arives throughout the month but all orders are filled on the last day of the month.

(c) if demand exceeds the stock on hand, the customer goes elsewhere to purchase the product (excess demand is lost),

(d) the revenues and costs, and the demand distribution are stationary (identical each month)

(e) the product can be sold only in whole units, and

(f) the warehouse capacity is M units.

Let s_t denote the inventory on hand at the beginning of month t, a_t the additional amount of product ordered in month t and D_t the random demand in month t. The demand has a known probability distribution given by $p_j = P\{D_t = j\}$, $j = 0, 1, 2, \ldots$. The cost of ordering u units in any month is $O(u)$ and the cost of storing u units for 1 month is $h(u)$. The ordering cost is given by:

$$O(u) = \begin{cases} K + c(u) & \text{if } u > 0\,, \\ 0 & \text{if } u = 0\,. \end{cases} \tag{3.1}$$

The functions $c(u)$ and $h(u)$ are increasing in u. For finite horizon problems, the inventory on hand after the last decision epoch has value $g(u)$. Finally, if j

units of product are demanded in a month and the inventory u exceeds demand, the manager receives $f(j)$, if demand exceeds inventory, then he receives $f(u)$. Let $F(u)$ be the expected revenue in a month if the inventory prior to receipt of customer orders is u units. It is given in period t by

$$F(u) = \sum_{j=0}^{u-1} f(j)p_j + f(u)P\{D_t \geq u\} . \tag{3.2}$$

This problem is formulated as a Markov decision process as follows. Decision epochs:

$$T = \{1, 2, \ldots, N\} , \quad N \leq \infty .$$

States (the amount of inventory on hand at the start of a month):

$$S_t = \{0, 1, 2, \ldots, M\} , \quad t = 1, 2, \ldots, N+1 .$$

Actions (the amount of additional stock to order in month t):

$$A_{s,t} = \{0, 1, 2, \ldots, M - s\} , \quad t = 1, 2, \ldots, N .$$

Expected rewards (expected revenue less ordering and holding costs):

$$r_t(s, a) = F(s + a) - O(a) - h(s + a) , \quad t = 1, 2, \ldots, N ;$$

(the value of terminal inventory);

$$r_{N+1}(s, a) = g(s) , \quad t = N + 1 .$$

Transition probabilities (see explanation below):

$$p_t(j \mid s, a) = \begin{cases} 0 & \text{if } M \geq j > s + a , \\ p_{s+a-j} & \text{if } M \geq s + a \geq j > 0 , \\ q_{s+a} & \text{if } j = 0, s + a \leq M \text{ and } s + a \leq D_t , \end{cases}$$

where

$$q_{s+a} = P\{D_t \geq s + a\} = \sum_{d=s+a}^{\infty} p_d .$$

A brief explanation of the derivation of the transition probabilities follows. If the inventory on hand at the beginning of period t is s units and an order is placed for a units, the inventory prior to external demand is $s + a$ units. For the inventory on hand at the start of period $t + 1$ to be $j > 0$ units, the demand in period t had to have been $s + a - j$ units. This occurs with probability p_{s+a-j}.

If the demand exceeds $s + a$ units, then the inventory at the start of period $t + 1$ is 0 units. This occurs with probability q_{s+a}. Finally the probability that the inventory level ever exceeds $s + a$ units is 0, since demand is non-negative.

As a consequence of assumption (b) above, the inventory on hand throughout the month is $s + a$ so that the total monthly holding cost is $h(s + a)$. If instead, the demand is assumed to arrive at the beginning of a month $h(s + a)$ is the expected holding cost.

The decision sets consist of all rules which assign the quantity of inventory to be ordered each month to each possible starting inventory position in a month. A policy is a sequence of such ordering rules. An example of a decision rule is: order only if the inventory level is below 3 units at the start of the month and order the quantity which raises the stock level to 10 units. In month t this decision rule is given by:

$$d_t(s) = \begin{cases} 10 - s, & s < 3, \\ 0, & s \geqslant 3. \end{cases}$$

Such a policy is called in (s, S) policy (See Chapter 12 for more details).

A numerical example is now provided. It will be solved in subsequent sections using dynamic programming methods. The data for the problem are as follows: $K = 4$, $c(u) = 2u$, $g(u) = 0$, $h(u) = u$, $M = 3$, $N = 3$, $f(u) = 8u$ and

$$p_d = \begin{cases} \frac{1}{4} & \text{if } d = 0, \\ \frac{1}{2} & \text{if } d = 1, \\ \frac{1}{4} & \text{if } d = 2. \end{cases}$$

The inventory is constrained to be 3 or fewer units and the manager wishes to consider the effects over three months. All costs and revenues are linear. This means that for each unit ordered the per unit cost is 2, for each unit held in inventory for 1 month, the per unit cost is 1 and for each unit sold the per unit revenue is 8. The expected revenue when u units of stock are on hand prior to receipt of an order is given in Table 3.1.

Table 3.1

u	$F(u)$
0	0
1	$0 \times \frac{1}{4} + 8 \times \frac{3}{4} = 6$
2	$0 \times \frac{1}{4} + 8 \times \frac{1}{2} + 16 \times \frac{1}{4} = 8$
3	$0 \times \frac{1}{4} + 8 \times \frac{1}{2} + 16 \times \frac{1}{4} = 8$

Combining the expected revenue with the ordering, and holding costs gives the expected profit in period t if the inventory level is s at the start of the period and an order for a units is placed. If $a = 0$, the ordering and holding cost equals s and if a is positive, it equals $4 + s + 3a$. It is summarized in the table

below where an × corresponds to an infeasible action. Transition probabilities only depend on the total inventory on hand prior to receipt of orders. They are the same for any s and a which have the same value for $s + a$. To reduce redundant information, transition probabilities are presented as functions of $s + a$ only. The information in Table 3.2 defines this problem completely.

Table 3.2

	$r_t(s, a)$					$p_t(j \mid s, a)$			
	$a=0$	$a=1$	$a=2$	$a=3$		$j=0$	$j=1$	$j=2$	$j=3$
$s=0$	0	−1	−2	−5	$s+a=0$	1	0	0	0
$s=1$	5	0	−3	×	$s+a=1$	$\frac{3}{4}$	$\frac{1}{4}$	0	0
$s=2$	6	−1	×	×	$s+a=2$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	0
$s=3$	5	×	×	×	$s+a=3$	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

4. The finite horizon case

This section presents and analyzes finite horizon, discrete time Markov decision problems. It introduces a concept of optimality and discusses the structure of optimal policies and their computation. The Principle of Optimality which underlies the backward induction procedure is shown to be the basis for analysis. The section concludes with a numerical example.

4.1. Optimality criteria

Each policy yields a stream of random rewards over the decision making horizon. In order to determine which policy is best, a method of comparing these reward streams is necessary. Most of the dynamic programming literature assumes that the decision maker has a linear, additive and risk neutral utility function over time and uses expected utility as an evaluation function. Consequently, the expected total reward over the decision making horizon is used for reward stream evaluation and comparison.

The results in this section requires a formal definition of the history of a Markov decision process. Let H_t denote the history up to epoch t, $t = 1, 2, \ldots, N+1$. Define $A_t = \times_{s \in S_t} A_{s,t}$. Then

$$H_1 = \{S_1\}\,, \tag{4.1a}$$

$$H_t = \{S_1, A_1, S_2, \ldots, A_{t-1}, S_t\}$$
$$= \{H_{t-1}, A_{t-1}, S_t\}\,, \quad t = 2, \ldots, N+1\,. \tag{4.1b}$$

Equation (4.1b) shows that H_t can be defined inductively in terms of H_{t-1}. This means that for $h_t \in H_t$, $h_t = (s_1, a_1, s_2, \ldots, a_{t-1}, s_t) = (h_{t-1}, a_{t-1}, s_t)$ with $h_{t-1} \in H_{t-1}$. The history contains the sequence of states and realized actions of

the process up to decision epoch t. Clearly, requiring policies to depend on entire histories will be computationally inhibitive and impractical.

Let $\pi = (d_1, d_2, \ldots, d_N)$ be a history dependent policy. That is, for each $t = 1, \ldots, N$, $d_t : H_t \rightarrow A_t$. When a policy π is selected and a history realized, denote the corresponding history by H_t^π. Let $v_N^\pi(s)$ equal the expected total reward over the planning horizon if policy π is used and the system is in state s at the first decision epoch. It is given by

$$v_N^\pi(s) = E_{\pi,s}\left\{\sum_{t=1}^{N} r_t(X_t^\pi, d_t(H_t^\pi)) + r_{N+1}(X_{N+1}^\pi)\right\} \tag{4.2}$$

where $E_{\pi,s}$ denotes expectation with respect to the joint probability distribution of the stochastic process determined by π conditional on the state of the system prior to the first decision being s. If the policy is randomized, this distribution also takes into account the realization of the action selection process at each decision epoch.

Under the assumption that $r_t(s, a)$ is bounded for $(s, t) \in S_t \times A_{s,t}$, $v_N^\pi(s)$ exists and is bounded for each $\pi \in \Pi$ and each $N < \infty$. If rewards are *discounted*, that is a reward received in a subsequent period is worth less than a reward received in the current period, a discount factor λ^{t-1}, $0 < \lambda < 1$, is included inside the summation in (4.2). This will not alter any results in this section but will be important in the infinite horizon case.

The decision maker's objective is to specify (at decision epoch 1) a policy $\pi \in \Pi$ with the largest expected total reward. When both S_t and $A_{s,t}$ are finite there are only finitely many policies so such a policy is guaranteed to exist and can be found by enumeration. In this case, the decision maker's problem is that of finding a π^* with the property that

$$v_N^{\pi^*}(s) = \max_{\pi \in \Pi} v_N^\pi(s) \equiv v_N^*(s), \quad s \in S_1. \tag{4.3}$$

The policy π^* is called an *optimal policy* and $v_N^*(s)$ is the *optimal value function* or *value* of the finite horizon Markov decision problem. Theory in the finite horizon case is concerned with characterizing π^* and computing $v_N^*(s)$.

When the problem is such that the maximum in (4.3) is not attained, the maximum is replaced by a supremum and the value of the problem is given by

$$v_N^*(s) = \sup_{\pi \in \Pi} v_N^\pi(s), \quad s \in S_1. \tag{4.4}$$

In such cases, the decision maker's objective is to find an *ε-optimal policy*, that is, for any $\varepsilon > 0$, a policy π_ε^* with the property that

$$v_N^{\pi_\varepsilon^*}(s) + \varepsilon > v_N^*(s), \quad s \in S_1.$$

By the definition of the supremum, such a policy is guaranteed to exist. The problem defined above is a discrete time, finite horizon Markov decision problem with expected total reward criterion.

4.2. Policy evaluation

In this subsection the basic recursion of dynamic programming is introduced in the context of computing the expected total reward of a *fixed* policy. Let $\pi = (d_1, d_2, \ldots, d_N)$ be a history remembering policy. For each t define the expected reward received in periods $t, t+1, \ldots, N+1$ if the history at epoch t is $h_t \in H_t$ by

$$u_t^\pi(h_t) = E_{\pi,h_t}\left\{\sum_{n=t}^{N+1} r_n(X_n^\pi, d_n(H_n^\pi))\right\}. \tag{4.5}$$

The expectation in (4.5) is with respect to the process determined by policy π conditional on the history up to epoch t being h_t. Note that $u_1^\pi = v_N^\pi$. The difference between these quantities is that v_N^π is defined in terms of the entire future, while u_t^π is defined in terms of a portion of the future beginning in period t.

The following algorithm gives an inductive procedure for evaluating the return of a fixed history dependent policy. It is a basis for several of the results below. The *policy evaluation algorithm* for evaluating the return of policy $\pi = (d_1, d_2, \ldots, d_N)$ is as follows. For ease of exposition, it is assumed that π is deterministic but it is not required to be Markov or stationary.

The Finite Horizon Policy Evaluation Algorithm.

1. Set $t = N+1$ and

$$u_{N+1}^\pi(h_{N+1}) = r_{N+1}(s_{N+1}) \quad \text{for all } h_{N+1} = (h_N, a_N, s_{N+1}) \in H_{N+1}.$$

2. Substitute $t-1$ for t and compute $u_t^\pi(h_t)$ for each $h_t \in H_t$ by

$$u_t^\pi(h_t) = r_t(s_t, d_t(h_t)) + \sum_{j \in S_{t+1}} p_t(j \mid s_t, d_t(h_t)) u_{t+1}^\pi(h_t, d_t(h_t), j) \tag{4.6}$$

noting that $(h_t, d_t(h_t), j) = h_{t+1} \in H_{t+1}$.

3. If $t = 1$, stop, otherwise return to step 2.

The idea leading to equation (4.6) for general t is as follows. The expected value of policy π over periods $t, t+1, \ldots, N+1$ if the history at epoch t is h_t is equal to the immediate reward received if action $d_t(h_t)$ is selected plus the expected reward over the remaining periods. The second term contains the product of the probability of being in state j at epoch $t+1$ if action $d_t(h_t)$ is used, and the expected reward obtained using policy π over periods $t+1, \ldots, N+1$ if the history at epoch $t+1$ is $h_{t+1} = (h_t, d_t(h_t), j)$. Summing over all possible j gives the desired expectation expressed in terms of u_{t+1}^π instead of in terms of the reward functions and conditional probabilities required to explicitly write out (4.5).

This inductive scheme reduces the problem of computing expected rewards over $N+1$ periods to a sequence of N similar 1 period problems having immediate reward r_t and terminal reward u_{t+1}^{π}. This reduction is the essence of dynamic programming; *multistage problems are reduced to a sequence of simpler inductively defined single stage problems*. The procedure to find optimal policies that is described below, is quite similar.

That u_t^{π} agrees with that defined in equation (4.5) is based on the following argument which can be made formal through induction. First this algorithm fixes the value of u_{N+1}^{π} to be the terminal reward that would be obtained if the history at epoch $N+1$ was h_{N+1}. Clearly this is the correct value. It next evaluates u_N^{π} for all possible histories h_N using equation (4.6) in step 2. This equation is the basic recurrence. By writing out u_N^{π} from (4.5) explicitly and substituting u_{N+1}^{π} for r_{N+1}, these expressions are seen to agree.

Fundamental to justifying the inductive calculation above is the additivity of the policy evaluation equation (4.2). This additivity arises from the assumption of linear utility. Other utility functions have been considered by Howard and Matheson (1972), Jacquette (1973), Eagle (1975) and Rothblum (1984). White (1988) surveys the use of variance and other probabilistic criteria.

4.3. The optimality equation and the principle of optimality

This section introduces the optimality equation and investigates its properties. It shows that solutions correspond to optimal value functions and that these value functions can be used to determine optimal policies. Proofs of results appear in Heyman and Sobel (1984, p. 112–124) and Derman (1970, p. 11–17). Hinderer (1970) considers the problem in more generality by allowing the set of feasible decisions at each epoch to be history dependent.

Let

$$u_t^*(h_t) = \sup_{\pi \in \Pi} u_t^{\pi}(h_t) . \tag{4.7}$$

The quantity u_t^* is the supremal return over the remainder of the decision horizon when the history up to time t is h_t. When minimizing costs instead of maximizing rewards this is sometimes called a *cost-to-go* function (Bertsekas, 1987).

The *optimality equations* of dynamic programming are the fundamental entities in the theory of Markov decision problems. They are often referred to as *functional equations* or *Bellman equations* and are the basis for the backward induction algorithm. They are given by

$$u_t(h_t) = \sup_{a \in A_{s_t,t}} \left\{ r_t(s_t, a) + \sum_{j \in S_{t+1}} p_t(j \mid s_t, a) u_{t+1}(h_t, a, j) \right\} \tag{4.8}$$

for $t = 1, \ldots, N$ and $h_t \in H_t$. When $t = N+1$, the boundary condition $u_{N+1} = r_{N+1}$ is imposed. In many applications, as well as in Section 7, r_{N+1} is identically zero. These equations reduce to the policy evaluation equations

(4.6) by replacing supremum over all actions by the action corresponding to a specific policy. When the supremum in (4.8) is attained, for instance when each $A_{s_t,t}$ is finite, 'max' is used instead.

A solution to the system of equations (4.8) is a sequence of functions $u_t : H_t \to A_t$, $t = 1, \ldots, N$, with the property that u_N satisfies the Nth equation, u_{N-1} satisfies the $(N-1)$th equation with the u_N which satisfies the Nth equation substituted into the right hand side of the $(N-1)$th equation, etc. These equations have several important and useful properties:

(a) Solutions to the optimality equations are the optimal returns from period t onward for each t.

(b) They provide sufficient conditions to determine whether a policy is optimal.

(c) They yield an efficient procedure for computing optimal return functions and policies.

(d) They can be used to determine theoretical properties of policies and return functions.

The following theorem summarizes the optimality properties.

Theorem 4.1. *Suppose u_t is a solution of* (4.8) *for $t = 1, \ldots, N$ and $u_{N+1} = r_{N+1}$. Then*

(a) $u_t(h_t) = u_t^*(h_t)$ *for all* $h_t \in H_t$, $t = 1, \ldots, N+1$, *and*

(b) $u_1(s_1) = v_N^*(s_1)$ *for all* $s_1 \in S_1$.

Result (a) means that solutions of the optimality equation are the optimal value functions from period t onward for each t and result (b) means that the solution to the first equation is the value function for the MDP. Note that no assumptions have been imposed on the state space and the result is valid whenever the summation in (4.8) is defined. In particular, the results hold for finite and countable state problems.

Result (b) is the statement that the optimal value from epoch 1 onward is the optimal value function for the N period problem. It is an immediate consequence of (a). The proof of (a) is based on the backward induction argument; it appears in the references above.

The next theorem shows how the optimality equation can be used to find optimal policies when the maximum is attained on the right hand side of the optimality equation. Theorem 4.3 considers the case of a supremum.

Theorem 4.2. *Suppose u_t^*, $t = 1, \ldots, N$, are solutions of the optimality equations* (4.8) *and $u_{N+1} = r_{N+1}$. Define the policy $\pi^* = (d_1^*, d_2^*, \ldots, d_N^*)$ for $t = 1, \ldots, N$ by*

$$r_t(s_t, d_t^*(h_t)) + \sum_{j \in S_{t+1}} p_{t+1}(j \mid s_t, d_t^*(h_t)) u_{t+1}^*(h_t, d_t^*(h_t), j)$$

$$= \max_{a \in A_{s_t,t}} \left\{ r_t(s_t, a) + \sum_{j \in S_{t+1}} p_t(j \mid s_t, a) u_{t+1}^*(h_t, a, j) \right\}. \tag{4.9}$$

Then:

(a) π^* *is an optimal policy and*

$$v_N^{\pi^*}(s) = v_N^*(s), \quad s \in S_1. \tag{4.10}$$

(b) *For each* $t = 1, 2, \ldots, N+1$,

$$u_t^{\pi^*}(h_t) = u_t^*(h_t), \quad h_t \in H_t. \tag{4.11}$$

Equation (4.9) is often expressed as

$$d_t^*(h_t) = \arg\max_{a \in A_{s_t,t}} \left\{ r_t(s_t, a) + \sum_{j \in S_{t+1}} p_t(j \mid s_t, a) u_{t+1}^*(h_t, a, j) \right\}. \tag{4.12}$$

The operation 'arg max' corresponds to choosing an action which attains the maximum on the right hand side of (4.12). It is not necessarily unique.

The theorem means that an optimal policy is found by first solving the optimality equations and then for each history choosing a decision rule which selects any action which attains the maximum on the right hand side of (4.9). When using these equations in computation, the right hand side is evaluated for all $a \in A_{s_t,t}$ and the maximizing actions are recorded. An optimal policy is one which for each history selects any of these maximizing actions.

Part (b) of this theorem is known as 'The Principle of Optimality', and is considered to be the basic paradigm of dynamic programming. It first appeared formally in Bellman (1957, p. 83) as:

"*An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*"

An equivalent statement that yields further insight appears in Denardo (1982, p. 15). It can be paraphrased in the language of this chapter as:

There exists at least one policy that is optimal for the remainder of the decision making horizon for each state at every stage.

The policy π^* has these properties.

In case the supremum in (4.8) is not attained, the decision maker can use an ε-optimal policy which is found as follows.

Theorem 4.3. *Let* $\varepsilon > 0$ *be arbitrary and suppose* u_t^*, $t = 1, \ldots, N+1$, *are solutions of the optimality equations* (4.8) *and* $u_{N+1}^* = r_{N+1}$. *Define the policy* $\pi^\varepsilon = (d_1^\varepsilon, d_2^\varepsilon, \ldots, d_N^\varepsilon)$ *to be any policy satisfying*

$$r_t(s_t, d_t^\varepsilon(h_t)) + \sum_{j \in S_{t+1}} p_t(j \mid s_t, d_t^\varepsilon(h_t)) u_{t+1}^*(h_t, d_t^\varepsilon(h_t), j) + \frac{\varepsilon}{N}$$

$$\geqslant \sup_{a \in A_{s_t,t}} \left\{ r_t(s_t, a) + \sum_{j \in S_{t+1}} p_t(j \mid s_t, a) u_{t+1}^*(h_t, a, j) \right\}. \tag{4.13}$$

Then:

(a) π^{ε} *is an* ε*-optimal policy with*

$$v_N^{\pi^{\varepsilon}}(s) + \varepsilon \geq v_N^*(s), \quad s \in S_1. \tag{4.14}$$

(b) *For each* $t = 1, 2, \ldots, N$,

$$u_t^{\pi^{\varepsilon}}(h_t) + (N - t + 1)(\varepsilon/N) \geq u_t^*(h_t), \quad h_t \in H_t. \tag{4.15}$$

The following corollary is an immediate consequence of Theorems 4.2 and 4.3.

Corollary 4.4. *There exists an* (ε-) *optimal deterministic policy.*

The result follows since the policy obtained in these two theorems is deterministic. When the maximum is attained, the policy defined by (4.12) is optimal, otherwise the policy defined by (4.13) is ε-optimal.

The following important theorem states that the optimal value functions and policies depend only on the state of the system at decision epochs and not on the past, that is, when rewards and transition probabilities are Markov, the optimal policy depends on the past only throught the current state.

Theorem 4.5. *Let* u_t, $t = 1, \ldots, N$, *with* $u_{N+1} = r_{N+1}$, *be solutions of* (4.8). *Then if* $u_{N+1}(h_{N+1})$ *depends on* h_{N+1} *only through* s_{N+1}:

(a) *for each* $t = 1, \ldots, N$, $u_t(h_t)$ *depends on* h_t *only through* s_t, *and*

(b) *there exists* (ε-) *optimal deterministic Markov policies.*

A formal proof is based on induction. The key idea in establishing this result is that if for some n, u_{n+1} depends on the history only through the current state, then the maximizing action and u_n depend on the history only through the current state.

4.4. The backward induction algorithm

This section presents the backward induction or dynamic programming algorithm for finding optimal policies and value functions for finite horizon MDP's. The maximum in (4.8) is assumed to be attained. The results of Theorem 4.5 allow restriction to deterministic Markov policies.

The Backward Induction Algorithm.

1. Set $t = N + 1$ and

$$u_t(s_t) = r_t(s_t) \quad \text{for all } s_t \in S_t.$$

2. Substitute $t - 1$ for t and compute $u_t(s_t)$ for each $s_t \in S_t$ by

$$u_t(s_t) = \max_{a \in A_{s_t,t}} \left\{ r_t(s_t, a) + \sum_{j \in S_{t+1}} p_t(j \mid s_t, a) u_{t+1}(s_t) \right\}. \tag{4.16}$$

Denote by $A^*_{s_t,t}$, the set of actions $a_{s_t,t}$ satisfying

$$a_{s_t,t} = \arg\max_{a \in A_{s_t,t}} \left\{ r_t(s_t, a) + \sum_{j \in S_{t+1}} p_t(j \mid s_t, a) u_{t+1}(s_t) \right\} . \tag{4.17}$$

3. If $t = 1$, stop. Otherwise return to step 2.

Comparison of this algorithm to the policy evaluation algorithm of Section 4.2, shows that it accomplishes the following:

(a) It finds sets of actions $A^*_{s_t,t}$ which contain all actions in $A_{s_t,t}$ which attain the maximum in (4.17).

(b) It evaluates any policy which selects an action in $A^*_{s_t,t}$ for each $s_t \in S_t$ for all $t \in T$.

(c) It computes the expected total reward for the entire decision making horizon and from each period to the end of the horizon for any policy described in (b).

An immediate consequence of Theorem 4.1 is that $u_1 = v^*_N$ and $u_t = u^*_t$ for all t. Thus this algorithm finds the optimal reward functions and *all* optimal policies. A single optimal policy can be found by restricting step 2 to obtain only one action (instead of all actions) which attains the arg max in (4.17).

4.5. Computational results

The backward induction algorithm is used to solve the numerical version of the stochastic inventory example of Section 3.2. (Since the data are stationary, the time index may be deleted.) Define $u_t(s, a)$ by

$$u_t(s, a) = r(s, a) + \sum_{j \in S} p(j \mid s, a) u_{t+1}(j) . \tag{4.18}$$

Computations proceed as follows.

1. Set $t = 4$ and $u_4(s) = r_4(s) = 0$, $s = 0, 1, 2, 3$.
2. Since $t \neq 1$, continue. Set $t = 3$ and for $s = 0, 1, 2, 3$,

$$u_3(s) = \max_{a \in A_s} \left\{ r(s, a) + \sum_{j \in S} p(j \mid s, a) u_4(j) \right\} = \max_{a \in A_s} \{ r(s, a) \} .$$

In each state the maximizing action is 0. Thus for all s, $A^*_{s,3} = \{0\}$ and $u_3(0) = 0$, $u_3(1) = 6$, $u_3(2) = 6$ and $u_3(3) = 5$.

3. Since $t \neq 1$, continue. Set $t = 2$ and

$$u_2(s) = \max_{a \in A_s} \{ u_2(s, a) \} .$$

The quantities $u_2(s, a)$, $u_2(s)$ and $A^*_{s,2}$ are summarized in Table 4.1 where $\times$'s denote non-existent actions.

Table 4.1

	$u_2(s, a)$				$u_2(s)$	$A^*_{s,2}$
	$a=0$	$a=1$	$a=2$	$a=3$		
$s=0$	0	1/4	2	1/2	2	2
$s=1$	$6\frac{1}{4}$	4	$2\frac{1}{2}$	×	$6\frac{1}{4}$	0
$s=2$	10	$4\frac{1}{4}$	×	×	10	0
$s=3$	$10\frac{1}{2}$	×	×	×	$10\frac{1}{2}$	0

4. Since $t \neq 1$, continue. Set $t = 1$ and

$$u_1(s) = \max_{a \in A_s} \{u_1(s, a)\} .$$

The quantities $u_1(s, a)$, $u_1(s)$ and $A^*_{s,1}$ are summarized in Table 4.2.

Table 4.2

	$u_1(s, a)$				$u_1(s)$	$A^*_{s,1}$
	$a=0$	$a=1$	$a=2$	$a=3$		
$s=0$	2	33/16	66/16	67/16	67/16	3
$s=1$	129/16	98/16	99/16	×	129/16	0
$s=2$	194/16	131/16	×	×	194/16	0
$s=3$	227/16	×	×	×	227/16	0

5. Since $t = 1$. Stop.

This procedure has produced the optimal expected total reward function $v^*_3(s)$ and optimal policy $\pi^* = (d^*_1(s), d^*_2(s), d^*_3(s))$ which are reproduced in Table 4.3.

Table 4.3

s	$d^*_1(s)$	$d^*_2(s)$	$d^*_3(s)$	$v^*_3(s)$
0	3	2	0	67/16
1	0	0	0	129/16
2	0	0	0	194/16
3	0	0	0	227/16

The quantity $v^*_3(s)$ gives the expected total reward obtained using this policy when the inventory at the start of month 1 is s units.

This policy has a particularly simple form; if at the start of month 1 the inventory is 0 units, order 3 units, otherwise do not order, if at the start of month 2 the inventory is 0 units, order 2 units, otherwise do not order; and do not order in month 3. This is an example of an (s, S) policy.

5. Foundations of infinite horizon models

This section introduces several optimality criteria for infinite horizon MDP's and discusses the relationship between them. It provides an introduction to the material in Sections 6–8. *We assume the data of the problem are stationary and S is either finite or countable.*

5.1. Policy valuation

In a stationary infinite horizon Markov decision process, each policy $\pi = (d_1, d_2, \ldots)$ induces a bivariate discrete time stochastic process; $\{[X_t^\pi, r(X_t^\pi, d_t(X_t^\pi))]; t = 1, 2, \ldots\}$. The first component X_t^π is the state of the system at time t and the second component is the reward received if the system is in X_t^π and decision rule $d_t(X_t^\pi)$ is used. When π is randomized, the marginal distribution of r depends on the distribution on the action space induced by the decision rule. When π is Markov, the above stochastic process is called a Markov reward process.

Unless otherwise noted rewards are bounded, i.e.,

$$\sup_{s\in S} \sup_{a\in A_s} |r(s, a)| = M < \infty . \tag{5.1}$$

To evaluate an infinite sequence of rewards or expected rewards requires a notion of convergence. Since S is assumed discrete, limits are always taken pointwise. When S is finite this is equivalent to the stronger notion of convergence in supremum norm however for more general S, this equivalence does not hold. Fortunately, convergence in norm is appropriate for computational results which are most widely applicable in the finite state case.

Several methods have been proposed to provide a value for a fixed policy in the infinite horizon case. They are now described.

(a) *The expected total reward* of policy π is given by

$$v^\pi(s) = E_{\pi,s}\left\{\sum_{t=1}^{\infty} r(X_t^\pi, d_t(X_t^\pi))\right\} . \tag{5.2}$$

In most applications, the series on the right hand side of (5.2) is divergent or not even well defined. When the limit exists, v^π satisfies

$$v^\pi(s) = \lim_{N\to\infty} v_N^\pi(s)$$

where v_N^π is defined by (4.2) with $r_{N+1} = 0$. Several special cases are distinguished for which this criterion is appropriate. These are introduced in the next subsection and are the subject of Section 7.

(b) *The expected discounted reward* of policy π is defined by

$$v_\lambda^\pi(s) = E_{\pi,s}\left\{\sum_{t=1}^{\infty} \lambda^{t-1} r(X_t^\pi, d_t(X_t^\pi))\right\} \tag{5.3}$$

for $0 \leq \lambda < 1$. Note that $v^\pi = \lim_{\lambda \uparrow 1} v_\lambda^\pi$ when the limit exists. Condition (5.1) ensures that $|v_\lambda^\pi(s)| < (1-\lambda)^{-1}M$ for all $s \in S$ and $\pi \in \Pi$.

In (5.3), the present value of the reward received in the first period has value r. This is equivalent to assuming that rewards are received at the beginning of the period immediately after the decision rule is applied.

(c) *the average reward or gain* of policy π is given by

$$g^\pi(s) = \lim_{N \to \infty} \frac{1}{N} v_N^\pi(s) \tag{5.4}$$

where $v_N^\pi(s)$ is defined by (4.2). In this case, it is not necessary that r_{N+1} equals zero since averaging removes the effect of the terminal reward. When the limit in (5.4) does not exist, the limit is replaced by the limit inferior to account for the worst possible limiting behavior under policy π.

The quantity g^π is the average expected reward per period obtained using policy π. Justification for calling it the gain is provided in Section 8.

5.2. *The expected total and discounted reward criteria*

A plausible objective in infinite horizon problems is to find a policy that maximizes v^π. However, even when r is bounded, v^π may be infinite or not well defined. To deal with this unsatisfactory situation, two approaches have been followed; to distinguish cases when this quantity is finite and to choose alternative optimality criteria that account for the rate at which v_N^π diverges.

When $v^\pi(s)$ is used to evaluate policies, investigators have distinguished the following cases;

(a) *The positive bounded case* (Blackwell, 1967): For each $s \in S$ and $a \in A_s$, $r(s, a) \geq 0$ and $\sup_{\pi \in \Pi} v^\pi(s) < +\infty$.

(b) *The negative case* (Strauch, 1966): For each $s \in S$ and $a \in A_s$, $r(s, a) \leq 0$ and there exists a $\pi \in \Pi$ with $v^\pi(s) > -\infty$.

(c) *The convergent case* (Hordijk, 1974): For each $s \in S$,

$$\sup_{\pi \in \Pi} E_{\pi,s}\left\{\sum_{t=1}^{\infty} |r(X_t^\pi, d_t(X_t^\pi))|\right\} < +\infty . \tag{5.5}$$

(d) *The discounted case* (Howard, 1960): In (5.3), $0 \leq \lambda < 1$.

These cases are related as follows. The positive case is a special case of the convergent case. If non-stationary rewards are considered and a transformed reward function is defined by $r_t(s, a) = \lambda^{t-1} r(s, a)$ or restrictions are placed on the transition probabilities, the discounted case can also be shown to be a special case of the convergent case. The negative case is distinct because it allows the quantity in (5.2) to be infinite.

Positive dynamic programming has been applied to optimal stopping problems and gambling problems (Ross 1983, pp. 76–83). Mathematically, the positive case is convenient because it ensures that $v^\pi(s)$ is well defined. It can arise in several different ways and is intimately related to the chain structure of the underlying Markov chains. For instance, if S is finite and under every

policy the chain ends up in a recurrent class in which the rewards are zero, then $v^{\pi}(s)$ is bounded.

Negative problems arise in the context of minimization of expected total costs when immediate costs are non-negative. Changing signs converts all costs to negative rewards and minimization to maximization. The condition that at least one policy has $v^{\pi}(s) > -\infty$ is equivalent to the existence of a policy with finite total expected cost. Such problems also arise in the context of minimizing the probability of reaching an undesirable state, minimizing the expected time to reach a desirable state (Demko and Hill, 1981) and optimal stopping with minimum expected total cost criterion.

Restricting rewards to be negative ensures that $v^{\pi}(s)$ is well defined, however it may be infinite for many policies. The restriction that at least one policy has $v^{\pi}(s)$ finite ensures that the expected total reward criteria is useful. Theoretically this problem is more challenging than the positive case because it permits policies with infinite rewards.

The discounted case is the most important in economic applications and the best understood theoretically and computationally. It will be studied in detail in Section 6. Discounting arises naturally in an economic context when the time values of the rewards are taken into account. The discount factor λ is the present value of one unit of currency received in the subsequent period so that v_{λ}^{π} is the expected total present value of the income stream obtained using policy π. Allowing λ to be non-constant leads to non-stationary problems.

Derman (1970, pp. 31–32) shows that discounting is equivalent to a problem with expected total reward criteria and a random termination time, τ, that is independent of the actions of the decision maker and geometrically distributed with parameter λ.

Generalizations of the discounted case include the transient case (Veinott, 1969, Hordijk, 1974, Pliska, 1978 and Whittle, 1983) and problems in which there is a single absorbing state and the expected time until absorption is bounded for all policies (Blackwell, 1962, Mine and Osaki, 1968 and van Dawen, 1986a).

5.3. *Optimality criteria*

The valuation expressions defined in Section 5.1 lead to natural notions of optimality. These are sometimes unsatisfactory and additional optimality criteria have been considered. Several are described below.

A policy π^* is said to be *total reward optimal* if

$$v^{\pi^*}(s) \geq v^{\pi}(s) \quad \text{for each } s \in S \text{ and all } \pi \in \Pi .$$

This concept is applicable in the cases distinguished in Section 5.2. In such cases the *value* of the MDP is given by

$$v^*(s) = \sup_{\pi \in \Pi} v^{\pi}(s) . \tag{5.6}$$

An optimal policy π^* exists when

$$v^{\pi^*}(s) = v^*(s) \quad \text{for all } s \in S .$$

An equivalent criterion is available when the expected discounted reward is used to evaluate policies. A policy π^* is said to be *discount-optimal* if the fixed λ, $0 \leq \lambda < 1$,

$$v_\lambda^{\pi^*}(s) \geq v_\lambda^{\pi}(s) \quad \text{for each } s \in S \text{ and all } \pi \in \Pi .$$

In such cases the value of the MDP is

$$v_\lambda^*(s) = \sup_{\pi \in \Pi} v_\lambda^{\pi}(s) . \tag{5.7}$$

A discount-optimal policy π^* exists whenever

$$v_\lambda^{\pi^*}(s) = v_\lambda^*(s) \quad \text{for all } s \in S .$$

A policy π^* is said to be *gain optimal* or *average optimal* if

$$g^{\pi^*}(s) \geq g^{\pi}(s) \quad \text{for each } s \in S \text{ and all } \pi \in \Pi .$$

The gain of the MDP is

$$g^*(s) = \sup_{\pi \in \Pi} g^{\pi}(s) . \tag{5.8}$$

When the limits defining $g^{\pi}(s)$ do not exist, two notions of gain optimality have been considered (Flynn, 1976, Federgruen, Hordijk and Tijms, 1979, and Federgruen, Schweitzer and Tijms, 1983). A policy π^* is said to be *average optimal in the strong sense* if its smallest limit point is as least as great as *any* limit point of any other policy. That is for each $s \in S$,

$$\begin{aligned} g^{\pi^*}(s) &= \liminf_{N \to \infty} \frac{1}{N} v_N^{\pi^*}(s) \\ &\geq \limsup_{N \to \infty} \frac{1}{N} v_N^{\pi}(s) \quad \text{for all } \pi \in \Pi . \end{aligned}$$

A policy π^* is said to be *average optimal in the weak sense* if its largest limit point is as least as great as *any* limit point of any other policy. That is, for each $s \in S$,

$$\begin{aligned} g^{\pi^*}(s) &= \limsup_{N \to \infty} \frac{1}{N} v_N^{\pi^*}(s) \\ &\geq \limsup_{N \to \infty} \frac{1}{N} v_N^{\pi}(s) \quad \text{for all } \pi \in \Pi . \end{aligned}$$

The following simple example motivates the optimality criteria described below.

Example 5.1. Let $S = \{1, 2, 3\}$ and suppose the action sets, rewards and transition probabilities are as follows. For $s = 1$, $A_s = \{a, b\}$, $r(s, a) = 1$, $r(s, b) = 0$ and $p(2|s, a) = 1$, $p(3|s, b) = 1$. For $s = 2$, $A_s = \{a\}$, $r(s, a) = 0$, $p(1|s, a) = 1$ and for $s = 3$, $A_s = \{b\}$, $r(s, b) = 1$ and $p(1|s, b) = 1$. Clearly the stationary policies which always use action a or b yield average rewards of $\frac{1}{2}$. It is easy to see that these policies are average optimal.

This example shows that the average reward criterion does not distinguish between policies which might have different appeal to the decision maker. Starting in state 1, policy a with reward stream $(1, 0, 1, 0, \ldots)$ is clearly superior to b with reward stream $(0, 1, 0, 1, \ldots)$ because it provides 1 unit in the first period which can be put to alternative use. Denardo and Miller (1968) called a criteria such as the average reward *unselective* because it depends only on the tail behavior of the sequence of rewards and does not distinguish policies with returns which differ only in a finite number of perods. Several more selective criteria have been proposed. They are based on either

(a) the comparative finite horizon expected total reward as the number of periods becomes large, or

(b) the comparative expected discounted reward as the discount factor λ increases to 1.

Those based on v_N^π are discussed first. Denardo and Miller (1968) called a policy π^* *overtaking optimality* if for each $s \in S$,

$$\liminf_{N\to\infty} v_N^{\pi^*}(s) - v_N^{\pi}(s) \geq 0 \quad \text{for all } \pi \in \Pi\,, \tag{5.9}$$

and showed with an example that this criterion is *overselective*, that is, there need not exist an optimal policy with respect to this criterion. The following criterion (Veinott, 1966) is less selective. A policy π^* is said to be *average overtaking optimal* if for each $s \in S$,

$$\liminf_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \{v_n^{\pi^*}(s) - v_n^{\pi}(s) \geq 0\} \quad \text{for all } \pi \in \Pi\,. \tag{5.10}$$

Note that neither of these criteria requires that $\lim_{N\to\infty} v_N^\pi$ exists.

Sladky (1974) generalized the average overtaking optimal criterion as follows. Let $v_{N,1}^\pi$ be the expected total return for policy π up to period N and define $v_{N,n}^\pi$ recursively for $n \geq 1$ by

$$v_{N,n}^{\pi} = \sum_{j=1}^{N} v_{j,n-1}^{\pi}\,.$$

A policy $\pi^* \in \Pi$ is said to be *n-average optimal for* $n = -1, 0, 1, \ldots$ if

$$\liminf_{N\to\infty} N^{-1}\{v^{\pi^*}_{N,n+2} - v^{\pi}_{N,n+2}\} \geq 0$$

for all $\pi \in \Pi$. Observe that 0-average optimality corresponds to average overtaking optimality.

Several optimality criteria have been based on the asymptotic behavior of the total discounted reward. The following important concept of optimality was first proposed by Blackwell (1962). A policy π^* is said to be 1-*optimal* if for each $s \in S$ there exists a $\lambda^*(s)$ such that

$$v^{\pi^*}_{\lambda}(s) - v^{\pi}_{\lambda}(s) \geq 0 \quad \text{for all } \pi \in \Pi \text{ for } \lambda^*(s) \leq \lambda < 1\,. \tag{5.11}$$

Such policies are now referred to as *Blackwell optimal*. Blackwell proposed this criterion in the context of S finite in which case $\lambda^* = \sup_{s\in S} \lambda^*(s)$ is attained. In countable state problems this supremum might equal 1. Dekker (1985) distinguishes cases when $\lambda^* < 1$ as *strongly Blackwell optimal*.

Veinott (1969) generalized Blackwell optimality by proposing the following family of sensitive optimality criteria. A policy π^* is said to be *n-discount optimal* if for each $s \in S$,

$$\liminf_{\lambda\uparrow 1} (1-\lambda)^{-n}[v^{\pi^*}_{\lambda}(s) - v^{\pi}_{\lambda}(s)] \geq 0 \quad \text{for all } \pi \in \Pi\,. \tag{5.12}$$

This criterion unified several optimality criteria based on the expected discounted reward including average or gain optimality, bias and Blackwell optimality. It has been shown that (-1)-discount optimality is equivalent to average optimality, 0-discount optimality is equivalent to bias optimality and ∞-discount optimality is equivalent to Blackwell optimality. These equivalences are discussed in more detail in Section 8.9 where the Laurent series expansion on which these are based is presented.

Blackwell optimality is the most selective of the n-discount optimality criteria as it implies n-discount optimality for all finite n. It implies gain and bias optimality. In general, n-discount optimality implies m-discount optimality for all $m < n$ so that bias optimality $(n = 0)$ is more selective than gain optimality $(n = -1)$.

Optimality criteria have also been based on the asymptotic behavior of policies for finite horizon problems as the horizon gets large. Morton (1978) calls a policy *forecast horizon optimal* if it is the pointwise limit of optimal policies for finite horizon problems. Since the limit need not exist, Hopp, Bean and Smith (1988) have introduced a weaker criteria, periodic forecast horizon optimality. A policy is said to be *periodic forecast horizon optimal* if it is the limit of a subsequence of optimal policies for finite problems in an appropriate metric. These two criteria are of particular importance in nonstationary problems.

When the assumption that $\Sigma_{j\in S} p(j|s,a) \leq 1$ is not satisfied the above criteria are inappropriate. Rothblum (1984) showed that problems for which $\Sigma_{j\in S} p(j|s,a) > 1$ include Markov decision processes with multiplicative utilities (Howard and Matheson, 1972) and controlled branching processes (Mandl, 1967, Pliska, 1976). Optimality criteria in this case are based on choosing policies which maximize the spectral radius (Bellman, 1957, p. 329). More general optimality criteria have been proposed by Rothblum and Veinott (1975).

6. Discounted Markov decision problems

This section analyzes infinite horizon MDP's under the expected total discounted cost optimality criterion. The optimality equation is introduced and its fundamental role in Markov decision process theory and computation is demonstrated. Several algorithms for solving the optimality equation are presented and discussed; the section concludes with a numerical example and a discussion of discounted Markov decision problems with unbounded rewards. Throughout this section it is assumed that the problem is stationary and as before, S is assumed to be discrete.

6.1. The optimality equation

The optimality or Bellman equation

$$v(s) = \sup_{a\in A_s} \left\{ r(s,a) + \sum_{j\in S} \lambda p(j|s,a) v(j) \right\}, \quad s \in S, \tag{6.1}$$

plays a key role in the theory of Markov decision problems. In vector notation it can be written as

$$v = \sup_{d\in D} \{ r_d + \lambda P_d v \}. \tag{6.2}$$

The supremum in (6.2) is understood to be taken componentwise so that (6.2) is shorthand notation for (6.1). A formulation based on using (6.2) as the optimality equation would allow decision sets which incorporate constraints across states, however, the consequences of such a formulation will not be explored in this chapter.

When the supremum on the right hand side of (6.1) or (6.2) is attained, for example when A_s is finite, 'sup' will be replaced by 'max'.

Define the operator $T: V \to V$ by

$$Tv \equiv \sup_{d\in D} \{ r_d + \lambda P_d v \} \tag{6.3}$$

and for each $d \in D$ define the operator $T_d : V \to V$ by

$$T_d v \equiv r_d + \lambda P_d v \,. \tag{6.4}$$

Comparing (6.2) and (6.3) shows that the optimality equation can be expressed as $v = Tv$. Thus, a solution v of the optimality equation is a *fixed point* of T. This observation will be fundamental to Section 6.2.

The main properties of the optimality equation to be presented below include:

(1) If a solution of the optimality equation exists, it equals the value of the discounted MDP (Theorem 6.3).

(2) The value of the discounted MDP satisfies the optimality equation (Corollary 6.7).

(3) The solution of the optimality equation is unique (Corollary 6.3).

(4) The optimality equation characterizes optimal policies (Theorem 6.4).

A recursive equation to compute the expected total discounted reward of a fixed policy π is now developed. Let $\pi = (d_1, d_2, \ldots)$ be an arbitrary policy. Its expected total discounted reward $v_\lambda^\pi(s)$ was given by (5.3). The expectation in (5.3) can be expressed in terms of transition probabilities as follows

$$\begin{aligned} v_\lambda^\pi &= \sum_{n=1}^{\infty} \lambda^{n-1} P_\pi^{n-1} r_{d_n} \qquad (6.5) \\ &= r_{d_1} + \lambda P_{d_1} r_{d_2} + \lambda^2 P_{d_1} P_{d_2} r_{d_3} + \cdots \\ &= r_{d_1} + \lambda P_{d_1} (r_{d_2} + \lambda P_{d_2} r_{d_3} + \lambda^2 P_{d_2} P_{d_3} r_{d_4} + \cdots) \\ &= r_{d_1} + \lambda P_{d_1} v_\lambda^{\pi'} \qquad (6.6) \end{aligned}$$

where $\pi' = (d_2, d_3, \ldots)$ and the limit implicit in (6.5) is componentwise. The relationship in (6.6) holds for arbitrary policies, however if π is stationary, $\pi' = \pi$. Denote the stationary policy $\pi = (d, d, \ldots)$ by d. Rewriting the above relationship yields

$$v_\lambda^d = r_d + \lambda P_d v_\lambda^d \equiv T_d v_\lambda^d \,. \tag{6.7}$$

This equation shows that v_λ^d is a solution of $v = r_d + \lambda P_d v$. This is extended in the following.

Proposition 6.1. *For any stationary policy d, v_λ^d is the unique solution of*

$$v = r_d + \lambda P_d v = T_d v \,. \tag{6.8}$$

Further,

$$v_\lambda^d = (I - \lambda P_d)^{-1} r_d = \sum_{n=1}^{\infty} \lambda^{n-1} P_d^{n-1} r_d \,. \tag{6.9}$$

The following is the fundamental result about the optimality equation and its solutions. A proof appears in Blackwell (1962).

Theorem 6.2. *Suppose* $v \in V$ *satisfies*

$$v \geq (\leq) \sup_{d \in D} \{r_d + \lambda P_d v\} . \tag{6.10}$$

Then $v \geq (\leq) v_\lambda^*$.

Since any solution of the optimality equation must satisfy both of the above inequalities, it is equal to the optimal value function and consequently is unique. Alternatively, uniqueness can be established using the contraction mapping methods of Section 6.3.

Theorem 6.3. *If the equation* $v = Tv$ *has a solution, it is unique and equals* v_λ^*.

Thus, the value function is a solution of the optimality equation and if $v \geq (\leq) Tv$, v is an upper (lower) bound for v_λ^*. These inequalities are fundamental to demonstrating the convergence of algorithms for solving the optimality equation and developing computable bounds for v_λ^*.

6.2. The existence of optimal policies

In this section it is shown that existence of a solution to the optimality equation, together with attainment of the supremum in (6.2) is sufficient for the existence of a deterministic stationary optimal policy. Assume first that the maximum is attained in (6.2) so the optimality equation is given by

$$v = \max_{d \in D} \{r_d + \lambda P_d v\} = Tv . \tag{6.11}$$

The following terminology will be used frequently throughout this chapter. For $v \in V$, a decision rule d_v is said to be *v-improving* if

$$d_v = \arg\max_{d \in D} \{r_d + \lambda P_d v\} \tag{6.12}$$

or equivalently,

$$T_{d_v} v = r_{d_v} + \lambda P_{d_v} v = \max_{d \in D} \{r_d + \lambda P_d v\} = Tv . \tag{6.13}$$

Using component notation, d_v is v-improving if for all $s \in S$,

$$r(s, d_v(s)) + \sum_{j \in S} \lambda p(j \mid s, d_v(s)) v(s)$$

$$= \max_{a \in A_s} \left\{ r(s, a) + \sum_{j \in S} \lambda p(j \mid s, a) v(j) \right\} .$$

Theorem 6.4. *Suppose* $v^* \in V$ *satisfies* (6.11) *and* $d^* \in D$ *is* v^**-improving. Then*

$$v_\lambda^{d^*} = \max_{\pi \in \Pi} v_\lambda^\pi .$$

A v^*-improving decision rule is often said to be *conserving* (Dubins and Savage, 1965). In other words, d^* is conserving if

$$d^* = \arg\max_{d \in D} \{r_d + \lambda P_d v^*\} \tag{6.14}$$

or

$$T_{d^*} v^* = Tv^* . \tag{6.15}$$

The following important corollary is a restatement of Theorem 6.4 in this terminology.

Corollary 6.5. *Suppose a conserving decision rule* d^* *exists. Then the deterministic Markov stationary policy which uses* d^* *every period is optimal in the class of all policies.*

Conserving decision rules exist whenever the optimality equation has a solution and the maximum in (6.11) is attained. It will be shown in Section 6.3 that the optimality equation *always* has a unique solution when $0 \leq \lambda < 1$, so that in the discounted case, attainment of the maximum in (6.11) is equivalent to the existence of optimal policies in Π_D. In such cases there is no need to consider history dependent or randomized policies; it is sufficient to optimize over the class of stationary deterministic Markov policies.

In terms of the data of the problem, sufficient conditions for the attainment of the maximum and the existence of optimal policies in Π_D include:

(a) A_s is finite for each $s \in S$,

(b) for each $s \in S$, A_s is compact, $r(s, a)$ is upper semi-continuous in a and for each $j \in S$, $p(j \mid s, a)$ is continuous in a (Maitra, 1968).

Blackwell (1965) provided an example with S finite (a singleton) and A countable in which there does not exist an optimal policy, and an example in which there does not exist an ε-optimal policy for sufficiently small $\varepsilon > 0$. That is, there is no policy such that

$$v_\lambda^\pi(s) > v^*(s) - \varepsilon \quad \text{for all } s \in S .$$

In that paper he provided conditions under which ε-optimal stationary policies exist.

6.3. Value iteration and its variants

This section shows how the theory of contraction mappings on Banach spaces is used to demonstrate the existence of a unique solution to the optimality equation, and to analyze the convergence of the value iteration method for solving the optimality equation. Shapley (1953) introduced the basic ideas of value iteration in the context of stochastic games; the use of contraction mappings for MDP's is usually attributed to Denardo (1967); a good summary and some extensions appear in Federgruen and Schweitzer (1978).

6.3.1. Theoretical considerations

The results below are based on Denardo's (1967) observations that the operator T defined in (6.3) is a contraction mapping on V, the space of bounded real valued functions on S with supremum norm, and that a solution of the optimality equation is a fixed point of T. The operator $T: V \rightarrow V$ is a *contraction mapping* because

$$\|Tu - Tv\| \leq \lambda \|u - v\| \quad \text{for all } u, v \in V . \tag{6.16}$$

where λ satisfies $0 \leq \lambda < 1$.

Since V is a complete normed linear space or Banach Space ($L^{\infty}(S)$) the Banach Fixed Point Theorem (Liusternik and Sobolev, 1961) can be applied to obtain the following important result.

Theorem 6.6. *The operator T has a unique fixed point $v^* \in V$ and for every $v^0 \in V$, the sequence $\{v^n\}$ defined by $v^{n+1} = Tv^n$ converges in norm to v^*.*

The convergence in Theorem 6.6 is in the norm sense, that is

$$\lim_{n \to \infty} \|v^n - v^*\| = 0$$

where $\|\cdot\|$ is defined in Section 2.2. When S is finite this is equivalent to pointwise convergence but in the countable case, it is a considerably stronger result. The following corollaries are the main applications of this theorem in the discounted case. The first is an application of the above theorem when the supremum is attained in (6.2) so that

$$Tv = \max_{d \in D} \{r_d + \lambda P_d v\} .$$

Corollary 6.7. *Suppose the supremum in* (6.2) *is attained. Then*

(a) *v_{λ}^* is the unique solution to the optimality equation,*

(b) *there exist conserving decision rules, and*

(c) *the stationary deterministic Markov policy that uses any conserving decision rule is optimal among the class of all policies.*

When the supremum is not attained, only ε-optimal policies are possible. This result is summarized as follows.

Corollary 6.8. *Suppose the optimality equation is given by* (6.2). *Then*
(a) v_λ^* *is the unique solution of* (6.2) *and*
(b) *for* $\varepsilon > 0$, *the deterministic, stationary policy which uses the decision rule* d^ε *defined by*

$$r_{d^\varepsilon} + \lambda P_{d^\varepsilon} v_\lambda^* + \varepsilon(1-\lambda) \geqslant \sup_{d \in D} \{r_d + \lambda P_d v_\lambda^*\} \tag{6.17}$$

is ε*-optimal among the class of all policies.*

6.3.2. The value iteration algorithm

Another important consequence of Theorem 6.6 is the convergence of the value iteration algorithm for solving the optimality equation. The value iteration algorithm finds a stationary policy that is ε-optimal. In general, ε-optimality occurs if either
(i) the supremum is (6.2) is not attained, or
(ii) the algorithm is terminated in a finite number of iterations.

It will be assumed throughout this section that the supremum in (6.2) *is attained so that the source of* ε*-optimality is the finite termination of the algorithm.* Under this assumption $Tv = \max_{d \in D} \{r_d + \lambda P_d v\}$. Otherwise the stopping criterion in step 3 below requires modification to account for the two sources of ε-optimality.

The Value Iteration Algorithm.
1. Select $v^0 \in V$, specify $\varepsilon > 0$ and set $n = 0$.
2. For each $s \in S$, compute $v^{n+1}(s)$ by

$$v^{n+1}(s) = \max_{a \in A_s} \left\{ r(s, a) + \sum_{j \in S} \lambda p(j \mid s, a) v^n(j) \right\}. \tag{6.18}$$

3. If $\|v^{n+1} - v^n\| < \varepsilon(1-\lambda)/2\lambda$ go to step 4. Otherwise increment n by 1 and return to step 2.
4. For each $s \in S$, set

$$d^\varepsilon(s) = \arg\max_{a \in A_s} \left\{ r(s, a) + \sum_{j \in S} \lambda p(j \mid s, a) v^{n+1}(j) \right\}. \tag{6.19}$$

and stop. If the arg max in (6.19) is not unique, any action achieving this maximum can be selected.

The main step in the algorithm is 2 which gives the recursion $v^{n+1} = Tv^n$ in component notation. Theorem 6.6 guarantees the convergence of the algorithm to the optimal value function and that the stopping criterion is satisfied in

finitely many iterations. When the stopping criterion in step 3 is met, the stationary policy corresponding to a v^{n+1}-improving decision rule is ε-optimal. Improved stopping rules are discussed in Section 6.7.

Theorem 6.6 ensures convergence of value iteration for arbitrary state spaces provided that the appropriate norm is selected so that the value functions and norm are a Banach space. This means that value iteration will converge in norm if S is finite, countable, compact or Borel. Unfortunately direct implementation of the maximization in (6.20) is only practical when S is finite. For more general state spaces, the maximization can only be carried out by using special structure of the rewards, transition probabilities and value functions to determine the structure of maximizing decision rules. If it can be established that a property of v^n is preserved by induction, for example unimodality, and that this property ensures that the optimizing decision rule is of a certain form, i.e., control limit, then if the property of v^n holds in the limit, as a consequence of Corollary 6.7 there exists an optimal stationary policy with the special structure. This idea has been used extensively in inventory theory (Chapter 12), replacement theory and queueing control to determine the structure of optimal policies for infinite horizon problems.

The value iteration algorithm as defined above terminates in a finite number of iterations when the stopping criteria in step 3 is satisfied. Consequently, there is no guarantee that the resulting policy is optimal. In special cases, action elimination procedures discussed in Section 6.7.3 can be used to ensure termination with an optimal policy. Note that, the policy determined in step 4 is ε-optimal in the norm sense, that is

$$\|v_\lambda^{d^\varepsilon} - v_\lambda^*\| < \varepsilon .$$

The above algorithm is also called *successive approximation*, *backward induction* or *dynamic programming*.

Results about the convergence of value iteration are summarized in the following theorem.

Theorem 6.9. *Let $v^0 \in V$ be arbitrary. Then*

(a) *the iteratives of value iteration converge in norm to v_λ^* and*

(b) *the algorithm terminates in a finite number of iterates with an ε-optimal policy determined by* (6.19).

Some further properties of the iterates of the algorithm are that

$$\|v^{n+1} - v_\lambda^*\| = \|Tv^n - Tv_\lambda^*\| \leq \lambda \|v^n - v_\lambda^*\| . \tag{6.20}$$

This inequality means that the convergence rate of the algorithm is linear. This is often referred to as geometric convergence because iterating (6.19) yields

$$\|v^n - v_\lambda^*\| \leq \lambda^n \|v^0 - v_\lambda^*\| . \tag{6.21}$$

When λ is close to one, the above bounds suggest that the convergence of this algorithm will be quite slow. The subsequent subsections discuss other more efficient methods for solving discounted MDP's.

Using standard arguments, the following error bound for the iterates of value iteration can be obtained;

$$\|v^n - v_\lambda^*\| \leq \frac{\lambda^n}{1-\lambda} \|v^1 - v^0\| . \tag{6.22}$$

In practice, the error bound below is more useful:

$$\|v_\lambda^{d^\varepsilon} - v_\lambda^*\| \leq \frac{2\lambda^n}{1-\lambda} \|v^1 - v^0\| . \tag{6.23}$$

By specifying ε a priori and performing one value iteration step, (6.23) can be used to estimate the number of additional iterations required to obtain the desired precision.

6.3.3. Variants of value iteration

One of the major disadvantages of using value iteration is that it converges geometrically at rate λ. If λ is close to 1, solution by this method would require a large number of iterations. Several authors, including Morton (1971) and Morton and Wecker (1977), have suggested instead that value iteration be normalized by either

(a) subtracting an appropriate vector, or

(b) using relative differences at sucessive iterates.

In either case the normalized iterates $\{w^n\}$ satisfy

$$\|w^{n+1} - w_\lambda^*\| \leq \lambda\alpha_n \|w^n - w_\lambda^*\|$$

where w_λ^* is the normalized optimal value function and α_n is the modulus of the subdominant eigenvalue (the second largest eigenvalue in modulus) of the transition matrix of the v^n-improving decision rule. The advantage of this approach is that if the subdominant eigenvalues for most policies (especially the optimal one) are considerably smaller than 1, the rate of convergence of this normalized or *relative value iteration* will be considerably faster than that for value iteration.

When the transition matrices of all policies are irreducible, relative value iteration can be implemented by selecting an arbitrary state s_0, defining w^0 for each $s \in S$ by

$$w^0(s) = v^0(s) - v^0(s_0)$$

and iterating according to

$$w^{n+1}(s) = Tw^n(s) - Tw^n(s_0) \quad \text{for } s \in S .$$

When the policies have more general chain structure, a different normalization which requires identification of recurrent classes can achieve this improved rate of convergence.

Another modification that will accelerate computations is to use the Gauss–Seidel variant of value iteration (Hastings, 1969). In it, updated values of $v^{n+1}(s)$ are substituted into the recursive equation as soon as they are evaluated. Suppose that the states are labelled $s_1, s_2, \ldots, s_N$ and are evaluated in order of their subscripts. Then the Gauss–Seidel iterative recursion is

$$v^{n+1}(s_j) = \max_{a \in A_{s_j}} \left\{ r(s, a) + \lambda \left[\sum_{i<j} p(s_i \mid s_j, a) v^{n+1}(s_i) + \sum_{i \geq j} p(s_i \mid s_j, a) v^n(s_i) \right] \right\}.$$

As a consequence of a result in Senata (1981, pp. 41–44), the rate of convergence of Gauss–Seidel iteration will be greater than that of value iteration. This has been observed empirically by several authors (cf. Porteus and Totten, 1978).

6.3.4. Convergence of policies and turnpike theory

Results in the preceding sections are concerned with the behavior of the sequence of values $\{v^n\}$ obtained from the value iteration algorithm. Even if v^n is close to v_λ^*, there is no guarantee that the decision rule that attains the maximum in (6.14) is similar to the optimal decision rule. Brown (1965), Shapiro (1968) and Federgruen and Schweitzer (1978) investigate the asymptotic properties of the sets of v^n-improving decision rules.

In practice, infinite horizon models are usually approximations to finite horizon models with many stages. A question of practical concern is, "Under what conditions is the optimal policy for the infinite-horizon model optimal for the finite horizon model?" An answer to this question is provided through turnpike or planning horizon theory. One such result is the following (Shapiro, 1968).

Theorem 6.10. *There exists an N^* such that for any $n \geq N^*$, the optimal decision in a finite horizon problem when there are n periods remaining is in D^*, the set of optimal stationary policies for the infinite horizon problem.*

A consequence of this result is that if there is a unique optimal stationary policy for the infinite horizon problem, then it is optimal to use the corresponding decision rule in the first $n - N^*$ periods of a problem with finite horizon $n > N^*$. The optimal policy in the remaining N^* periods must be determined by backward induction. The optimal infinite-horizon strategy is referred to as the turnpike and it is reached after travelling N^* periods on the nonstationary 'side roads'.

Another interpretation of this result is that it is optimal to use any $d \in D^*$ for the first decision in a finite horizon problem in which the horizon is known to exceed N^*. Thus it is not necessary to know the horizon specifically but only that it exceeds N^*. For this reason, N^* is often called a *planning horizon*. A bound on N^* is given in Denardo (1982, p. 176). The concept of planning horizons has been extended to non-stationary models by Hopp, Bean and Smith (1988).

6.4. Policy iteration

Policy iteration or *approximation in policy space* was introduced by Bellman (1957) and independently by Howard (1960). It is a highly efficient procedure for solving Markov decision problems. This section discusses this algorithm for finite state problems with finite and compact action sets. The maximum in (6.2) is assumed to be attained.

6.4.1. The policy iteration algorithm

The algorithm is as follows.

The Policy Iteration Algorithm (Howard, 1960).

1. Set $n = 0$ and select an arbitrary decision rule $d_0 \in D$.
2. (Policy evaluation) obtain v_{d_n} by solving

$$(I - \lambda P_{d_n})v = r_{d_n}. \tag{6.24}$$

3. (Policy improvement) Choose d_{n+1} to satisfy

$$r_{d_{n+1}} + \lambda P_{d_{n+1}} v_{d_n} = \max_{d \in D} \{r_d + \lambda P_d v_{d_n}\} \tag{6.25}$$

and set $d_{n+1} = d_n$ if possible.

4. If $d_{n+1} = d_n$, stop and set $d^* = d_n$. Otherwise increment n by 1 and return to 2.

The above algorithm yields a sequence of policies $\{d_n\}$ and value functions $\{v_{d_n}\}$. It terminates when the maximixing policy in step 3 repeats. This occurs with certainty in a finite number of iterations in finite state and action problems but not in compact action problems for which the number of stationary policies is infinite.

Step 2 is called the policy evaluation step because in it, (6.25) is solved to obtain the expected discounted reward of stationary policy d_n. This equation is usually solved by Gauss elimination. In step 3, a v_{d_n}-improving decision rule is selected. Since the decision rule is not necessarily unique, the condition that $d_{n+1} = d_n$ is included to avoid cycling and ensure termination.

To carry out step 3, the set of all v_{d_n}-improving decision rules is required before selecting a particular decision rule. An alternative specification of the

algorithm would retain the entire set of v_{d_n}-improving decision rules and terminate when it repeats. This modification is unnecessary since at termination, $v^n = v_\lambda^*$, so that all conserving decision rules are available.

Alternatively, one might implement step 3 by just finding a decision rule d_{n+1} with the property that

$$r_{d_{n+1}} + \lambda P_{d_{n+1}} v_{d_n} \geqslant r_{d_n} + \lambda P_{d_n} v_{d_n}$$

with strict inequality for at least one component. If this specification is used, the algorithm will still converge in the finite action case, but at a much slower rate than using the implementation in step 3. If the set of actions is compact, convergence to v_λ^* is not guaranteed.

6.4.2. The finite action case

This section discusses convergence of the policy iteration algorithm in the finite state and action case. Fundamental is the result that $(I - \lambda P_d)^{-1}$ is a positive matrix, i.e., if $u \geqslant 0$, $(I - \lambda P_d)^{-1} u \geqslant 0$. Consequently the values at successive iterations of policy iteration are monotone non-decreasing.

Proposition 6.11. *Suppose d_{n+1} is v_{d_n}-improving. Then*

$$v_{d_{n+1}} \geqslant v_{d_n}.$$

Since there are only finitely many deterministic stationary policies, the algorithm must terminate in a finite number of iterations. At termination, $d_{n+1} = d_n$, so that

$$v_{d_n} = r_{d_{n+1}} + \lambda P_{d_{n+1}} v_{d_n} = \max_{d \in D} \{r_d + \lambda P_d v_{d_n}\}.$$

Thus v_{d_n} solves the optimality equation and d^* is conserving. Applying Theorems 6.3 and 6.4 gives the following important result.

Theorem 6.12. *Suppose S is finite and for each $s \in S$, A_s is finite. Then the policy iteration algorithm terminates in a finite number of iterations and the policy d^* is discount optimal.*

6.4.3. The compact action case

When the decision set is not finite, the argument used to prove Theorem 6.12 is no longer valid since there is no guarantee that the stopping criterion in Step 4 will ever be satisfied. In such cases, an analytic approach can be used to demonstrate convergence. Drawing a parallel to the analysis in Section 6.2, other issues of concern are:

(a) What is the consequence of initiating the algorithm at step 3 (instead of at step 1) with a v^0 that is not the return of some policy?

(b) What is the rate of convergence of the algorithm?

The development relies on the theoretical foundation of Puterman and Brumele (1978, 1979) and Puterman and Shin (1978). Define the operator $B : V \to V$ by

$$Bv \equiv \max_{d \in D} \{r_d + (\lambda P_d - I)v\} . \tag{6.26}$$

Then the optimality equation (6.13) can be expressed as

$$Bv = 0 \tag{6.27}$$

and solving the MDP can be regarded as finding a zero of B instead of a fixed point of T. The key point is that policy iteration is equivalent to using Kantorovich's generalization of Newton's method for finding a zero of B (Kantorovich, 1952).

For $v \in V$ define the set of decision rules D_v to be all $d_v \in D$ satisfying

$$d_v = \arg\max_{d \in D} \{r_d + (\lambda P_d - I)v\} . \tag{6.28}$$

Note that the I is (6.28) does not effect the maximization so that D_v is the set of v-improving decision rules.

Proposition 6.13. *For $u, v \in V$ and any $d_v \in D_v$,*

$$Bu \geqslant Bv + (\lambda P_{d_v} - I)(u - v) . \tag{6.29}$$

This result follows easily from the definitions of the quantities in (6.29). It is called the 'support inequality' and is a vector space generalization of the gradient inequality which defines convex functions in R^n. Thus in a generalized sense, the operator B is 'convex' and $\lambda P_{d_v} - I$ is the 'support' of B at v.

Figure 6.1 illustrates the convexity and construction of Bv. In the situation depicted, there are four policies. For each, the function $r_i + (P_i - I)v$ is given. At each $v \in V$, Bv is the maximum of these functions. With the exception of $r_4 + (P_4 - I)v$, all are supports for some v in the illustrated portion of V. Note Bv is convex.

The following proposition provides a closed form representation for the sequence of values generated by policy iteration and is fundamental to this analysis.

Proposition 6.14. *Suppose the sequence $\{v^n\}$ is obtained from the policy iteration algorithm. Then for any $d_{v^n} \in D_{v^n}$,*

$$v^{n+1} = v^n - (\lambda P_{d_{v^n}} - I)^{-1} Bv^n . \tag{6.30}$$

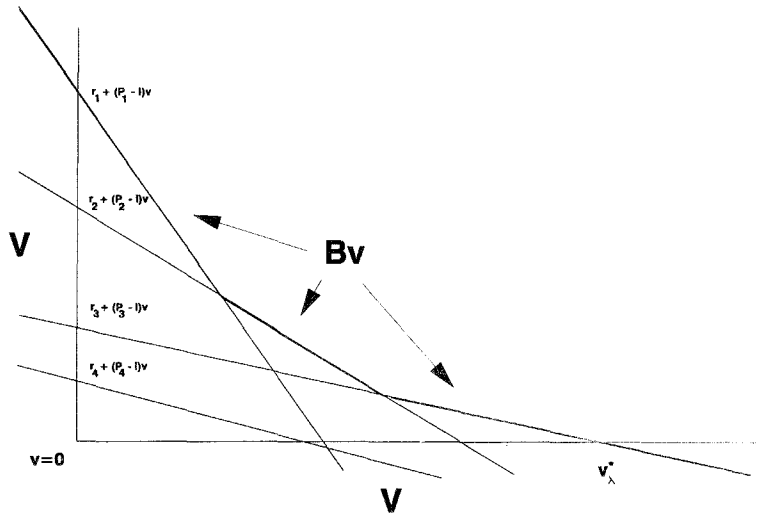

Fig. 6.1. Construction of Bv.

Noting the analogy between the support in v and the derivative in R^1, expression (6.30) is a vector space version of Newton's method. Note also that d_{v^n} in (6.30) corresponds to the decision rule d_{n+1} obtained in step 3 of the policy iteration algorithm. In R^1, if a function $f(x)$ is convex decreasing and has a zero, then starting Newton's method at a point at which the function is positive ensures that the iterates converge monotonically to the zero. This observation is the basis for Theorem 6.15 below. A proof can be based on comparing the iterates of policy iteration to those of value iteration and showing that if policy iteration and value iteration begin at the same point, then the iterates of policy iteration are always bounded below by those of value iteration and above by v^*_λ, the solution of the optimality equation which is assumed to exist. For a more general approach see Puterman and Brumelle (1978, 1979).

Theorem 6.15. *Suppose* $Bv^0 \geq 0$ *and there exists a unique* v^* *such that* $Bv^* = 0$. *Then the sequence of iterates* $\{v^n\}$ *defined by* (6.30) *converges monotonically and in norm to the zero of* B, v^*.

Since v^1 is the expected total discounted reward of a v^0-improving decision rule, $Bv^1 \geq 0$, and the above conclusions hold for arbitrary v^0. In terms of the policy iteration algorithm, this means that:

(a) The sequence of values generated by policy iteration converges monotonically and in norm to the solution of the optimality equation.

(b) If the policy iteration algorithm is initiated in step 3 with an arbitrary v^0, the conclusion in (a) holds.

These results are not restricted to the finite state, compact action case. They require only that the maximum be attained in (6.2). Consequently, they apply in the important case when S is countable, A_s is compact, and $p(j|s, a)$ and $r(s, a)$ are continuous in a for each $s \in S$ with V equal to the family of bounded functions of S with supremum norm.

Implementation of policy iteration when the set of decision rules is not finite requires a stopping rule to guarantee finite convergence. That is, step 3 of the value iteration algorithm can be incorporated to ensure finite convergence to an ε-optimal policy. For more details on this point, see Section 6.5. As in the case of value iteration, when S is not finite, step 3 cannot be implemented unless special structure is available.

6.4.4. Rates of convergence

When S is discrete, the norm of a matrix H given by

$$\|H\| = \sup_{s \in S} \sum_{j \in S} |h_{sj}| .$$

If S is finite, this supremum is attained.

The following is the main result on convergence rates (Puterman and Brumelle, 1979).

Theorem 6.16. *Suppose* $\{v^n\}$ *is generated by policy iteration and there exists a* K, $0 < K < \infty$, *such that*

$$\|P_{d_v} - P_{d_u}\| < K\|v - u\| \quad \textit{for all } u, v \in V . \tag{6.31}$$

Then

$$\|v^{n+1} - v_\lambda^*\| \leq \frac{K\lambda}{1-\lambda} \|v^n - v_\lambda^*\|^2 . \tag{6.32}$$

This theorem says that if (6.31) holds, policy iteration converges *quadratically* to the optimal value function. This accounts for the fast convergence of policy iteration is practice. In contrast, value iteration and its variants converge linearly.

In terms of the data of the problem, sufficient conditions for (6.31) to hold are that for each $s \in S$:

(a) A_s is compact and convex,

(b) $p(j|s, a)$ is affine in a, and

(c) $r(s, a)$ is strictly concave and twice continuously differentiable in a.

When A_s is finite, (6.31) need not hold because P_{d_v} is not unique at several $v \in V$; however, if a rule such as that in step 3 of the policy iteration algorithm is used to break ties, the algorithm provides a unique support at each v. Thus convergence will be quadratic although K might be large. Other conditions which imply (6.31) can be derived from selection theorems in Fleming and Rishel (1975).

Corollary 6.17. *Suppose* $\{v^n\}$ *is generated by policy iteration and*

$$\lim_{n\to\infty} \|P_{d_{v^n}} - P_{d_{v_\lambda^*}}\| = 0 . \tag{6.33}$$

Then

$$\lim_{n\to\infty} \frac{\|v^{n+1} - v_\lambda^*\|}{\|v^n - v_\lambda^*\|} = 0 . \tag{6.34}$$

If a sequence satisfies (6.34) its convergence is said to be *superlinear* (cf. Ortega and Rheinboldt, 1970). This corollary says that if (6.33) holds, the sequence generated by policy iteration converges superlinearly to v_λ^*. This means that the convergence is asmptotically faster than geometric convergence with any convergence rate constant. Thus under (6.33), policy iteration will attain the same degree of precision in fewer iterations than value iteration. In terms of the data of the problem, if the conditions for quadratic convergence above are relaxed to require only that $r(s, a)$ be strictly concave in a, then (6.34) holds.

Puterman and Brumelle (1979) also have developed error bounds for $\|v^n - v_\lambda^*\|$ in terms of $\|v^1 - v^0\|$, but evaluating them is tedious.

6.5. *Modified policy iteration*

The evaluation step of the policy iteration algorithm is usually implemented by solving the linear system

$$(I - \lambda P_{d_n})v = r_{d_n} \tag{6.35}$$

by Gaussian elimination. This requires $\frac{1}{3}M^3$ multiplications and divisions, where M is the number of states. For large M, exact solution of (6.35) can be computationally prohibitive. But, it is not necessary to determine this quantity precisely to identify an improved policy (see Figure 6.2). An approximate solution to the above equation can be obtained by using successive approximations with a fixed policy.

Morton (1971) suggested this approach: it was formalized by van Nunen (1976a) and Puterman and Shin (1978). Van Nunen called this algorithm 'value oriented successive approximations' and regarded it as a variant of value iteration in which the same decision rule is used for several evaluations.

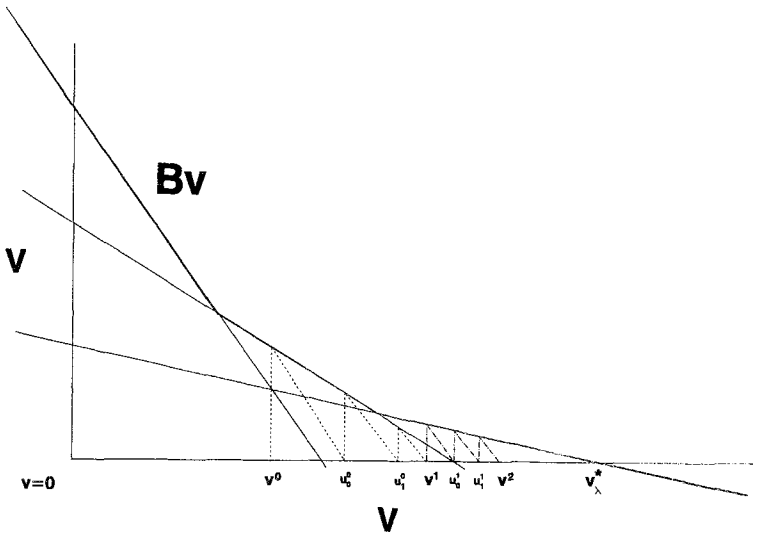

Fig. 6.2. Illustration of modified policy iteration of order 2.

Puterman and Shin called it 'modified policy iteration' and viewed it as a variant of policy iteration in which policy evaluation is implemented iteratively. Their approach is the basis for this section.

The Modified Policy Iteration Algorithm (MPI) of Order m.

1. Select $v^0 \in V$, specify $\varepsilon > 0$ and set $n = 0$.
2. (Policy improvement) Choose d_{n+1} to be any v^n-improving decision rule.
3. (Policy evaluation)
   (a) Set $k = 0$ and define $u_0^n(s)$ by

$$u_0^n(s) = \max_{a \in A_s} \left\{ r(s, a) + \sum_{j \in S} \lambda p(j \mid s, a) v^n(j) \right\}. \tag{6.36}$$

   (b) If $\|u_0^n - v^n\| < \varepsilon(1 - \lambda)/2\lambda$ go to 4. Otherwise go to (c).
   (c) If $k = m$, go to (e). Otherwise, compute u_{k+1}^n by

$$u_{k+1}^n(s) = r(s, d_{n+1}(s)) + \sum_{j \in S} \lambda p(j \mid s, d_{n+1}(s)) u_k^n(j). \tag{6.37}$$

   (d) Increment k by 1 and return to (c).
   (e) Set $v^{n+1} = u_m^n$ and go to step 2.
4. Set $d^\varepsilon = d_{n+1}$ and stop.

This algorithm combines features of both policy iteration and value iteration. Like value iteration, it is an iterative algorithm. The stopping criterion used in step 3b is identical to that of value iteration; when it is satisfied, the resulting policy is ε-optimal. The computation of u_0^n in step 3a requires no additional

work because it already has been determined in step 2 when obtaining a v^n-improving decision rule.

Like policy iteration, the algorithm contains an improvement step, step 2, and an evaluation step, step 3; however, the evaluation is not done exactly. Instead it is carried out iteratively in step 3c, which is repeated m times. In vector notation this corresponds to

$$v^{n+1} = (T_{d_{n+1}})^{m+1} v^n .$$

The quantity m can be selected in advance or adaptively. For instance, m can be chosen so that $\|u^n_{m+1} - u^n_m\|$ is less than some prespecified tolerance which can vary with n. Ohno and Ichiki (1987) investigate alternative specifications for this tolerance and show numerically that fixed low orders of m work well, adaptive choice is better and considerable reduction in effort is obtained by using Gauss–Seidel methods in both the improvement and evaluation steps.

The algorithm is based on the following policy iteration representation from Proposition 6.14:

$$v^{n+1} = v^n + (I - \lambda P_{d_{n+1}})^{-1} B v^n . \tag{6.38}$$

Expanding $(I - \lambda P_{d_{n+1}})^{-1}$ in its Neumann series representation (Yosida, 1968), truncating it at m and substituting into (6.38) gives the following representation for the iterates of modified policy iteration:

$$v^{n+1} = v^n + \sum_{k=0}^{m} (\lambda P_{d_{n+1}})^k B v^n . \tag{6.39}$$

Equation (6.39) shows that modified policy iteration includes value iteration and policy iteration as extreme cases; modified policy iteration of order 0 is value iteration and of infinite order is policy iteration. The modified policy iteration algorithm corresponds to performing one value iteration step in which the maximum in (6.36) is computed and then m successive approximation steps with the fixed decision rule d_{n+1}. Figure 6.2 illustrates this for modified policy iteration of order 2.

The quantity v^{n+1} is the expected total discounted reward obtained by using the stationary policy d_{n+1} in a problem with finite horizon m and terminal reward v^n. Alternatively, v^{n+1} is the expected total discounted reward of the policy which used d_{n+1} for the first m periods, d_n for the next m periods and so forth, in an $(n+1)m$ period problem with terminal reward v^0.

The convergence of the algorithm has been demonstrated by Puterman and Shin (1978) and Rothblum (1979) and can be summarized as follows.

Theorem 6.18. *Suppose* $Bv_0 \geq 0$. *Then*

(i) *the sequence of iterates of modified policy iteration converge monotonically and in norm to* v^*_λ, *and*

(ii) *the algorithm terminates in a finite number of iterations with an ε-optimal policy.*

One might conjecture that the iterates of modified policy iteration of order $m + k$ $(k \geq 0)$ always dominate those for MPI order m when started at the same initial value. An example of van der Wal and van Nunen (1977) which appears in Puterman and Shin (1978) indicates that this conjecture is false.

Puterman and Shin (1978) provide the following result regarding the rate of convergence.

Theorem 6.19. *If*

$$\lim_{n\to\infty} \|P_{d_n} - P_{d_{v_\lambda^*}}\| = 0\,, \tag{6.40}$$

then

$$\limsup_{n\to\infty} \frac{\|v^{n+1} - v_\lambda^*\|}{\|v^n - v_\lambda^*\|} = \lambda^{m+1}\,.$$

This result demonstrates the appeal of this algorithm. When the policy is close to optimal, the convergence rate of the algorithm is close to that of $m + 1$ steps of value iteration. Computationally this represents a major savings over value iteration because MPI avoids the maximization at each pass through the algorithm. Conditions which imply (6.40) were given in the previous section. It always holds for finite state and action problems in which a rule is used to uniquely choose the v^n-improving policy is step 2.

The MPI algorithm will converge in fewer iterations than value iteration and at least as many iterations as policy iteration; however the computational effort per iteration exceeds that for value iteration and is less than that for policy iteration. Computational results in Puterman and Shin (1978) suggest that it is a more computationally efficient method for solution of practical Markov decision problems then either value iteration or policy iteration. Determining an efficient procedure for selecting m is still an open problem although results of Dembo and Haviv (1984) provide insight.

6.6. Linear programming

The discounted infinite horizon MDP can be formulated as a linear programming problem (d'Epenoux, 1960). The following discussion gives results based on that formulation. The development follows Derman (1970) and Kallenberg (1983).

Theorem 6.2 showed that if

$$v \geq r_d + \lambda P_d v$$

for all $d \in D$, then v is an upper bound for the value of the MDP, v_λ^*. Since v_λ^* also satisfies this inequality, it must be the smallest such solution. This is the basis for the following linear program.

Primal Linear Program.

$$\text{Minimize} \quad \sum_{j \in S} \alpha_j v(j)$$

subject to

$$v(s) \geq r(s, a) + \sum_{j \in S} \lambda p(j \mid s, a) v(j), \quad a \in A_s \text{ and } s \in S,$$

and $v(s)$ unconstrained.

The constants α_j are arbitrary positive quantities which are assumed without loss of generality to satisfy $\Sigma_{j \in S} \alpha_j = 1$. Its dual is:

Dual Linear Program.

$$\text{Maximize} \quad \sum_{s \in S} \sum_{a \in A_s} r(s, a) x(s, a)$$

subject to

$$\sum_{a \in A_s} x(j, a) - \sum_{s \in S} \sum_{a \in A_s} \lambda p(j \mid s, a) x(s, a) = \alpha_j, \quad j \in S,$$

and $x(j, a) \geq 0$ for all $a \in A_j$, $j \in S$.

Numerical results adopted from Koehler (1976) which appear in Puterman and Shin (1978) demonstrate that modified policy iteration is considerably more efficient than simplex method based linear programming codes for solving discounted Markov decision problems. Other computational disadvantages of linear programming include the additional effort required to generate the linear programming tableau and the inability of linear programming methods to take advantage of the easily available initial basis feasible solution described in Theorem 6.21 below. However, recent theoretical and computational advances in solution methods for linear programming could alleviate these shortcomings. Two clear advantages of the linear programming approach are that it allows easy inclusion of constraints (Kallenberg, 1983, pp. 72–77) and it facilitates sensitivity analysis.

The interest in the linear programming formulation is partly theoretical and partly due to excellent software. Most important results are based on the dual formulation. They are as follows.

Theorem 6.20. *The dual problem is always feasible and bounded. For a randomized stationary policy d, the quantities*

$$x(s, a) = \sum_{j \in S} \alpha_j \sum_{n=0}^{\infty} \lambda^n P(X_n^d = s, d(X_n^d) = a \mid X_0^d = j), \quad a \in A_s, \ s \in S, \tag{6.41}$$

are a feasible solution to the dual problem.

Conversely, if $x(s, a)$ is a solution to the dual problem, then the randomized stationary policy d defined by

$$P\{d(s) = a\} = \frac{x(s, a)}{\sum_{a' \in A_s} x(s, a')}, \quad a \in A_s, s \in S,$$

satisfies (6.41).

The quantity $x(s, a)$ defined in (6.41) is the discounted joint probability that the system is in state s and action a is selected, averaged over initial distribution $\{\alpha_j\}$.

Corollary 6.21. *Any basic feasible solution has the property that for each $s \in S$, $x(s, a) > 0$ for only one $a \in A_s$. If x^* is an optimal basic feasible solution, an optimal deterministic stationary policy is obtained by setting $d^*(s) = a$ whenever $x^*(s, a) > 0$.*

The matrix defining the constraints in the dual problem is a Leontief matrix (Veinott, 1968), that is, each column has exactly one positive entry and for any non-negative right hand side the linear system has a non-negative solution. A consequence of this observation is that for any non-negative vector α the dual linear program has the same optimal basic feasible solution.

The relationship between the simplex algorithm and the dynamic programming algorithms is as follows. When the dual problem is solved by the simplex algorithm with block pivotting, it is equivalent to policy iteration. When policy iteration is implemented by changing only the action which gives the maximum improvement over all states, it is equivalent to solving the dual problem by the usual simplex method. Modified policy iteration is equivalent to a variant of linear programming in which the basic feasible solution is evaluated by relaxation instead of direct solution of the linear system.

6.7. Bounds and action elimination

The methods presented in this section can be used to improve the efficiency of value iteration, policy iteration and modified policy iteration. Fundamental are bounds on the optimal value function.

6.7.1. Bounds for discounted Markov decision processes

This section presents iteratively determined upper and lower bounds for the optimal value function. They are of considerable importance computationally because they can be used to:

(a) provide stopping criteria for the non-finite iterative algorithms,

(b) provide improved terminal value functions when the algorithms have been stopped and

(c) eliminate suboptimal actions throughout the iterative process.

They are based on the result in Theorem 6.2 that if $Bv \geq (\leq)0$, then $v \leq (\geq)v_\lambda^*$.

References on this topic include MacQueen (1967), Porteus (1971), White (1978) and Puterman and Shin (1982). Define the real valued functions L and U for $v \in V$ by

$$L(v) = \min_{s \in S} v(s) \quad \text{and} \quad U(v) = \max_{s \in S} v(s) . \tag{6.42}$$

Let $1 \in V$ denote the function that is 1 for all $s \in S$. It is assumed throughout this section that S is finite so that the extrema in (6.42) are attained.

The following result is fundamental. A proof is based on applying Proposition 6.13 and Theorem 6.2. The decision rule d_v in the tightest lower bound is any v-improving decision rule.

Proposition 6.22. *For any $v \in V$ and $m \geq 0$,*

$$\begin{aligned} v + (1-\lambda)^{-1}L(Bv)1 &\leq v + Bv + \lambda(1-\lambda)^{-1}L(Bv)1 \\ &\leq v + \sum_{k=0}^{m} (\lambda P_{d_v})^k Bv + \lambda^{m+1}(1-\lambda)^{-1}L(Bv)1 \leq v_\lambda^* \\ &\leq v + Bv + \lambda(1-\lambda)^{-1}U(Bv)1 \leq v + (1-\lambda)^{-1}U(Bv)1 . \end{aligned} \tag{6.43}$$

These bounds can be applied during any iterative procedure by replacing v in (6.43) by the current value v^n. Results for value iteration and policy iteration are obtained by setting $m = 0$ and letting $m \to \infty$ respectively in (6.43). For value iteration, the two tightest lower bounds are identical. For modified policy iteration and policy iteration, the tightest lower bound becomes $v^{n+1} \leq v_\lambda^*$; in the case of policy iteration $v^{n+1} = v_{d_{n+1}}$.

6.7.2. Stopping criteria and extrapolations

Value iteration and modified policy iteration are non-finite algorithms and consequently require conditions to ensure that they stop in a finite number of iterations with an ε-optimal solution. The stopping criterion used in the algorithms of Sections 6.2–6.4 is too conservative. Empirical evidence indicates that these criteria require many unnecessary iterations to confirm that a policy identified early in the iterative process is optimal (Section 6.8). The bounds in Proposition 6.22 are the basis for more efficient stopping criteria.

The *span* of v, denoted by $\mathrm{sp}(v)$ is defined by

$$\mathrm{sp}(v) = U(v) - L(v) .$$

The result below is an immediate consequence of Proposition 6.22.

Proposition 6.23. *Suppose for $v \in V$ and $\varepsilon > 0$ that*

$$\mathrm{sp}(Bv) = U(Bv) - L(Bv) < \frac{(1-\lambda)}{\lambda}\,\varepsilon . \tag{6.44}$$

Then

$$\|v + Bv + \lambda(1-\lambda)^{-1}L(Bv)1 - v_\lambda^*\| < \varepsilon \tag{6.45}$$

and for any v-improving decision rule d_v,

$$\|v_{d_v} - v_\lambda^*\| < \varepsilon . \tag{6.46}$$

When ε is small and (6.44) is satisfied, Bv is nearly constant so that the value of a v-improving decision rule differs from v by nearly a constant amount. Thus at the subsequent value, Bv will again be nearly constant. It can be shown that this constant must be close to 0. When this constant has been added to v, the resulting value is closer to v_λ^* as shown by (6.45).

As a consequence of Proposition 6.23, (6.44) provides an alternative stopping rule for value iteration and modified policy iteration. It replaces the stopping rule in step 3 of the value iteration algorithm or step 3b of the modified policy iteration algorithm. Note that in both of these algorithms, $Bv^n = Tv^n - v^n$ is available prior to testing whether the stopping criteria are met. Determining $U(Bv^n)$ and $L(Bv^n)$ requires little additional effort. Although policy iteration is a finite algorithm, the above stopping criteria can be included after step 3, if all that is required is an ε-optimal policy.

The quantity $v + Bv + (1-\lambda)^{-1}\lambda L(Bv)1$ in (6.48) is called a *lower bound extrapolation* by Porteus and Totten (1978) and gives an improved approximation to v_λ^* upon termination of the algorithm. One might conjecture that convergence of the algorithms would be faster if such extrapolations could be incorporated at each iteration. Unfortunately this is not the case because the set of maximizing actions is unaffected by addition of a scalar. Such extrapolations are more useful in semi-Markov decision problems in which transition matrices do not have equal row sums. Other extrapolations are also available (Porteus, 1980a).

6.7.3. Action elimination

The above bounds can be used to identify suboptimal actions at each iteration of any algorithm for solving discounted MDP's. The suboptimal

actions are eliminated from the action set at subsequent iterations and

$$r(s, a) + \sum_{j \in S} \lambda p(j|s, a) v^n(j)$$

must be evaluated for fewer actions. Also, the identification and elimination of suboptimal actions is the *only* way of determining an optimal (as opposed to an ε-optimal) policy when using a non-finite iterative algorithm such as value iteration or modified policy iteration. When all but one action is eliminated in each state, the stationary policy which uses the remaining decision rule is necessarily optimal.

Action elimination procedures are based on the following observation of MacQueen (1967).

Proposition 6.24. *If*

$$r(s, a) + \sum_{j \in S} \lambda p(j|s, a) v_\lambda^*(j) - v_\lambda^*(s) < 0\,, \tag{6.47}$$

then any stationary policy which uses action a in state s cannot be optimal.

Referring to Figure 6.1 shows why this result holds. In its policy 3 is optimal and the one-step improvement functions of all other policies are bounded above by 0 at v_λ^*.

Since v_λ^* is unknown, the result in Proposition 6.24 cannot be used directly to identify suboptimal actions. Instead, Proposition 6.24 permits upper and lower bounds for v_λ^* to be substituted into (6.47) to obtain an implementable elimination rule.

Theorem 6.25. *Suppose* $v^+ \geq v_\lambda^* \geq v^-$. *Then if*

$$r(s, a) + \sum_{j \in S} \lambda p(j|s, a) v^+(j) < v^-(s)\,, \tag{6.48}$$

any stationary policy which uses action a in state s is suboptimal.

Implementations of these bounds for MPI appear in Ohno (1980) and Puterman and Shin (1982). Algorithms which identify actions that cannot be chosen in the improvement step at the subsequent iteration have been developed by Hastings and van Nunen (1977), Hubner (1977) and Puterman and Shin (1982). These are based on similar principles but their implementation is considerably more tedious.

6.8. Computational results

In this section, an infinite horizon version of the stochastic inventory model of Section 3.2 is solved using value iteration, policy iteration and modified

policy iteration and the results are discussed. The data are the same as those used in the finite horizon case in Section 4.5, the discount rate λ is 0.9. The objective is to determine the stationary policy that maximizes the expected total infinite horizon discounted reward. All calculations were carried out using MDPLAB (Lamond, 1984).

6.8.1. Value iteration

To initiate the algorithm, $v^0 = 0$ and $\varepsilon = 0.1$. The algorithm will terminate with a stationary policy that has expected total discounted reward within 0.1 of optimal. Calculations proceed as in the finite horizon backward induction algorithm until the stopping criterion of

$$\|v^{n+1} - v^n\| \leq \frac{\varepsilon(1-\lambda)}{2\lambda} = \frac{0.1 \times 0.1}{2 \times 0.9} = 0.0056$$

is satisfied. The value functions v^n and the maximizing actions obtained in Step 2 at several iterations are provided in Table 6.1.

Table 6.1
Value iteration results

n	$v^n(s)$				$d^n(s)$				Δ^n
	$s=0$	$s=1$	$s=2$	$s=3$	$s=0$	$s=1$	$s=2$	$s=3$	
0	0	0	0	0	0	0	0	0	
1	0	5.0	6.0	5.0	2	0	0	0	6.0000
2	1.6	6.125	9.6	9.95	2	0	0	0	2.8250
3	3.2762	7.4581	11.2762	12.9368	3	0	0	0	1.6537
4	4.6632	8.8895	12.6305	14.6632	3	0	0	0	0.3721
5	5.9831	10.1478	13.8914	15.9831	3	0	0	0	0.0616
6	7.1306	11.3218	15.0383	17.1306	3	0	0	0	0.0271
7	8.1690	12.3605	16.0828	18.1690	3	0	0	0	0.0061
10	10.7071	14.8966	18.6194	20.7071	3	0	0	0	
15	13.5019	17.6913	21.4142	23.0542	3	0	0	0	
30	16.6099	20.7994	24.5222	26.6099	3	0	0	0	
50	17.4197	21.6092	25.3321	27.4197	3	0	0	0	
56	17.4722	21.6617	25.3845	27.4722	3	0	0	0	
57	17.4782	21.6676	25.3905	27.4782	3	0	0	0	
58	17.4836	21.6730	25.3959	27.4836	3	0	0	0	

Estimates based on the error bound in (6.23) indicate that 68 iterations are required to obtain a 0.1-optimal policy. In fact, using the stopping criterion above leads to termination after 58 iterations when $\|v^{58} - v^{57}\| = 0.0054$. The 0.1-optimal stationary policy is $d^\varepsilon = (3, 0, 0, 0)$. This is the policy which orders only when the stock level is 0, and in that case orders 3 units. Observe that the optimal policy was first identified at iteration 3 but the algorithm did not terminate until iteration 58.

The stopping criterion of Section 6.7.2 yields

$$\Delta^n \equiv U(Bv^n) - L(Bv^n) < \frac{1-\lambda}{\lambda}\,\varepsilon = \frac{0.1}{0.9} \times 0.1 = 0.0111\,. \tag{6.49}$$

To apply this stopping rule, note that $Bv^n = v^{n+1} - v^n$. Observe from the last column of Table 6.1 that when using this stopping rule, the algorithm terminates with a 0.1-optimal policy after only 7 iterations.

6.8.2. Policy iteration

To start the policy iteration algorithm choose the myopic policy, i.e., that which maximizes the immediate one-period reward $r(s, a)$. The algorithm then proceeds as follows:

1. Set $d^0 = (0, 0, 0, 0)$ and $n = 0$.
2. Solve the evaluation equations obtained by substituting the transition probabilities and rewards corresponding to policy d^0 into (6.24) to obtain $v^0 = (0, 6.4516, 11.4880, 14.9951)$.
3. For each s the quantities

$$r(s, a) + \sum_{j=0}^{3} p(j|s, a)v^0(j)$$

are computed for $a = 0, \ldots, 3 - s$ and the actions which achieve the maximum are placed into $A^*_{0,s}$. In this example there is a unique maximizing action in each state so that

$$A^*_{0,0} = \{3\}\,, \quad A^*_{0,1} = \{2\}\,, \quad A^*_{0,2} = \{0\}\,, \quad A^*_{0,3} = \{0\}\,.$$

4. Since $d^0(0) = 0$, it is not contained in $A^*_{0,0}$, so continue.
5. Set $d^1 = (3, 2, 0, 0)$, $n = 1$ and return to the evaluation step.

The detailed step by step calculations for the remainder of the algorithm are omitted. The value functions and corresponding maximizing actions are presented below. Since there is a unique maximizing action in the improvement step at each iteration, $A^*_{n,s}$ is equivalent to $d^n(s)$ and only the latter is displayed, in Table 6.2.

The algorithm terminates in three iterations with the optimal policy $d^* = (3, 0, 0, 0)$. Observe that an evaluation was unnecessary at iteration 3 since $d^2 = d^3$ terminated the algorithm prior to the evaluation step. Unlike value iteration, the algorithm has produced an optimal policy as well as its expected total discounted reward $v^3 = v^*_\lambda$. Note in this example that the ε-optimal policy found using value iteration is optimal but it could not be recognized as such without using action elimination.

Table 6.2
Policy iteration results

n	$v^n(s)$				$d^n(s)$			
	$s=0$	$s=1$	$s=2$	$s=3$	$s=0$	$s=1$	$s=2$	$s=3$
0	0	6.4516	11.4880	14.9951	0	0	0	0
1	10.7955	12.7955	18.3056	20.7955	3	2	0	0
2	17.5312	21.7215	25.4442	27.5318	3	0	0	0
3	×	×	×	×	3	0	0	0

6.8.3. *Modified policy iteration*

The following illustrates the application of modified policy iteration of order 5. The first pass through the algorithm is described in detail; calculations for the remainder are presented in tabular form.

1. Set $v^0 = (0, 0, 0, 0)$, $n = 0$ and $\varepsilon = 0.1$.
2. Observe that

$$r(s, a) + \sum_{j=0}^{3} \lambda p(j \mid s, a) v^0(j) = r(s, a)$$

so that for each s, the maximum value occurs when $a = 0$. Thus $A^*_{n,s} = \{0\}$ for $s = 0, 1, 2, 3$ and $d^n = (0, 0, 0, 0)$.

3.0. Set $k = 0$ and $u_0^0 = (0, 5, 6, 5)$.

3.1. Since $\|u_0^0 - v^0\| = 6 > 0.0056$, continue.

3.2. Since $k = 0 < 5$, continue. Compute u_1^0 by

$$u_1^0(s) = r(s, d^0(s)) + \sum_{j=0}^{3} \lambda p(j \mid s, d^0(s)) u_0^0(j)$$

$$= r(s, 0) + \sum_{j=0}^{3} \lambda p(j \mid s, 0) u_0^0(j)$$

to obtain $u_1^0 = (0, 6.125, 9.60, 10.95)$.

3.3. Set $k = 2$ and return to 3.2.

The loop is repeated 4 more times to evaluate u_2^0, u_3^0, u_4^0 and u_5^0. Then v^1 is set equal to u_5^0 and the maximization in step 2 is carried out. The resulting iterates appear in Table 6.3.

In Step 3.1, following iteration 11, the computed value of u^0 is (17.4976, 21.6871, 25.4100, 27.4976), so that $\|u_0^{11} - v^{11}\| = 0.0038$ and the policy $(3, 0, 0, 0)$ is ε-optimal with $\varepsilon = 0.1$.

If instead the stopping criteria of Section 6.7.2 is applied the algorithm terminates after 4 iterations. The last column of Table 6.3 gives Δ^n as defined in (6.50).

Table 6.3
Iterates of modified policy iteration

n	$v^n(s)$				$d^n(s)$				Δ^n
	$s=0$	$s=1$	$s=2$	$s=3$	$s=0$	$s=1$	$s=2$	$s=3$	
0	0	0	0	0	0	0	0	0	
1	0	6.4507	11.4765	14.9200	3	2	0	0	6.0000
2	7.1215	9.1215	14.6323	17.1215	3	0	0	0	4.9642
3	11.5709	15.7593	19.4844	21.5709	3	0	0	0	2.6011
4	14.3639	18.5534	22.2763	24.3639	3	0	0	0	0.0022
5	15.8483	20.0377	23.7606	25.8483	3	0	0	0	
10	17.4604	21.6499	25.3727	27.4604	3	0	0	0	
11	17.4738	21.6833	25.4062	27.4938	3	0	0	0	

6.8.4. Discussion of results

Using the norm based stopping criterion, value iteration required 58 iterations to obtain a 0.1-optimal solution while using the span based criterion (6.49), value iteration required 7 iterations and modified iteration of order 5 required 4 iterations. Clearly the span based stopping rule greatly improved the efficiency of each of these procedures.

At each pass through the value iteration algorithm, a maximization over the action set was required in each state. This means that the quantity

$$r(s,a) + \sum_{j \in S} \lambda p(j|s,a) v^n(j)$$

had to be computed for each action at each iteration. Thus in problems with large action sets, this step of the algorithm would be time consuming. Modified policy iteration performs this maximization far less frequently so that one would expect a considerable improvement in efficiency, especially when the maximizing actions do not change often. Tables 6.1 and 6.3 show that after the third iteration there was no change in the v^n-improving decision rule so that the value iteration algorithm performed many unnecessary maximizations. Based on results using the span based stopping rule, value iteration required 8 maximizations while modified policy iteration of order 5 required 5. While not a dramatic savings in this small problem, it illustrates the potential for considerable improvement in larger problems.

Note also that the sequence of v^n-improving decision rules obtained using value iteration and MPI was different. Comparing Tables 6.1 and 6.2 shows that those of modified policy iteration and policy iteration were identical. This is to be expected, since modified policy iteration does an approximate policy evaluation at each pass through the algorithm which is usually adequate to identify an improved decision rule. In the example above, when the span based stopping rule was used with modified policy iteration, MPI required 5 maximizations while policy iteration required 4. Thus because Gaussian elimination was avoided, MPI probably required fewer multiplications and divisions. In

this small problem, such improvements are unimportant but in problems with large state spaces, MPI can be considerably more efficient than policy iteration. This was demonstrated numerically in Puterman and Shin (1978). An open question in implementing modified policy iteration is how best to choose the order which can be varied from iteration to iteration.

Calculations using action elimination are not given here. The reader is referred to Puterman and Shin (1982) and Ohno and Ichiki (1987) for results using such methods. In Puterman and Shin, one-step ahead action-elimination algorithms were shown to be most efficient and are recommended for use together with a Gauss–Seidel version of modified policy iteration whenever solving a large discounted MDP. It is expected that increased efficiency can be attained by incorporating other methods of Section 6.3.3.

6.9. Unbounded rewards and countable state spaces

Countable state spaces are natural settings for applications of MDP's to queueing control and inventory management. In such applications, rewards or costs are often assumed to be linear in the state variable and consequently are unbounded. Almost all of the previous results in this section implicitly assumed that r and v were bounded. Most importantly, existence of a solution to the optimality equation was based on the result that Tv defined in (6.2) is a contraction mapping on the set of *bounded* real valued functions on S. This subsection discusses a modified approach when the boundedness requirement is removed.

Harrison (1972) is the first author to explicitly deal with this issue. Subsequent contributions for discounted problems include Lippman (1975), van Nunen (1976b), Wessels (1978) and van Nunen and Wessels (1978). Most of these results are based on modifying V so that it includes a sufficient number of unbounded functions. The state space is assumed to be countable in this section.

Let w be an arbitrary positive real valued function on S (for example, when S is a subset of $(0, \infty)$, $w(s) = s$). Define the *weighted supremum norm* $\|\cdot\|_w$ for real valued functions v on S by

$$\|v\|_w = \sup_{s \in S} w(s)^{-1} |v(s)| \tag{6.50}$$

and let V_w be the Banach space of real valued functions v on S satisfying $\|v\|_w < \infty$.

For a matrix $H : V_w \to V_w$ the above norm induces the matrix norm given by

$$\|H\|_w = \sup_{s \in S} w(s)^{-1} \sum_{j \in S} |h_{sj}| \, w(j)$$

where h_{sj} is the (s, j)th component of H.

The following two conditions are adapted from Lippman (1975):
(1) There exists a constant M such that for all $d \in D$,

$$\|r_d\|_w \leq M . \tag{6.51}$$

(2) There exists a finite non-negative constant L such that for all $d \in D$,

$$P_d w \leq w + L . \tag{6.52}$$

If (6.51) and (6.52) holds with $L = 0$, then T defined in (6.3) is a contraction mapping on V_w so that the results of Section 6.3 apply. Unfortunately, the requirement that $L = 0$ is too restrictive for most applications. Lippman showed that with $L > 0$, that T is an n-stage contraction on V_w, so that results of Denardo (1967) can be applied to obtain the following generalization of earlier results.

Theorem 6.26. *Suppose that* (6.51) *and* (6.52) *hold. Then the optimality equation*

$$Tv = \sup_{d \in D} \{r_d + \lambda P_d v\} \tag{6.53}$$

has a unique solution v_λ^* *in* V_w *which can be found by value iteration. Furthermore, if the supremum in* (6.53) *is attained, there exists an optimal deterministic stationary policy.*

The condition that $\|r_d\|_w \leq M$ is equivalent to $|r(s, a)| \leq Mw(s)$ for all $a \in A_s$ and $s \in S$ which implies that r grows at most rate w in s. Based on this, a suitable choice for w is

$$w(s) = \sup_{a \in A_s} |r(s, a)| \tag{6.54}$$

in which case M can be chosen to be 1 in (6.52).

The condition that $P_d w \leq w + L$ is equivalent to

$$E_{s,a} w(X_1) = \sum_{j \in S} p(j \mid s, a) w(j) \leq w(s) + L \tag{6.55}$$

for all $a \in A_s$ and $s \in S$ where X_1 is the random variable with values in S and probability distribution given by $p(\cdot \mid s, a)$. This means that when w is chosen so that (6.52) holds, under any decision rule, the expected reward in the next period cannot exceed w by more than L units. This condition places restrictions on allowable transitions but does allow transitions to distant states with small weighted probability.

A countable state version of the inventory model of Section 3.3 can be shown to satisfy (6.51) and (6.52) under the reasonable assumption that the

maximum quantity ordered in any period is bounded. Lippman (1975) shows that these conditions are satisfied in certain machine replacement, optimal consumption and queueing control problems.

7. Undiscounted Markov decision problems—I

The next two sections are concerned with MDP's without discounting. In this section the focus is problems with expected total reward criterion. Assumptions are imposed so that the expected total reward is well defined for all policies and finite for some policies. Implicit in these formulations are restrictions on the reward functions. When the expected total rewards is unbounded or not well defined, for all policies, the average and sensitive optimality criteria of Section 8 are of greater practical significance.

In Section 6, a complete theory for discounted problems was presented. Crucial to this theory was the existence of a discount factor $\lambda < 1$ which ensured that

(a) the operator T defined in (6.3) was a contraction,
(b) $(I - \lambda P_d)^{-1}$ existed and was non-negative for all d, and
(c) for bounded v,

$$\lim_{n\to\infty} \lambda^n P_\pi^n v = \lim_{n\to\infty} \lambda^n E_\pi\{v(X_n^\pi)\} = 0 .$$

As a result of (a), T has a unique fixed point which could be obtained by value iteration. Because of (b), v_λ^π is well defined, any v_λ^π-improving decision rule has an expected total discounted reward at least as great as that of π, and bounds on the optimal total discounted reward are available. Condition (c) is used to show that v_λ^* is a solution of the optimality equation.

The focus of this section will be the expected total reward criterion (5.2). That is, policies will be compared on the basis of

$$v^\pi(s) = E_{\pi,s}\left\{\sum_{t=1}^{\infty} r(X_t^\pi, d_t(X_t^\pi))\right\}$$

which is the expected total discounted reward with $\lambda = 1$. Without additional assumptions, there is no guarantee that the above limit exists. Also, (a)–(c) above are not valid. This situation necessitates restricting attention to cases when v^π is well defined in addition to using different methods of analysis.

Strauch (1966) and Blackwell (1967) distinguished the positive and negative cases in which the expected total discounted reward is well defined. In the positive case all rewards are non-negative and in the negative case all rewards are non-positive. These two cases are distinct because maximization is done in each so that one cannot be obtained from the other by sign reversal.

The key mathematical properties that will be used in this section are that v^π is well defined, the optimal return operator is monotone and monotone

sequences are convergent. The behavior of $P_\pi^n v$ for large n is crucial and results below about its properties will indicate why the positive and negative cases are distinguished. To paraphrase Strauch (1966), analysis in the total reward cases is based on improving the tails while that in the discounted case is based on ignoring the tails.

Several authors including Hordijk (1974), Schal (1975), van Hee (1978), van der Wal (1984) and van Dawen (1986a, 1986b) have analyzed MDP's with total reward without distinguishing the positive and negative cases; however, in this section they will be discussed separately. The state space will be assumed to be general except where noted.

7.1. The positive bounded case

An MDP is said to be *positive* if $r(s, a) \geq 0$ for all $a \in A_s$ and $s \in S$ and *bounded* if there exists an $M < \infty$ such that

$$v^\pi = \sum_{t=1}^{\infty} P_\pi^{t-1} r_{d_t} \leq M \tag{7.1}$$

for all $\pi \in \Pi$. For some results, the uniform boundedness assumption on v^π can be relaxed (van Hee, 1978).

When S is finite and $\Sigma_{j \in S} p(j|s, a) = 1$ for each $a \in A_s$ and $s \in S$, (7.1) holds for each π if and only if $r = 0$ on the recurrent classes of π. When S is infinite, $r = 0$ at each positive recurrent state implies (7.1).

The objectives in analyzing the positive bounded case are to characterize the value of the MDP,

$$v^*(s) = \sup_{\pi \in \Pi} v^\pi(s) ,$$

to determine when an optimal or ε-optimal policy π^* exists, and to characterize its form. Motivated by Blackwell (1967), several authors have investigated the positive case including Strauch (1966), Ornstein (1969) and Hinderer (1970); other references appear above.

7.1.1. The optimality equation

The optimality equation in undiscounted MDP's with expected total reward criteria is given by

$$v(s) = \max_{a \in A_s} \left\{ r(s, a) + \sum_{j \in S} p(j|s, a) v(j) \right\}, \quad s \in S .$$

It can be expressed in matrix–vector notation by

$$v = \max_{d \in D} \{ r_d + P_d v \} \equiv Tv \tag{7.2}$$

where conventions of previous sections regarding the maximum are assumed to hold. The operator T will be referred to as the *optimal return* operator. Since the returns of all policies are non-negative, it will be expeditious to analyze (7.2) on $V^+ = \{v \in V: v \geq 0\}$. The optimal return operator T maps V^+ into V^+ and the optimality equation corresponds to a fixed point equation for T on V^+.

The result below is analogous to Theorem 6.2 in the discounted case. Its proof depends explicitly on the assumed positivity of v.

Theorem 7.1. *Suppose there exists a* $v \in V^+$ *for which* $v \geq Tv$. *Then* $v \geq v^*$.

In contrast to the discounted case, the converse of Theorem 7.1 that $v \leq Tv$ implies $v \leq v^*$, does not hold without further assumptions. The following simple example demonstrates this and also motivates several other results in this section.

Example 7.2. Let $S = \{1, 2\}$, $A_1 = \{a, b\}$, $A_2 = \{a\}$, $r(1, a) = 0$, $r(1, b) = 1$, $r(2, a) = 0$, $p(1|1, a) = 1$, $p(2|1, b) = 1$, $p(2|2, a) = 1$ and $p(j|s, a') = 0$ otherwise. There are two available decision rules d and e given by $d(1) = a$, $d(2) = a$, $e(1) = b$ and $e(2) = a$. Under d, both states 1 and 2 are recurrent, while under e, 1 is transient and 2 is recurrent. The optimality equation is

$$v(1) = \max\{v(1), 1 + v(2)\} \quad \text{and} \quad v(2) = v(2)\,.$$

It is easy to see that decision rule e is optimal, $v^e(1) = 1$, $v^e(2) = 0$ and $v^e = v^*$. Let $v(1) = v(2) = 2$ and observe that $Tv(1) = 3$, $Tv(2) = 2$. Thus $Tv \geq v$ does not guarantee that $v \leq v^*$ without further conditions.

This example also illustrates another important feature of the positive case, that the optimality equation *does not* possess a unique solution, since $v^* + c$ satisifies $Tv^* = v^*$ for any constant vector c.

The following result adapted from Schal (1975) and van Dawen (1986a, b) provides sufficient conditions for the above implication to be valid.

Theorem 7.3. *Suppose there exists a* $v \in V^+$ *such that* $v \leq Tv$. *Then* $v \leq v^*$ *if*

$$\liminf_{N\to\infty} E_\pi\{v(X_N^\pi)\} = \liminf_{N\to\infty} P_\pi^N v = 0 \quad \textit{for all } \pi \in \Pi\,. \tag{7.3}$$

Sufficient conditions for (7.3) to hold in the positive case include

(a) $v = 0$ or

(b) all policies are aperiodic and $v(s) = 0$ on all states that are in the positive recurrent class of some policy.

Combining the above two theorems gives an analogous result to Theorem 6.3 in the discounted case.

Theorem 7.4. *Suppose there exists a* $v \in V^+$, *such that* $v = Tv$ *and* (7.3) *holds. Then* $v = v^*$.

The following theorem (Blackwell, 1967) contains analogous results to those of Theorems 6.3 and 6.9 in the discounted case. Among other things, it states that value iteration starting from $v^0 = 0$ converges to a solution of the optimality equation and it provides an alternative identification of the optimal solution to that in Corollary 7.4.

Theorem 7.5. *The optimal return* v^* *is the smallest solution of the optimality equation* $Tv = v$ *in* V^+ *and* $v^* = \lim_{n\to\infty} T^n 0$.

7.1.2. Identification and existence of optimal policies

In the discounted case, Theorem 6.4, showed that a stationary policy based on any v^*-improving (conserving) decision rule is optimal. That is not true in the positive case as illustrated by Example 7.2 in which both decision rules are v^*-improving but d is sub-optimal. The following weaker result is available.

Corollary 7.6. *The stationary policy based on* d' *is optimal if and only if* $v_{d'}$ *is a fixed point of* T.

Van Dawen (1986b) refers to the decision rule described in the above corollary as *unimprovable*, that is, d' is unimprovable if $Tv^{d'} = v^{d'}$. Corollary 7.6 is equivalent to: d' is optimal if and only if d' is unimprovable.

In the discounted case it was shown that an optimal stationary policy existed under very weak assumptions. In particular, any condition which guaranteed the existence of a conserving decision rule was sufficient to ensure the existence of a stationary optimal policy. Unfortunately, this is not the situation in the positive case as demonstrated by examples of Strauch (1966) and Blackwell (1967).

The following theorem is based on a subtle argument allowing generalization of results in the discounted case.

Theorem 7.7. *Suppose that* S *is finite and for each* $s \in S$, A_s *is finite. Then there exists an optimal stationary policy.*

When S is more general, different notions of optimality and classes of optimal policies have been considered by Ornstein (1969) and Strauch (1966). Related results have been obtained by Derman and Strauch (1966), van Hee (1978), van der Wal (1981, 1984) and van Dawen (1986a,b).

7.1.3. Computational methods in the positive case

Value iteration: Theorem 7.5 showed that a solution of the optimality equation can in principle be obtained by the value iteration scheme $v^n = Tv^{n-1}$ where $v^0 = 0$. Unfortunately, stopping rules, bounds and action elimination criteria

are not available for this procedure so its significance is primarily theoretical. This convergence result can also be used to characterize the structure of an optimal policy (Kreps and Porteus, 1977) so that search for optima can be carried out in a smaller class of policies with special form.

Policy iteration: Strauch (1966) provides a countable state example in which the decision rule determined in the improvement step of policy iteration has a *strictly* smaller return than the initial policy. Consequently for countable state problems, policy iteration cannot be used to determine an optimal policy.

When S is finite, Example 7.2 shows that choosing *any* v^{d_n}-improving policy in the improvement step does not guarantee a policy with greater return, since decision rule d is v^e-improving but $v^d(1) < v^e(1)$. However if the same decision rule is v_{d_n}-improving at two successive iterations, it is unimprovable and consequently by Corollary 7.6, it is optimal. In the discounted case, the rule "set $d_{n+1} = d_n$ if possible" was included to avoid cycling, here it is essential to ensure that policy iteration yields monotone increasing value functions (cf. van Dawen, 1986b).

Another complication in implementing policy iteration is that that $v = r_d + P_d v$ does not necessarily have a unique solution (cf. Example 7.4). Applying Theorem 7.5 with A_s replaced by $\{d(s)\}$ characterizes v^d as the minimal solution of $v = r_d + P_d v$ in V^+.

Thus in the finite state and action case, policy iteration can be used to obtain an optimal policy in the positive case provided that in the evaluation step the minimal positive solution of the evaluation equation is selected, the rule "set $d_{n+1} = d_n$ if possible" is followed in the improvement step and it is started from the return of a stationary policy instead of an arbitrary $v_0 \in V^+$.

Linear programming: Linear programming for finite state and action positive MDP's has been studied by Kallenberg (1983). The formulation is similar to that in the discounted case; the constraints of the primal problem are identical to those in the discounted case with λ set equal to 1. However, the additional condition that $v \geqslant 0$ is added and consequently the equality constraints in the dual problem are replaced by inequalities. That v^* is a solution of the primal problem follows from Theorem 7.5. The dual problem is feasible because $x(s, a) = 0$ satisfies its inequality constraints. If the dual program has a finite optimum, then the problem can be solved by the simplex method and an optimal stationary policy is given by

$$d(s) = \begin{cases} a & \text{if } x(s, a) > 0 \text{ and } s \in S^*, \\ \text{arbitrary} & \text{if } s \in S - S^*, \end{cases}$$

where $S^* = \{s \in S: x(s, a_s) > 0 \text{ for some } a \in A_s\}$. When the dual is unbounded a more complicated procedure is provided by Kallenberg.

7.2. *The negative case*

An MDP is said to be *negative* if $r(s, a) \leqslant 0$ for all $a \in A_s$ and $s \in S$. Under this assumption, v^π is well defined but might equal $-\infty$. Without the further

assumption that v^π is finite for at least one policy $\pi \in \Pi$, $v^\pi = -\infty$ for all $\pi \in \Pi$ so that all policies are equivalent under the total expected reward criterion. If this is the case, the average reward criterion can be used to discriminate between policies.

The most natural setting for the negative case is cost minimization with non-negative costs. In such problems the reward function r is interpreted as negative cost; maximization of expected total reward corresponds to minimization of expected total cost. Contributors to theory in the negative case include Blackwell (1961), Strauch (1966), Kreps and Porteus (1977), Schal (1978), Whittle (1979), (1980a and b), Hartley (1980), Demko and Hill (1981) and van Dawen (1985).

Analysis in this section will parallel that in Section 7.1 as closely as possible. S will be arbitrary except where noted.

7.2.1. The optimality equation

The optimality equation for the negative case is given in (7.2). Since $r \leq 0$, solutions will be sought in V^-, the set of non-positive real valued functions on S. Unlike V^+, elements of V^- are not required to be bounded.

The following results are analogous to Theorems 7.1, 7.3 and 7.4 in the positive case.

Theorem 7.8. *Suppose there exists a* $v \in V^-$ *satisfying* $v \leq Tv$. *Then* $v \leq v^*$.

For the implication with the inequalities reversed, additional conditions are required as illustrated by Example 7.11 below. One such sufficient condition is given in the following result.

Theorem 7.9. *Suppose there exists a* $v \in V^-$ *such that* $Tv \leq v$ *and*

$$\limsup_{N\to\infty} E_\pi\{v(X_N^\pi)\} = \limsup_{N\to\infty} P_\pi^N v = 0\,. \tag{7.4}$$

Then $v \geq v^*$.

Sufficient conditions for (7.4) to hold are identical to those which imply (7.3) in the positive case. Combining Theorems 7.8 and 7.9 gives the following important result.

Theorem 7.10. *Suppose there exists a* $v \in V^-$ *such that* $v = Tv$ *and* (7.4) *holds. Then* $v = v^*$.

The following example of Blackwell (1960) provides a counterexample to many conjectures in the negative case.

Example 7.11. Let $S = \{1, 2\}$, $A_1 = \{a, b\}$, $A_2 = \{a\}$, $r(1, a) = 0$, $r(1, b) = -1$, $r(2, a) = 0$, $p(1|1, a) = 1$, $p(2|1, b) = 1$, $p(2|2, a) = 1$ and $p(j|s, a') = 0$ other-

wise. There are two available decision rules, d and e, given by $d(1) = a$, $d(2) = a$ and $e(1) = b$, $e(2) = a$. The optimality equation is

$$v(1) = \max\{v(1), -1 + v(2)\} \quad \text{and} \quad v(2) = v(2)\,.$$

Decision rule d is optimal and $v^d(1) = 0$, $v^d(2) = 0$ and $v^d = v^*$. Choosing $v(1) = v(2) = -1$ implies that $Tv(1) = -1$ and $Tv(2) = -1$, so that $Tv \leq v$ but $v \leq v^*$. Note that (7.4) is not satisfied in this example. Also, the optimality equation does not have a unique solution, since any $v \in V^-$ satisfying $v(1) \geq -1 + v(2)$ satisfies $Tv = v$.

The following result is the analog of Theorem 7.5 in the positive case. It provides an alternative characterization of the optimal return to that of Theorem 7.10.

Theorem 7.12. *The optimal expected total reward v^* is the largest solution of $Tv = v$ in V^- and $v^* = \lim_{n\to\infty} T^n 0$.*

Whittle (1979, 1980a,b) provides some generalizations of this theorem.

7.2.2. Identification and existence of optimal policies

In contrast to the positive case, v^*-improving (conserving) decision rules are optimal. This has important consequences for the existence of optimal stationary policies in the negative case. Recall that d^* is conserving if

$$d^* = \arg\max_{d\in D} \{r_d + P_d v^*\}\,.$$

Theorem 7.13. *Suppose d' is conserving. Then the stationary policy d' is optimal.*

In Example 7.11, $v^e(1) = -1$ and $v^e(2) = 0$ so that v^e is a solution of the optimality equation while e is suboptimal. Thus, in contrast to the positive case, unimprovable policies, that is policies π for which $Tv^\pi = v^\pi$, are not necessarily optimal.

Optimal policies exist under considerably more general conditions in the negative case than in the positive case. Strauch (1966) obtained the following result when the set of actions is finite in each state. Note that it is a considerably stronger result than in the positive case where a counterexample was available for S countable.

Theorem 7.14. *Suppose for each $s \in S$, A_s is finite. Then there exists an optimal stationary policy.*

By Theorem 7.12, there exists a solution to the optimality equation, so as a consequence of Theorem 7.14, any condition which ensures the existence of

conserving decision rules is sufficient to guarantee the existence of a stationary policy. Further results for this case are provided by Strauch (1966) and Schal (1975).

7.2.3. Computation

Value iteration: As a consequence of Theorem 7.12, value iteration is convergent provided $v^0 = 0$. As in the positive case, the absence of bounds on $v^n - v^*$ makes solution by value iteration impractical. Van Dawen (1985) shows that if S is finite and $v^* > -\infty$, the convergence rate is geometric.

Policy iteration: In the negative case the situation regarding policy iteration is the reverse of the positive case. That is, the improvement step is valid while the termination criterion is not. If d is the current stationary policy and d' is chosen in the improvement step, then $v^{d'} \geqslant v^d$ so successive iterates are monotone. But d' can satisfy the stopping criterion

$$r_{d'} + P_{d'}v^{d'} = \max_{d \in D} \{r_d + P_d v^{d'}\} \tag{7.5}$$

and not be optimal. This is because (7.5) implies only that $v^{d'}$ is a solution of the optimality equation but since the optimality equation does not have a unique solution, it need not equal v^*.

To illustrate this, suppose, in Example 7.11, that policy iteration begins with stationary policy e. Then

$$-1 + v^e(2) = \max\{v^e(1), -1 + v^e(2)\} = Tv^e(1)$$

and

$$v^e(2) = Tv^e(2)\,,$$

so that (7.5) is satisfied. Thus, the algorithm will terminate in the improvement step with the suboptimal policy e.

Linear programming: The optimal expected return and optimal policies cannot be determined by a direct application of linear programming in the negative case. The primal linear programming problem derived from Theorem 7.12 is given by

$$\text{Maximize} \quad \sum_{j \in S} \alpha_j v(j)$$

subject to

$$v(s) \geqslant r(s, a) + \sum_{j \in S} p(j \mid s, a) v(j)\,, \quad a \in A_s,\ s \in S\,,$$

and

$$v(s) \leqslant 0\,, \quad s \in S\,.$$

As illustrated by Example 7.11, the region bounded by the above constraints is not convex, so the above LP cannot be solved by the simplex algorithm. Kallenberg (1983) provides an algorithm based on average reward methods that is applicable.

8. Undiscounted Markov decision problems—II

This section presents the theory of Markov decision problems with average and sensitive optimality criteria. Whittle (1983, p. 118) summarizes the difficulty in analyzing such problems as follows:

"The field of average cost optimization is a strange one. Counterexamples exist to almost all the natural conjectures and yet these conjectures are the basis of a proper intuition and are valid if reformulated right or if natural conditions are imposed."

Because results for these criteria are highly dependent on the structure of the Markov chains induced by stationary policies; the reader is referred to Chapter 2 on stochastic processes or to a basic text such as Kemeny and Snell (1960) for basic definitions. Emphasis throughout Section 8 will be problems with finite state spaces, but extensions to countable state problems will also be considered.

8.1. Markov chains, Markov reward processes and evaluation equations

Let $\{X_t, t = 1, 2, \ldots\}$ be a Markov chain with state space S and transition probability matrix P with entries $p(j|s)$. Elements of the n-step transition probability matrix P^n are denoted by $p^n(j|s)$. The expression, *chain structure* of a Markov chain, refers to the nature of its decomposition into classes. A Markov chain is said to be *unichain* if it contains one recurrent class plus additional transient states and *irreducible* if it consists of a single (necessarily recurrent) class. Otherwise a Markov chain is said to be *multichain*.

8.1.1. Some matrix theory

Two matrices that are derived from P play a particularly important role in MDP theory. Define the *limiting matrix* P^* by

$$P^* = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} P^{t-1} \tag{8.1}$$

where the limit is pointwise in S. When P is aperiodic,

$$P^* = \lim_{N\to\infty} P^N .$$

The *deviation matrix* H_P is defined by

$$H_P = (I - (P - P^*))^{-1}(I - P^*) . \tag{8.2}$$

It is the Drazin generalized inverse of $I - P$ (Lamond, 1986, Lamond and Puterman, 1989) and plays an important role in the modern theory of Markov chains. Related to the deviation matrix is the *fundamental matrix* $Z_P = (I - (P - P^*))^{-1}$ (Kemeny and Snell, 1960).

The chain structure of the Markov chain determines the form of P^* and H_P. If the chain is irreducible, P^* has identical rows with strictly positive entries. The rows are the stationary distribution of the Markov chain and each entry is the reciprocal of the long run frequency of the system being in the corresponding state. When the system is unichain with recurrent class R, for all $s \in S$,

$$p^*(j|s) = \begin{cases} p^*(j), & j \in R, \\ 0, & j \notin R. \end{cases}$$

with $p^*(j)$ strictly positive. When X_t is multichain, P^* has strictly positive entries corresponding to states within the same recurrent class, zero entries if states are in different classes or are both transient and positive values for at least one entry whenever the origin state is transient and the destination state is recurrent.

8.1.2. The average reward, bias and Laurent series expansion

For a Markov reward process (MRP), i.e., a Markov chain together with a (expected) reward function r defined on S, the average reward or gain is

$$g(s) = \lim_{N\to\infty} \frac{1}{N} E_s\left\{\sum_{t=1}^{N} r(X_t)\right\} \tag{8.3}$$

where lim inf or lim sup replaces the limit in (8.3) when necessary (see Section 5). Evaluating the expectation in (8.3) and expressing the result in matrix terms yields

$$g = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} P^{t-1} r = P^* r. \tag{8.4}$$

Combining (8.4) and the above results on the structure of P^* yields the following important result about the form of g.

Proposition 8.1. *If s and j are in the same recurrent class, $g(s) = g(j)$. Further, if the chain is irreducible or unichain, $g(s)$ is constant.*

Consequently, in the irreducible and unichain cases, the average reward can be expressed as $g1$ where g is a scalar and 1 is a vector of ones. In the multichain case, the gain is constant on each recurrent class. This distinction has a major influence on the resulting theory.

In the discounted case, Proposition 6.1 shows that the expected total discounted reward is the unique solution of the linear system $v = r + \lambda Pv$ and consequently $v = (I - \lambda P)^{-1} r$. Analogous results are now developed for the

average reward case. They are based on the relationship between the discounted and average reward of Blackwell (1962) and Veinott (1969).

For this analysis, it is convenient to parameterize the problem in terms of the interest rate ρ instead of in terms of the discount rate λ. They are related by $\lambda = (1+\rho)^{-1}$ or $\rho = (1-\lambda)\lambda^{-1}$. The quantity $1+\rho$ is the value of 1 unit one period in the future. Transforming to this scale yields

$$\begin{aligned} v_\lambda &= (I - \lambda P)^{-1} r = (1+\rho)((1+\rho)I - P)^{-1} r \\ &= (1+\rho)(\rho I + (I-P))^{-1} r . \end{aligned} \tag{8.5}$$

The quantity $(\rho I + (I-P))^{-1}$ is called the *resolvent of* $I - P$ and has a Laurent series expansion (Hille and Phillips, 1957) around $\rho = 0$ as follows.

Theorem 8.2. *Let ν be the eigenvalue of P less than one with largest modulus. If $0 \leq \rho \leq 1 - |\nu|$, then*

$$(\rho I + I - P)^{-1} = \rho^{-1} P^* + \sum_{n=0}^{\infty} (-\rho)^n H_P^{n+1} \tag{8.6}$$

and

$$v_\lambda = (1+\rho)\left[\rho^{-1} y_{-1} + \sum_{n=0}^{\infty} (-\rho)^n y_n\right] \tag{8.7}$$

where

$$y_{-1} = P^* r \quad \text{and} \quad y_n = H_P^{n+1} r , \quad n = 0, 1, \ldots .$$

The expansion in (8.7) is the Laurent series expansion of the expected discounted reward and is obtained by multiplying both sides of (8.6) by r. Derivations of (8.6) in the finite state case appear in Miller and Veinott (1969) and Lamond and Puterman (1989) and for the countable state case in Hordijk and Sladky (1977). Note that $y_{-1} = g$.

The quantity $y_0 = H_P r$ is important for the analysis of MDP's with average reward criteria and will be denoted by h. It is called the *bias* or *transient reward* and can be interpreted by expanding $(I - (P - P^*))^{-1}$ in a Neumann series to obtain

$$\begin{aligned} h_P &= H_P r = (I - (P - P^*))^{-1}(I - P^*) r \\ &= \sum_{n=0}^{\infty} (P - P^*)^n (I - P^*) r = \sum_{n=0}^{\infty} (P^n - P^*) r \\ &= \sum_{n=0}^{\infty} P^n (r - g) = E\left\{ \sum_{n=0}^{\infty} [r(X_n) - g(X_n)] \right\} . \end{aligned} \tag{8.8}$$

The representation in (8.8) enables h_P to be interpreted as the expected total reward for an MRP with reward $r - g$. If P is aperiodic, the distribution of X_n converges to a limiting distribution, so eventually $r(X_n)$ and $g(X_n)$ will differ by very little. Thus h can be thought of as the expected total reward 'until convergence' or the expected total reward during the 'transient' phase of the chains evolution.

An alternative representation for h_P due to Howard (1960) and discussed by Denardo (1973) provides further insight into the interpretation of the bias. Define v_N as the total expected N period reward for an MRP with terminal reward zero. That is,

$$v_N = \sum_{n=0}^{N-1} P^n r .$$

Now using the next to last equality in (8.8) yields

$$h_P = \sum_{n=0}^{N-1} P^n r - Ng + \sum_{n=N}^{\infty} (P^n - P^*)r$$

so that

$$v_N = Ng + h_P + o(1) , \tag{8.9}$$

where $o(1)$ is a vector with components which approach zero pointwise as $N \to \infty$. Thus as N becomes large, for each $s \in S$, $v_N(s)$ approaches a straight line with slope $g(s)$ and intercept $h_P(s)$. When g is constant, $v_N(s) - v_N(j)$ approaches $h_P(s) - h_P(j)$, so that h_P is the asymptotic relative difference of starting the process in two states s and j. For this reason, h_P is often referred to as the *relative value*. Another immediate consequence of this representation is that $v_N - v_{N-1}$ grows at rate g for N large. This justifies calling g the gain of the process.

For optimization, policies will be initially compared on the basis of their respective gain rates. If several policies have identical gain rates then (8.9) implies that a decision maker would prefer the policy with the largest bias. Consequently the bias could be used to break ties. Veinott (1974) provides additional insight into the interpretation of the bias as well as the higher order terms in the Laurent series expansion of v_λ.

In the context of average reward Markov decision processes it will suffice to use Blackwell's (1962) truncated Laurent expansion of v_λ which is a consequence of (8.7).

Corollary 8.3.

$$v_\lambda = (1 - \lambda)^{-1} g + h + f(\lambda) \tag{8.10}$$

where $f(\lambda)$ is a vector which converges to zero componentwise as $\lambda \uparrow 1$.

The following corollary is useful for extending existence results from the discounted case to the average reward case (Derman, 1970, pp. 25–28) and establishing structural results for average reward problems from those for discounted problems. It follows immediately by multiplying both sides of (8.10) by $1 - \lambda$ and passing to the limit.

Corollary 8.4.

$$g = \lim_{\lambda \uparrow 1} (1 - \lambda) v_\lambda \,. \tag{8.11}$$

8.1.3. The evaluation equations

Representation (8.7) can be used to establish the average reward and sensitive policy evaluation equations. Substitution of (8.7) into the discounted reward evaluation equation expressed as

$$r + [(P - I) - \rho I]\{(1 + \rho)^{-1} v_\lambda\} = 0$$

and equating terms with like powers of ρ yields the following result (Veinott, 1969). The converse follows by multiplying the $(n + 1)$th equation by P^* and adding it to the nth equation.

Theorem 8.5. *The coefficients of the Laurent series expansion of v_λ satisfy the system of equations*

$$(P - I) y_{-1} = 0 \,, \tag{8.12}$$

$$r - y_{-1} + (P - I) y_0 = 0 \,, \tag{8.13}$$

$$-y_n + (P - I) y_{n+1} = 0 \,, \quad n \geqslant 0 \,. \tag{8.14}$$

Conversely, if $w_{-1}, w_0, \ldots, w_m, w_{m+1}$ satisfy equation (8.12), (8.13) *and* (8.14) *for $n = 0, 1, \ldots, m$, $m \geqslant 0$ then $w_{-1} = y_{-1}$, $w_0 = y_0, \ldots, w_{m-1} = y_{m-1}$, $w_m = y_m$ and w_{m+1} is unique up to a vector in the null space of $I - P$.*

The following two corollaries give the reduced form of Theorem 8.5 that will be directly applicable to average reward problems. The first gives the equations to be solved to determine the average reward for transition probability matrices with general chain structure.

Corollary 8.6. *The vectors g and h satisfy*

$$(P - I) g = 0 \tag{8.15}$$

and

$$r - g + (P - I) h = 0 \,. \tag{8.16}$$

Conversely if g' and h' satisfy (8.15) *and* (8.16) *then $g' = g$ and $h' = h + w$ where $(P - I)w = 0$.*

In the case that P is unichain, any solution of (8.15) is a constant vector so that the evaluation equation simplifies further. In this case g will be written as $g1$ where g is a scalar and 1 is a vector of 1's of appropriate dimension. Corollary 8.7 is the basis for policy evaluation in the unichain and recurrent cases and serves as the basis for the optimality equation in these settings.

Corollary 8.7. *Suppose P is unichain. Then the average reward and the bias satisfy*

$$r - g1 + (P - I)h = 0 . \tag{8.17}$$

Conversely if the scalar g' and the vector h satisfy (8.17), *then $g' = g$ and $h' = h + w$ where $(P - I)w = 0$.*

To implement and analyze policy improvement algorithms in the average reward case requires a unique specification of h. Conditions which ensure this are discussed in Section 8.3 and 8.8.

8.2. The average reward optimality equation—The unichain case

The section presents the optimality equation for Markov decision problems with average reward criteria under the assumption that transition matrices corresponding to all decision rules are unichain. This will be referred to as the unichain case. A *single* optimality equation is sufficient to characterize optimal policies and their average rewards making results simpler than in the multichain case where a *pair* of nested functional equations is necessary.

The development herein parallels that in Section 6.1 where possible. It is assumed throughout this section that the rewards and transition probabilities are stationary, that the action space, rewards and transition probabilities are such that appropriate maxima are attained and the state space is either finite or countable. Results which hold under weaker conditions than unichainicity will be noted.

Recall that a policy π^* is average or gain optimal if

$$g^{\pi^*} = \lim_{n\to\infty} \frac{1}{n} v_n^{\pi^*} \geqslant \lim_{n\to\infty} \frac{1}{n} v_n^{\pi} = g^{\pi} \quad \text{for all } \pi \in \Pi .$$

When the limits above do not exist, either weak or strong average optimal policies (Section 5) are sought.

The optimality equation in a unichain average reward MDP is given by

$$0 = \max_{a\in A_s} \left\{ r(s,a) - g + \sum_{j\in S} p(j\,|\,s,a)h(j) - h(s) \right\}, \quad s \in S . \tag{8.18}$$

It can be expressed in matrix–vector and operator notation as

$$0 = \max_{d \in D} \{r_d - g1 + (P_d - I)h\} \equiv B(g, h) . \tag{8.19}$$

As in previous sections, the maximum in (8.19) is componentwise. When D consists of a single decision rule, this equation reduces to (8.17). Recall that V denotes the set of bounded real valued functions on S.

The following theorem is the average reward counterpart of Theorem 6.2 (cf. Hordijk, 1974).

Theorem 8.8. *Suppose there exist a scalar g and an $h \in V$ that satisfy*

$$0 \geqslant \max_{d \in D} \{r_d - g1 + (P_d - I)h\} . \tag{8.20}$$

Then

$$g \geqslant \sup_{\pi \in \Pi} \left[\limsup_{n \to \infty} \frac{1}{n} v_n^\pi \right] .$$

If instead there exist a scalar g and an $h \in V$ that satisfy

$$0 \leqslant \max_{d \in D} \{r_d - g1 + (P_d - I)h\} \tag{8.21}$$

then

$$g \leqslant \sup_{\pi \in \Pi} \left[\liminf_{n \to \infty} \frac{1}{n} v_n^\pi \right] .$$

Combining (8.20) and (8.21) yields:

Theorem 8.9. *If equation* (8.19) *has a solution $(g, h) \in R^1 \times V$, then*
(a) *there exists a scalar g^* satisfying*

$$g^* = \sup_{\pi \in \Pi} \left[\lim_{n \to \infty} \frac{1}{n} v_n^\pi \right]$$

and
(b) *g is unique and equals g^*.*

The following result is that the optimality equation determines average optimal stationary policies. A decision rule d_h is said to be *h-improving* if

$$r_{d_h} - g1 + (P_{d_h} - I)h = \max_{d \in D} \{r_d - g1 + (P_d - I)h\} \tag{8.22}$$

or equivalently

$$r_{d_h} + P_{d_h} h = \max_{d \in D} \{r_d + P_d h\} .$$

Theorem 8.10. *Suppose there exist a scalar* g^* *and an* $h^* \in V$ *which satisfy* (8.19) *and* d^* *is* h^**-improving, then*

$$g^{d^*} = \max_{\pi \in \Pi} g^{\pi} = g^* .$$

This result can be restated as follows. If the optimality equation possesses a solution (g^*, h^*), then g^* is unique and any stationary policy which uses an h^*-improving decision rule every period is strongly average optimal. It remains to show that a solution to the optimality equation exists.

Three approaches have been used to establish existence of solutions to the average reward optimality equation in the unichain case. They are:

(1) Policy iteration (Howard, 1960).

(2) Extensions of results for the discounted case (Theorem 6.6) obtained by letting λ increase to 1 and using representations for the discounted reward such as (8.10) and (8.11) (Taylor, 1965).

(3) Fixed point theorems (Federgruen and Schweitzer, 1984b).

A sufficient condition for the existence of solutions to (8.19) is Ross's (1968a) following generalization of Taylor (1965).

Theorem 8.11. *Suppose there exist a finite* N *and an* $s_0 \in S$ *such that*

$$|v_\lambda^*(s) - v_\lambda^*(s_0)| < N \tag{8.23}$$

for all $s \in S$ *and* $0 < \lambda < 1$. *Then there exist an* $h^* \in V$ *and a scalar* g^* *which satisfy* (8.19) *and*

$$g^* = \lim_{\lambda \uparrow 1} (1 - \lambda) v_\lambda^*(s_0) . \tag{8.24}$$

Sufficient conditions for (8.23) to hold include:

(a) S is finite, A_s is finite for each $s \in S$ and every stationary policy is unichain;

(b) S is finite, A_s is compact for each $s \in S$, $r(s, a)$ and $p(j \mid s, a)$ are continuous in a and every stationary policy is unichain;

(c) S is finite and the set of transition matrices corresponding to stationary policies is a *communicating system* (Bather, 1973), that is, for every pair of states s, j there exists a decision rule δ and an integer $n \geq 1$ such that $P_\delta^n(j \mid s) > 0$;

(d) S is countable, rewards are uniformly bounded and the expected number of transitions to reach state s_0 from s is uniformly bounded for all policies and states s.

The results of this section can be summarized in the following theorem.

Theorem 8.12. *Suppose* (8.23) *holds. Then*:

(a) *there exists a unique* g^* *and* $h^* \in V$ *which satisfy the optimality equation* (8.19),

(b) *there exists* h^**-improving decision rules, and*

(c) *a stationary policy which uses an h^*-improving decision rule is strongly average optimal.*

Theorem 8.12 is valid in the countable state case under weaker conditions than (8.23) (Federgruen, Hordijk and Tijms, 1978, 1979). Related work includes Hordijk (1974), Wijngaard (1977), Federgruen, Schweitzer and Tijms (1983) and Deppe (1984).

The assumption that solutions to the optimality equation are bounded is crucial for the existence of average optimal stationary policies. Counterexamples have been provided by Fisher and Ross (1968) and Ross (1983), Bather (1973), Sheu and Farn (1980) and Schweitzer (1985).

8.3. Policy iteration in the unichain case

Policy iteration is an efficient procedure for solving the optimality equation and finding optimal policies in MDP's with average reward criterion. It generates a sequence of stationary policies with monotonically non-decreasing gains. It also is an important theoretical device for establishing existence of solutions to the average reward optimality equations.

8.3.1. The algorithm

This algorithm was developed by Howard (1960) for finite state and action MDP's. He demonstrated finite convergence under the assumption that all policies are recurrent. In the countable state case, Derman (1966) used policy iteration to constructively show the existence of a solution to the optimality equation (8.19) under the assumptions that all states are recurrent under each stationary policy and that the reward, gain and bias are uniformly bounded on the set of stationary policies. Federgruen and Tijms (1978), Hordijk and Puterman (1987) and Dekker (1985) demonstrated the convergence of policy iteration for problems with compact action spaces.

The Policy Iteration Algorithm.

1. Set $n = 0$ and select an arbitrary decision rule $d_n \in D$.
2. (Policy evaluation) Obtain g_{d_n} and an h_{d_n} by solving

$$0 = r_{d_n} - g1 + (P_{d_n} - I)h \,. \tag{8.25}$$

3. (Policy improvement) Choose d_{n+1} to satisfy

$$r_{d_{n+1}} + P_{d_{n+1}} h_{d_n} = \max_{d \in D} \{r_d + P_d h_{d_n}\} \tag{8.26}$$

setting $d_{n+1} = d_n$ if possible.

4. If $d_{n+1} = d_n$, stop and set $d^* = d_n$. Otherwise increment n by 1 and return to 2.

The above algorithm yields a sequence of decision rules $\{d_n\}$ and corresponding gains $\{g_{d_n}\}$. The relative value functions $\{h_{d_n}\}$ determined by solving (8.25) are unique up to an additive constant. The choice of the additive

constant has no effect on the maximizing decision rule in (8.26) since for any h satisfying (8.25) and any constant c,

$$r_d + P_d(h + c1) = r_d + P_d h + c1$$

for all $d \in D$. Computationally, it is convenient to set

$$h_{d_n}(s_0) = 0 \tag{8.27}$$

for an arbitrarily selected s_0. Solutions determined by different choice of s_0 differ by a constant. When h_{d_n} is determined by (8.27) it is denoted by $h^{\mathrm{RV}}_{d_n}$, the superscript RV denoting relative value.

To implement the evaluation step under condition (8.27), solve the linear system

$$r = (Q_{s_0})w \tag{8.28}$$

where Q_{s_0} is the matrix $I - P$ with the column corresponding to state s_0 replaced by a column of 1's. The solution of (8.28) is unique, satisfies (8.27) and has g_{d_n} as its s_0th component. It can be obtained by Gaussian elimination or any appropriate iterative method. From a theoretical perspective, the Blackwell (1962) specification that

$$P^*_{d_n} h_{d_n} = 0 \tag{8.29}$$

is more convenient since it ensures that

$$h_{d_n} = H_{P_{d_n}} r_{d_n}$$

as in Section 8.1.2. When h is obtained from (8.25) and (8.29), it will be denoted by $h^{\mathrm{B}}_{d_n}$, the superscript B denoting bias. It is easy to see that

$$h^{\mathrm{RV}}_{d_n} = h^{\mathrm{B}}_{d_n} - h^{\mathrm{B}}_{d_n}(s_0)1 .$$

Veinott (1969) provides a method for finding $h^{\mathrm{B}}_{d_n}$ without computing $P^*_{d_n}$.

8.3.2. Convergence of policy iteration

Convergence in the finite state and action case is a consequence of the lexicographic monotonicity of the iterates of the above algorithm and the finiteness of the set of stationary policies. If improvement occurs in a recurrent state under stationary policy d_{n+1}, the gain for the improved policy is greater than the gain for the previous policy (Howard, 1960). This is formally stated as:

Proposition 8.13. *Suppose d_{n+1} is determined in step 3 of the policy iteration algorithm. Then the following hold:*

(a) $$g_{d_{n+1}}1 = g_{d_n}1 + P^*_{d_{n+1}} B(g_{d_n}, h_{d_n}) . \tag{8.30}$$

(b) *If $B(g_{d_n}, h_{d_n})(s) > 0$ for a state s which is recurrent under d_{n+1}, then $g_{d_{n+1}} > g_{d_n}$.*

(c) *If $B(g_{d_n}, h_{d_n})(s) = 0$ for all states s which are recurrent under d_{n+1}, then $g_{d_{n+1}} = g_{d_n}$.*

Representation (8.30) (Hordijk and Puterman, 1987) can be thought of as a 'Newton method' representation for the gains at sucessive steps of the policy iteration. An immediate consequence of parts (b) and (c) is the following convergence result.

Theorem 8.14. *If all states are recurrent under every stationary policy and the sets of states and actions are finite, then policy iteration converges in a finite number of iterations.*

When there are transient states associated with some (or all) stationary policies, additional analysis is based on:

Proposition 8.15. *Suppose d_n is determined in the improvement step of the policy iteration algorithm and h_{d_n} is any solution of* (8.25). *Then*

(a) $$h^{\mathrm{B}}_{d_{n+1}} = h^{\mathrm{B}}_{d_n} - P^*_{d_{n+1}} h^{\mathrm{B}}_{d_n} + H_{d_{n+1}} B(g_{d_n}, h_{d_n}) . \tag{8.31}$$

(b) *If $B(g_{d_n}, h_{d_n})(s) = 0$ for all s that are recurrent under d_{n+1} and $B(g_{d_n}, h_{d_n})(s_0) > 0$ for some s_0 which is transient under d_{n+1}, then*

$$h^{\mathrm{B}}_{d_{n+1}} > h^{\mathrm{B}}_{d_n} .$$

(c) *If $B(g_{d_n}, h_{d_n})(s) = 0$ for all s that are recurrent under d_{n+1} and $B(g_{d_n}, h_{d_n})(s) = 0$ for all s which are transient under d_{n+1}, then*

$$h^{\mathrm{B}}_{d_{n+1}} = h^{\mathrm{B}}_{d_n} .$$

The result in Proposition 8.15(b) means that if there is no improvement in states which are recurrent under the new policy and an improvement in a state which is transient under the new policy, then the bias of the new policy will be strictly greater than that of the previous policy. Surprisingly, this result does not imply that the relative values are monotone increasing. Thus at sucessive iterates, the algorithm produces a stationary policy with a larger gain and if this is not possible then a policy with a larger bias. If neither of these alternatives are possible the algorithm terminates.

Theorem 8.16. *Suppose all stationary policies are unichain and the sets of states and actions are finite. Then policy iteration converges in a finite number of iterations.*

The above results provide additional insight to the behavior of the iterates of policy iteration. If s is recurrent and j is transient under action a, $P(j|s, a) = 0$. Consequently, once the optimality equation is satisfied on all states that are recurrent under a decision rule δ, which attains the maximum in the improvement step, there will be no future changes in the gain. Consequently any stationary policy which agrees with P_δ on its recurrent states is average optimal. Since, in subsequent iterations the bias is increased in transient states of the maximizing policy until no further improvement is possible, one might suspect that policy iteration terminates with a policy that is *bias-optimal*, that is it has the largest bias among all policies with the same gain as δ. This supposition is false (Example 2 in Denardo 1973 in which two policies have the same gain but different recurrent classes). What is attained in this case is the policy with the largest bias among policies which have the same recurrent class as δ. To find bias optimal policies requires solution of an additional optimality equation (Section 8.9).

8.4. Value iteration

This section provides sufficient conditions for convergence of value iteration in the finite state case. The development is in the spirit of the survey articles by Federgruen and Schweitzer (1978) and Schweitzer and Federgruen (1980). Books by van der Wal (1981), Whittle (1983) and Bertsekas (1987) provide valuable insight. Other important contributions include White (1963), Brown (1965), Lanery (1967) and Bather (1973).

8.4.1. Convergence

Value iteration in the undiscounted case is based on the operator

$$Tv = \max_{d \in D} [r_d + P_d v] \tag{8.32}$$

on V, the space of bounded real valued functions on S. This operator is not a contraction and consequently the iterative scheme $v^{n+1} = Tv^n$ is not necessarily convergent. The quantity v^n is the maximum expected total n-period return when the terminal reward is v^0. For a fixed decision rule, v^n can be expressed as

$$\begin{aligned} v^n &= \sum_{m=0}^{n-1} P_d^m r_d + P_d^n v^0 \\ &= \sum_{m=0}^{n-1} (P_d^m - P_d^*) r_d + n P_d^* r_d + P_d^n v^0 . \end{aligned} \tag{8.33}$$

When P_d is aperiodic, $\lim_{n\to\infty} P_d^n = P_d^*$ so it follows from (8.8) that for n sufficiently large,

$$v^n = H_d r_d + n g_d + P_d^* v^0 . \tag{8.34}$$

Consequently one might conjecture that

$$L = \lim_{n\to\infty} \{v^n - n g^*\} \tag{8.35}$$

always exists. The following simple example shows this conjecture is false.

Example 8.17. Let $S = \{1, 2\}$ and suppose there is a single decision rule d, with

$$r_d = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{and} \quad P_d = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Then clearly $g_d = 0$ and if

$$v^0 = \begin{bmatrix} a \\ b \end{bmatrix},$$

then

$$v^n = P^n v^0 = \begin{cases} \begin{bmatrix} a \\ b \end{bmatrix}, & n \text{ even}, \\ \begin{bmatrix} b \\ a \end{bmatrix}, & n \text{ odd}. \end{cases}$$

Thus unless $a = b = 0$, $\lim_{n\to\infty} \{v^n - n g^*\}$ does not exist, but for any choice of a and b, both $\lim_{n\to\infty} v^{2n}$ and $\lim_{n\to\infty} v^{2n+1}$ exist.

In this example, states 1 and 2 are both recurrent but each is periodic with period 2. This suggests that periodicity causes problems for the convergence of value iteration in average reward problems.

When the limit in (8.35) exists, value iteration can be used to solve the MDP because:

(a) for N sufficiently large,

$$v^N - v^{N-1} \approx L + N g^* - (L + (N-1) g^*) = g^* ;$$

(b) for n sufficiently large,

$$v^N - N g^* \approx h^*$$

where h^* is a solution of the optimality equation (8.19);

(c) for N sufficiently large, stationary policies which are v^N-improving are optimal; and

(d) upper and lower bounds for the optimal gain are available.

When S is finite, the following result summarizes conditions when the limit in (8.35) exists.

Theorem 8.18. *Let S be finite and let $v^{n+1} = Tv^n$. Then the limit in* (8.35) *exists for any $v^0 \in V$ if any of the following conditions hold*:

(a) *For all $s \in S$, $P(j|s,a) > 0$ for all $a \in A_s$ and $j \in S$* (Bellman, 1957).

(b) *There exists a state s_0 and an integer $\nu \geq 1$ such that*

$$(P_{d_1} P_{d_2} \cdots P_{d_\nu})(s, s_0) \geq \delta > 0 \tag{8.36}$$

for any decision rules $d_1, d_2, \ldots, d_\nu$ and all $s \in S$ (White, 1963).

(c) *For any decision rule d, P_d is aperiodic* (Schweitzer, 1965).

(d) *Every stationary optimal policy is unichained and at least one of them is aperiodic* (Denardo, 1973, Schweitzer and Federgruen, 1977).

Condition (a) is the least general; it implies (b)–(d). Condition (b) is stronger than condition (c); conditions (c) and (d) are distinct. Condition (a) ensures that the underlying Markov chain is irreducible and aperiodic under every policy. Condition (b) means that there exists a state s_0 which can be reached in ν transitions with positive probability from each starting state for any policy. Condition (d) allows non-optimal policies to be periodic and have arbitrary structure but places restriction on the class of optimal policies while condition (c) requires that all stationary policies be aperiodic but possibly multichained.

The transition matrix in Example 8.17 violates each of the conditions in Theorem 8.18. However, if P_d is replaced by

$$P_d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

then condition (c) is satisfied and v^n is (trivially) convergent for any v^0. In practice, conditions (a) and (c) are easiest to check.

Schweitzer and Federgruen (1977) provide necessary and sufficient conditions for the limit (8.35) to exist for all v^0, generalizing Brown (1965) and Lanery (1967).

8.4.2. Determination of optimal policies

An important practical issue in finite state problems is under what conditions value iteration can be used to determine an average optimal stationary policy. The following result provides a theoretical solution to this problem.

Theorem 8.19. *Suppose the limit in* (8.35) *exists. Then*:

(a) *there exists an integer n_0 such that for all $n \geq n_0$, any v_n-improving decision rule is average optimal* (Odoni, 1969; van der Wal, 1981), *and*

(b) *if d is v^n-improving for infinitely many n, then d is average optimal* (Brown, 1965).

This result can be restated as follows. If

$$r_d + P_d v^n = Tv^n$$

for $n \geq n_0$ or for infinitely many n, then $g_d = g^*$.

The results in this theorem cannot be used in computation because no bounds are available for n_0; i.e., the second condition is not verifiable. Even when the limit in (8.35) exists, asymptotic behavior of the sets of v^n-improving decision rules can be eratic; they can be strict subsets of the set of maximal gain rules for every n and they can oscillate periodically (Brown, 1965) or even aperiodically (Bather, 1973) within the set of maximal gain decision rules. Thus, convergence of the set of v^n-improving decision rules cannot be used as a termination condition for value iteration. When the limit in (8.35) fails to exist, Lanery (1967) provides an example where non-maximal gain decision rules appear infinitely often in the sequence of v^n-improving decision rules.

Result (a) in Theorem 8.19 is a turnpike theorem for the expected total reward criterion (Section 6.2.4). Whenever the horizon is known to exceed n_0 periods, then any v^n-improving decision rule ($n \geq n_0$) is optimal in the first period for a finite horizon problem under expected total reward criterion.

8.4.3. Bounds on the gain

In the unichain case, bounds on the optimal gain rate were given by Odoni (1969) and Hastings (1976). They are valid in general and are given in the following proposition.

Proposition 8.20. *Suppose h is bounded. Then*

$$L(h) \equiv \min_{s \in S} (Th(s) - h(s)) \leq g_d \leq g^* \leq \max_{s \in S} (Th(s) - h(s)) \equiv U(h) \tag{8.37}$$

where g_d is any h-improving decision rule.

These bounds have been applied to value iteration as follows (Platzman, 1977).

Theorem 8.21. *Suppose g^* is constant and $v^n = Tv^{n-1}$. Then for all n,*

$$\min_{s \in S} \{v^n(s) - v^{n-1}(s)\} \leq g_{d_n} \leq g^* \leq \max_{s \in S} \{v^n(s) - v^{n-1}(s)\} \tag{8.38}$$

where d_n is any v^{n-1}-improving decision rule. Further, if the limit (8.35) *exists, then*

$$L(v^{n-1}) \leq L(v^n) \leq g_{d_n} \leq g^* \leq U(v^n) \leq U(v^{n-1}) \tag{8.39}$$

and $\lim_{n \to \infty} \{U(v^n) - L(v^n)\} = 0$.

That g^* is constant is difficult to check a priori. Conditions which imply it include:

(a) all stationary policies are unichain,

(b) the set of policies is a communicating class, or

(c) S is *simply connected*, that is it can be partitioned into a communicating class and a set of states which are transient under all policies (Platzman, 1977).

Thus when all stationary policies are aperiodic and g^* is constant, the upper and lower bounds converge to zero and are monotonic. This means that an ε-optimal policy can be found by stopping value iteration when $U(v^n) - L(v^n) < \varepsilon$. At termination, g_{d_n} is within ε of g^* and the statinary policy d_n is ε-optimal. When the limit (8.35) does not exist or the problem is multichain, then $\lim_{n\to\infty}\{U(v^n) - L(v^n)\}$ will not equal zero and the bounds cannot be used as a stopping criterion.

In contrast to the discounted case, these bounds cannot be used for elimination of suboptimal actions. Instead bounds on h^* are required. Hastings (1976) provided a procedure for temporary action elimination in average reward problems.

8.4.4. Variants of value iteration

When applying value iteration in the average reward case, there are two potential problems.

(1) The divergence of v^n might lead to numerical instability.

(2) The limit in (8.35) might not exist because of the periodicity of transition matrices corresponding to some policies.

White (1963) provides a convergent value iteration scheme. Instead of using $v^{n+1} = Tv^n$, his procedure renormalizes v^n after each iteration so that one component is zero. That is, a state s_0 is selected and successive iterates are defined for all $s \in S$ by

$$w^0(s) = v^0(s) - v^0(s_0) \quad \text{and} \quad w^{n+1}(s) = Tw^n(s) - Tw^n(s_0)\,. \tag{8.40}$$

This procedure is called *relative value iteration*. In investigating its convergence, it is convenient to use the span of a vector $v \in V$ as defined in Section 6.7. The following combines results of White (1963) and Platzman (1977).

Proposition 8.22. *let w^n be generated by* (8.40). *Then if g^* is constant*:

(a) $$w = \lim_{n\to\infty} w^n(s)\,, \quad s \in S\,,$$

exists;

(b) $g = w(s_0)$ *and* $h(s) = w(s)$ *satisfy* $B(g, h) = 0$; *and*

(c) *if* (8.36) *holds*,

$$\operatorname{sp}(w^{n+1} - w^n) \leq (1 - \delta)^{1/\nu} \operatorname{sp}(w^n - w^{n-1})\,, \tag{8.41}$$

where δ is defined in (8.36).

This result states that w^n converges to a solution of the average reward optimality equation and consequently finds the optimal gain g and a relative value vector h. Part (c) implies that when (8.36) holds, the convergence rate is linear with constant at least equal to $(1-\delta)^{1/\nu}$. Error bounds can be derived from (8.41) and appear in Bertsekas (1987, p. 321) which also includes a proof of this proposition under (8.36). Bounds like those in Theorem 8.22 are available for relative value iteration.

When some policies, especially those that are average optimal have periodic transition matrices, the limit in (8.35) need not exist for every starting value. Schweitzer (1971) proposed transforming the problem so that all decision rules have aperiodic transition matrices and the transformed problem has the same relative costs and optimal policies as the original problem. The importance of this transformation is that it ensures the limit (8.36) exists and value iteration is globally convergent. The data transformation can be combined with relative value iteration to obtain a scheme which is convergent for any unichain MDP (Bertsekas, 1987, pp. 323–324).

8.5. Linear programming—Unichain case

The linear programming approach for the average reward MDP's is due to de Ghellinck (1960) and Manne (1960) for problems in which the transition probability matrix for each stationary policy is ergodic. Other work in this area is by Denardo and Fox (1968), Denardo (1970), Derman (1970), Hordijk and Kallenberg (1979, 1980) and Kallenberg (1983). The last reference is particularly noteworthy.

The primal problem is based on the result in Theorem 8.8 that $g \geq g^*$ whenever there exist $(g, h) \in R^1 \times V$ satisfying

$$g \geq r_d + (P_d - I)h \tag{8.42}$$

for all $d \in D$. As a consequence of Theorem 8.9, g^* is the smallest g for which there exists an $h \in V$ satisfying (8.42). It follows that g^* is an optimal solution of the following primal linear programming problem.

Primal Linear Program.

$$\text{Minimize} \quad g$$

subject to

$$g \geq r(s,a) + \sum_{j \in S} p(j \mid s,a)h(j) - h(s), \quad a \in A_s, s \in S,$$

and g and h unconstrained.

Its dual problem is as follows.

Dual Linear Program.

$$\text{Maximize} \quad \sum_{s\in S}\sum_{a\in A_s} r(s,a)x(s,a)$$

subject to

$$\sum_{a\in A_s} x(j,a) - \sum_{s\in S}\sum_{a\in A_s} p(j|s,a)x(s,a) = 0\,, \quad j\in S\,,$$

$$\sum_{s\in S}\sum_{a\in A_s} x(s,a) = 1\,, \tag{8.43}$$

and $x(s,a)\geqq 0$, $a\in A_s$, $s\in S$.

By Theorem 8.11, there exists an optimal solution to the primal problem so that by the duality theorem of linear programming, the dual problem processes an optimal solution. A stationary average optimal policy for the MDP can be found by using the simplex algorithm to obtain an extreme point solution x^* of the dual problem and then choosing a decision rule d^* satisfying

$$d^*(s) = \begin{cases} a & \text{if } x(s,a)>0\,,\ s\in S^*\,, \\ \text{arbitrary} & \text{if } s\in S-S^*\,, \end{cases}$$

where

$$S^* = \{s\in S\colon x^*(s,a)>0 \text{ for some } a\in A_s\}\,. \tag{8.44}$$

For any decision rule d^* obtained in the above manner, $x^*(s,d^*)$ is an optimal solution to the dual problem and satisfies the equalities

$$x^*(s,d^*(s)) - \sum_{s\in S} p(j|s,d^*(s))x^*(s,d^*(s)) = 0\,, \quad j\in S\,,$$

and

$$\sum_{s\in S} x^*(s,d^*(s)) = 1\,. \tag{8.45}$$

These imply that x^* is the stationary probability distribution corresponding to the stationary policy d^*. Since x^* is zero on $S-S^*$, such states are transient under d^*; S^* is a recurrent class.

In the unichain case, there is not a one-to-one relationship between feasible solutions to the dual problem and stationary policies. Under the assumption that all stationary policies are recurrent, then the following result can be obtained.

Theorem 8.23. *Suppose that the transition probability matrix of every stationary policy is irreducible. Let x be any feasible solution to the dual problem, and define the randomized stationary policy d by*

$$P\{d(s) = a\} = \frac{x(s, a)}{\sum_{a' \in A_s} x(s, a')}, \quad a \in A_s, s \in S. \tag{8.46}$$

Then

$$x(s, a) = P\{d(s) = a\}\pi_d(s), \quad a \in A_s, s \in S, \tag{8.47}$$

is a feasible solution to the dual problem where π_d is the solution of

$$\pi P_d = \pi \quad \text{and} \quad \sum_{s \in S} \pi(s) = 1. \tag{8.48}$$

Further, given any randomized stationary policy d, $x(s, a)$ defined by (8.47) *is a feasible solution to the dual problem.*

This theorem states that the mapping (8.47) is a one-to-one mapping between stationary policies and solutions to the dual problem in the recurrent case. Note that $x(s, a)$ is the stationary probability that the system is in state s and action a is chosen when using either a prespecified stationary policy in which case π is computed by (8.48) or using the stationary policy with decision rule defined by (8.46). For irreducible problems, $S^* = S$, so in this case the arbitrariness of the policy selected by the algorithm is removed. Consequently policy iteration corresponds to the simplex method with block pivotting and the simplex algorithm corresponds to a version of policy iteration in which only the maximum improvement is selected.

For unichain problems, this relationship is not valid because the policy iteration algorithm terminates with policies with special structure (Section 8.3.2) while the policy selected by the LP method is arbitrary on a subset of the state space.

8.6. Modified policy iteration—Unichain case

For the average reward criterion, modified policy iteration is best thought of as a variant of value iteration. It was proposed by Morton (1971) and subsequently analyzed by van der Wal (1981) and Ohno (1985).

The algorithm to obtain an ε-optimal policy is as follows:

Modified Policy Iteration Procedure of order m.

1. Select $v^0 \in V$, set $n = 0$ and specify $\varepsilon > 0$.
2. Determine a v^n-improving decision rule d_{n+1}.
3. Compute $L(v^n)$ and $U(v^n)$ using (8.37). If $U(v^n) = L(v^n) < \varepsilon$, stop and set $d^\varepsilon = d_{n+1}$, otherwise continue.

4. Obtain v^{n+1} by

$$v^{n+1} = (T_{d_{n+1}})^{m+1} v^n \tag{8.49}$$

where

$$T_d v \equiv r_d + P_d v \,. \tag{8.50}$$

5. Increment n by 1 and return to 2.

Modified policy iteration of order 0 is value iteration; in this case step 4 is superfluous. When m is large the algorithm is similar, although not identical to policy iteration.

The following result (van der Wal, 1981) is proved by showing that the sequence of upper and lower bounds derived from (8.37) converge to g^*.

Theorem 8.24. *Suppose that S and A_s for each $s \in S$ are finite, that for some $\alpha > 0$, $P_d \geq \alpha I$ for all $d \in D$ and that all stationary policies are unichain. Then if $\{v^n\}$ is generated by modified policy iteration*:

(a) *$L(v^n)$ converges monotonically and exponentially fast to g^* and*

(b) *$U(v^n)$ converges exponentially fast to g^*.*

The condition on P_d is referred to as *strong aperiodicity*; without it the algorithm can cycle. The lower bound converges monotonically to g^* but the convergence of the upper bound is not necessarily monotone (cf. Example 9.2 in van der Wal, 1981). As a consequence of Theorem 8.24, the stopping criterion in step 3 is satisfied in a finite number of iterations. When implementing the algorithm, the bounds in 3 are evaluated at the same time as a v^n-improving decision rule is determined in step 2.

The asymptotic properties of the values generated by modified policy iteration have not been investigated and the convergence of the algorithm for non-finite problems or those with more general chain structures has not been demonstrated.

8.7. Average reward—Multichain case

When there are multiple chains, that is, the Markov chains corresponding to stationary policies have more than one recurrent class, two optimality equations are required to characterize optimal policies. Consequently, theory and algorithms are more complex than in the unichain case.

Howard (1960) provided a policy iteration algorithm for solving this system of equations. Other important references in this case include Blackwell (1962), Denardo and Fox (1968), Veinott (1969), Denardo (1970), Schweitzer and Federgruen (1978), Hordijk and Kallenberg (1979), Kallenberg (1983) and Federgruen and Schweitzer (1984a,b).

Results are not as complete as in the unichain case and few are available when the set of stationary policies is infinite. The assumption that the sets of states and actions are *finite* is required for results in this section. Discussion will focus on the optimality equation and policy iteration. The reader is referred to Denardo and Fox (1968), Denardo (1970), Derman (1970), Dirickx and Rao (1979) and Kallenberg (1983) for linear programming for multichain MDP's. Little is known about value iteration and modified policy iteration for multichain problems.

8.7.1. Multichain optimality equations

For multichain average reward MDP's, a pair of nested optimality equations is required to compute the optimal gain and determine optimal policies. They are given by

$$\max_{d \in D} \{(P_d - I)g\} = 0 \tag{8.51}$$

and

$$\max_{d \in E} \{r_d - g + (P_d - I)h\} = 0 \tag{8.52}$$

where $E = \{d \in D: P_d g = g\}$.

A solution to this pair of nested functional equations is a pair $(g, h) \in V \times V$ such that $P_d g \leq g$ for all $d \in D$ with equality holding for at least one $d \in D$ and

$$r_d - g + P_d h \leq h \tag{8.53}$$

for all $d \in D$ for which $P_d g = g$ with equality holding in (8.53) for at least one such d.

In the unichain case the first optimality equation above is redundant and $E = D$. This is because when all decision rules are unichain, $P_d g = g$ implies that g is a constant so that equation (8.51) is satisfied for all $d \in D$. The above reduces to (8.18) in the unichain case. If D replaces E in (8.52) then possibly a different decision rule attains the maximum in each equation.

Establishing that solutions to this pair of equations characterize average optimal policies is not as straightforward as in the unichain case. A proof is based on the existence of a Blackwell optimal stationary policy as defined in (5.11). When S is finite, a policy π^* is Blackwell optimal if there exists a $\lambda^*, 0 \leq \lambda^* < 1$, such that π^* is discount optimal for all $\lambda \in [\lambda^*, 1)$. The following important theorem is due to Blackwell (1962). An elegant nonconstructive proof using function theory was provided by Blackwell, a constructive proof is based on the policy iteration algorithm in Section 8.9.

Theorem 8.25. *There exists a Blackwell optimal policy which is stationary.*

A closely related result is the following.

Theorem 8.26. *If d^* is Blackwell optimal then*
(a) *d^* is average optimal in the class of all policies* (Derman, 1970), *and*
(b) *(g_d^*, h_d^*) satisfy the average reward optimality equations* (8.51) *and* (8.52).

The converse of this, that an average optimal policy is Blackwell optimal and a policy determined by a solution to the optimality equation is Blackwell optimal, holds when there exists a unique average optimal policy. Blackwell (1962) provides a simple example in which an average optimal policy is not Blackwell optimal. The above results establish the existence of a solution to the optimality equations. In the next subsection, policy iteration will be used to construct such a solution. The following theorem gives the optimality properties of solutions to the optimality equations.

Theorem 8.27. *Suppose (g^*, h^*) satisfies* (8.51) *and* (8.52) *and d^* attains the maximum in* (8.52) *at (g^*, h^*). Then for all $\pi \in \Pi$,*

$$\liminf_{\lambda \uparrow 1} (1-\lambda)(v_\lambda^{d^*} - v_\lambda^{\pi}) \geq 0 \tag{8.54}$$

and

$$g_{d^*} \geqslant g_\pi \,.$$

8.8. Multichain policy iteration

Policy iteration in the multichain case consists of an improvement and an evaluation step. In the improvement step, a decision rule is sought which provides a strict improvement in (8.51) and if none is available, in (8.52). When no improvement is possible in either equation the algorithm is terminated. Proof of convergence is based on the partial Laurent expansion (8.10) and was provided by Blackwell (1962). The algorithm is due to Howard (1960). It is implemented componentwise.

The Policy Iteration Algorithm—Multichain case.

1. Set $n = 0$ and select an arbitrary decision rule $d_n \in D$.
2. (Policy evaluation) Obtain g_{d_n} and an h_{d_n} by solving

$$0 = (P_{d_n} - I)g \quad \text{and} \quad 0 = r_{d_n} - g + (P_{d_n} - I)h \,. \tag{8.55}$$

3. (Policy improvement)
   (a) Choose $d_{n+1} \in D$ to satisfy

$$P_{d_{n+1}} = \max_{d \in D} \{P_d g_{d_n}\} \tag{8.56}$$

and set $d_{n+1} = d_n$ if possible. If $d_{n+1} = d_n$, let $E_n = \{d \in D\colon P_d g_{d_n} = P_{d_n} g_{d_n}\}$ and go to (b), otherwise increment n by 1 and return to 2.

(b) Choose $d_{n+1} \in E_n$ to satisfy

$$r_{d_{n+1}} + P_{d_{n+1}} h_{d_n} = \max_{d \in E_n} \{r_d + P_d h_{d_n}\} \tag{8.57}$$

setting $d_{n+1} = d_n$ if possible.

4. If $d_{n+1} = d_n$, stop and set $d^* = d_n$. Otherwise, increment n by 1 and return to 2.

The above algorithm yields a sequence of decision rules $\{d_n\}$ and corresponding gains $\{g_{d_n}\}$. The pair of matrix equations (8.55) uniquely determines the gain, but the relative values $\{h_{d_n}\}$ are only unique up to a u satisfying $(P_{d_n} - I)u = 0$. If P_{d_n} has k recurrent classes, then h_{d_n} will be unique up to k arbitrary constants which can be determined by setting $h_{d_n}(s) = 0$ for an arbitrary s in each recurrent class of P_{d_n} (Howard 1960). Blackwell's specification (8.29) can also be used, but is computationally prohibitive. Veinott (1969) provides a method for finding an h which satisfies (8.58). In practice, any h will do.

The improvement step of the algorithm consists of two phases. First improvement is attempted through the first optimality equation (8.56), that is a g_{d_n}-improving decision rule is sought. (Call a decision rule d' *g-improving* if $d' = \arg\max_{d \in D} \{P_d g\}$.) If no *strict* improvement is possible, an h_{d_n}-improving decision rule is found among all g_{d_n}-improving rules and if no strict improvement is possible, the iterations are stopped. Otherwise, the improved policy is evaluated at the subsequent iteration.

In the unichain case, the first equation in (8.55) and part (a) of the improvement step are redundant so that the algorithm reduces to the unichain policy iteration algorithm.

Proofs that policy iteration is convergent in the multichain case for finite state and action problems have been provided by Blackwell (1962) using the partial Laurent expansion and Denardo and Fox (1968) using detailed analysis of the chain structure. The next result shows the monotone nature of the multichain policy iteration algorithm and is the basis for a proof of convergence.

Proposition 8.28. *Let $d \in D$ and suppose either*

(a) *$d' \in D$ is strictly g_d-improving, or*

(b) *$d' \in D$ is g_d-improving, $P_{d'} g_d = P_d g_d$ and d' is strictly h_d-improving.*

Then

$$\liminf_{\lambda \uparrow 1} (1 - \lambda)(v_\lambda^{d'} - v_\lambda^d) > 0 . \tag{8.58}$$

As a consequence of this result, sucessive iterates of policy iteration satisfy

$$\liminf_{\lambda \uparrow 1} (1 - \lambda)(v_\lambda^{d_{n+1}} - v_\lambda^{d_n}) > 0 \tag{8.59}$$

until the algorithm terminates with $d_{n+1} = d_n$ at which point the optimality equations are satisfied. Finite convergence is ensured since there are only finitely many policies and the algorithm is monotone in the sense of (8.59). This yields the following result.

Theorem 8.29. *Suppose A_s for each $s \in S$ and S are finite. Then the policy iteration algorithm terminates in a finite number of iterations with a gain optimal stationary policy and a pair (g^*, h^*) which satisfy the optimality equations* (8.51) *and* (8.52).

Based on the above arguments, one might speculate that this implies that the gains of successive policies are strictly increasing. This is not the case because (8.59) does not exclude the possibility that the gains of two successive policies are identical and improvement occurs in the bias term.

Howard (1960, p. 69–74) and Denardo and Fox (1968, p. 477–479) have analyzed the behavior of the iterative process in detail and have shown that:

(a) the gains of successive policies are monotone non-decreasing,

(b) if improvement occurs in step 3a of the algorithm, then it can only be in transient states of d_{n+1}, in which case $g_{d_{n+1}}(s) > g_{d_n}(s)$ where s is transient under d_{n+1},

(c) if no improvement occurs in step 3a of the algorithm and it occurs in a recurrent state of d_{n+1}, in step 3b of the algorithm then $g_{d_{n+1}}(s) > g_{d_n}(s)$ where s is recurrent under d_{n+1}, and

(d) if no improvement occurs in step 3a of the algorithm and it occurs in a transient state of d_{n+1} in part 3b of the algorithm then $h_{d_{n+1}}(s) > h_{d_n}(s)$ where s is transient under d_{n+1}.

In the special case that all policies are communicating, Haviv and Puterman (1990) provide a modification of the unichain policy iteration algorithm which avoids the pair of optimality equations.

At present, non-trivial conditions which imply the convergence of policy iteration in the non-finite case are not available. Dekker (1985, pp. 109–110) provides an example with finite states and compact actions in which an infinite number of improvements occur in step 3a and converge to a suboptimal policy. In it, the limiting policy has different ergodic classes than the optimal policy and since improvements through 3a cannot create new ergodic classes, the algorithm will not converge to an optimal policy.

8.9. Sensitive discount optimality

If (g^*, h^*) is a solution of the multichain average reward optimality equations (8.51) and (8.52) then any stationary policy which uses an h^*-improving decision rule is average optimal. Such a policy has the greatest bias among all stationary policies with the same chain structure as d^* but simple examples (cf. Blackwell (1962), Denardo (1973)) show that such a policy need not have the largest bias among *all* average optimal policies.

A policy with the largest bias among all average optimal policies is said to be *bias-optimal*. Veinott (1966), Denardo (1970) and Kallenberg (1983) provide methods for obtaining such policies in the finite state and action setting. Sheu and Farn (1980) and Mann (1983) have also studied this criterion.

Since bias-optimal policies need not be unique, a decision maker might wish to have some way of selecting a 'best' bias-optimal policy. Veinott (1969) introduced the concept of sensitive discount optimality and using the Laurent series expansion (8.7), showed that it provided a link between average optimality, bias-optimality and Blackwell optimality. Contributors to this theory include Miller and Veinott (1969), Veinott (1974), Hordijk and Sladky (1977), Wijngaard (1977), van der Wal (1981), Federgruen and Schweitzer (1984a) and Dekker (1985).

This section presents the theory of sensitive discount optimality in the finite state and action case.

8.9.1. Discount optimality criteria

Recall from (5.12) that a policy π^* is said to be *n-discount optimal* for $n = -1, 0, 1, \ldots$ if

$$\liminf_{\lambda \uparrow 1} (1-\lambda)^{-n}[v_\lambda^{\pi^*} - v_\lambda^{\pi}] \geq 0$$

for all $\pi \in \Pi$. This criterion can be reexpressed in terms of the interest rate $\rho = 1/(1-\lambda)$ (Section 8.1.2) as follows:

$$\liminf_{\rho \downarrow 0} \rho^{-n}[v_\lambda^{\pi^*} - v_\lambda^{\pi}] \geq 0\,. \tag{8.60}$$

Let A be an arbitrary matrix. Then A is said to be *lexicographically positive*, written $A > 0$, if A is not identically 0 and the first non-zero entry in each row of A is strictly positive. If A and B are two matrices of the same dimension, A is said to be *lexicographically greater than* B written $A > B$ if $A - B$ is lexicographically positive.

Let d and e denote two stationary policies. Using (8.7), the difference in their discounted rewards can be written as

$$v_\lambda^d - v_\lambda^e = (1+\rho)\left\{\frac{g_d - g_e}{\rho} + h_d^B - h_e^B + \sum_{n=1}^{\infty} (-\rho)^n [y_n^d - y_n^e]\right\} \tag{8.61}$$

where y_n^δ is the nth term in the Laurent series expansion corresponding to stationary policy $\delta = d$ or e. Multiplying (8.61) by ρ and taking the lim inf as ρ decreases to zero implies that

$$\liminf_{\rho \downarrow 0} \rho[v_\lambda^d - v_\lambda^e] \geq 0 \tag{8.62}$$

if and only if $g_d \geq g_e$. When $g_d = g_e$, the limit in (8.62) will be zero, in which case

$$\liminf_{\rho \downarrow 0} [v_\lambda^d - v_\lambda^e] \geq 0 \tag{8.63}$$

if and only if $h_d^B \geq h_e^B$. if $g_d(s) > g_e(s)$ for some s in S, then (8.63) will hold with strictly inequality in component s, regardless of the values of $h_d^B(s)$ and $h_e^B(s)$ (these quantities are defined in Section 8.3.1).

Similarily, if $g_d = g_e$ and $h_d^B = h_e^B$, then the lim inf's in both (8.62) and (8.63) will be zero and

$$\liminf_{\rho \downarrow 0} \rho[v_\lambda^d - v_\lambda^e] \geq 0 \tag{8.64}$$

if and only if $y_1^d \geq y_1^e$. Conversely the sth component of (8.64) will be strictly positive if either

(a) $g_d(s) > g_e(s)$, or

(b) $g_d(s) = g_e(s)$ and $h_d^B(s) > h_e^B(s)$, or

(c) $g_d(s) = g_e(s)$ and $h_d^B(s) = h_e^B(s)$ and $y_1^d(s) > y_1^e(s)$.

These arguments can be repeated indefinitely to demonstrate that:

(1) The larger the value of n, the more selective the discount optimal criteria. That is, if D_n^* denotes the set of n-discount optimal stationary policies, then $D_{n-1}^* \supset D_n^*$ for $n = 0, 1, \ldots$.

(2) A policy is (-1)-discount optimal if it maximizes the average reward, 0-discount optimal if maximizes the bias among all average optimal policies, 1-discount optimal if it maximizes the third term in the Laurent expansion among all policies that are bias optimal, etc.

(3) A stationary policy d is n-discount optimal among the class of stationary policies if $v_\lambda^{d,n+2} > v_\lambda^{e,n+2}$ for all stationary policies, e, where $v_\lambda^{d,n+2}$ is the $|S| \times (n+2)$ matrix with columns given by y_m^d, $m = -1, 0, \ldots, n$.

Another important consequence of (8.61) is that a stationary policy is Blackwell optimal if $v_\lambda^{d,\infty} > v_\lambda^{e,\infty}$ for all stationary policies e. That is, a Blackwell optimal policy is n-discount optimal for all n and consequently is more selective than any of the discount optimality criteria. If for some finite N, D_N^* contains a single decision rule, then the stationary policy corresponding to that decision rule is n-discount optimal for all $n \geq N$ and is Blackwell optimal.

The following result is an immediate consequence of Theorem 8.25.

Theorem 8.30. *If A_s for each $s \in S$ and S are finite, then there exists a stationary n-discount optimal policy for each n.*

8.9.2. Optimality equations

Corresponding to the n-discount optimality criteria are a series of nested optimality equations extending those in Section 8.7. They are obtained by

substituting the Laurent expansion of an n-discount optimal policy into the discount optimality equation and equating terms in like powers of ρ to obtain

$$\max_{d \in D_{-1}} \{(P_d - I)y_{-1}\} = 0, \tag{8.65}$$

$$\max_{d \in D_0} \{r_d - y_{-1} + (P_d - I)y_0\} = 0, \tag{8.66}$$

and for $n \geq 1$,

$$\max_{d \in D_n} \{-y_{n-1} + (P_d - I)y_n\} = 0 \tag{8.67}$$

with $D_{-1} = D$ and for $n \geq 0$, $D_n = \{d \in D_{n-1}: d$ attains the max in equation $n-1$ at y_{n-1} when $(y_{-1}, y_0, y_1, \ldots, y_{n-1})$ satisfy equations $m = -1, 0, \ldots, n\}$. Note that equations (8.65) and (8.66) are the optimality equations for multichain average reward problems.

The following result links solutions of the optimality equations to the terms in the Laurent series expansion of Blackwell optimal policies (Dekker, 1985).

Theorem 8.31. *Let d be a Blackwell optimal policy. Then for all n, $(y^d_{-1}, y^d_0, \ldots, y^d_n)$ is a solution of the first $n+2$ optimality equations and is the unique solution to the first $n+3$ optimality equations where y_{n+1} is arbitrary. Further if $(y_{-1}, y_0, \ldots, y_n)$ is any solution to the first $n+2$ optimality equations, then any $d \in D_{n+1}$ is n-discount optimal.*

In the case of bias-optimality the results simplify as follows.

Corollary 8.32. *Suppose (y^*_{-1}, y^*_0, y^*_1) is a solution of equations (8.65)–(8.67) with $n = 1$. Then*

(a) *any decision rule d in D_1 which maximizes $P_d y^*_0$ is bias optimal, and*

(b) *y^*_{-1} and y^*_0 are unique and equal the gain and bias of any bias optimal policy.*

To compute n-discount optimal policies requires solution of the optimality equations. Policy iteration methods are the most direct. Linear programming methods are available for $n = -1$ and $n = 0$; Federgruen and Schweitzer (1984) have proposed a value iteration scheme.

8.9.3. Policy iteration

The Policy Iteration Algorithm can be used to constructively establish the existence of and compute n-discount optimal and Blackwell optimal policies. The approach is to find the set of all -1-discount optimal policies and then within this set, to find the set of all 0-discount optimal policies and continue this process until the set of n-discount optimal policies is determined. The basic

algorithm is very similar to the multichain average reward policy iteration algorithm.

For notational convenience, set $r_d^m = r_d$ if $m = 0$, and 0 otherwise.

Policy Iteration—N-discount optimality.

1. Set $m = -1$, $D_{-1} = D$ and $y_{-2}^* = 0$.
2. Set $n = 0$ and select a $d_n \in D_m$.
3. (Policy evaluation) Obtain $y_m^{d_n}$ and $y_{m+1}^{d_n}$ by solving

$$r_{d_n}^m - y_{m-1}^* + (P_{d_n} - I)y_m = 0\,, \tag{8.68}$$

$$r_{d_n}^{m+1} - y_m + (P_{d_n} - I)y_{m+1} = 0\,, \tag{8.69}$$

subject to $y_{m+1}(s) = 0$ for one s in each recurrent class of P_{d_n}.

4. (Policy improvement)

(a) Choose a $d_{n+1} \in D_m$ to satisfy

$$r_{d_{n+1}}^m + P_{d_{n+1}} y_m^{d_n} = \max_{d \in D_m} \{r_d^m + P_d y_m^{d_n}\} \tag{8.70}$$

and set $d_{n+1} = d_n$ if possible. If $d_{n+1} = d_n$, set $E_n^m = \{d \in D_m\colon d$ attains the maximum in (8.70)$\}$ and go to (b), otherwise increment n by 1 and return to step 3.

(b) Choose a $d_{n+1} \in E_n^m$ to satisfy

$$r_{d_{n+1}}^{m+1} + P_{d_{n+1}} y_{m+1}^{d_n} = \max_{d \in D_m} \{r_d^{m+1} + P_d y_{m+1}^{d_n}\} \tag{8.71}$$

and set $d_{n+1} = d_n$ if possible. If $d_{n+1} = d_n$ go to 5, otherwise increment n by 1 and return to step 3.

5. Set $D_{m+1} = \{d' \in D_m\colon d'$ attains the maximum in (8.71)$\}$. If D_{m+1} is a singleton or $m = N$, stop. Otherwise, set $y_m^* = y_m^{d'}$ where $d' \in D_{m+1}$, increment m by 1 and return to 2.

The algorithm terminates with D_{n+1}, the set of N-discount stationary optimal policies. When $N = -1$, that is an average optimal policy is sought, the above algorithm is identical to that in Section 8.8. The specification to determine $y_{m+1}^{d_n}$ uniquely in Step 3 can be implemented by solving the system of three equations given by (8.68), (8.69) and the identical equation to (8.69) at $m + 2$ (Veinott, 1969).

Proof of convergence of the algorithm is similar to that in Section 8.8. For each fixed m, the algorithm finds a sequence of policies that are monotonically increasing in sense of terms of

$$\liminf_{\rho \downarrow 0} \rho^{-m}[v_\lambda^{d_{n+1}} - v_\lambda^{d_n}] \geqslant 0\,.$$

Since there are only *finitely* many stationary policies, this step terminates with D_{m+1}, the set of m-discount optimal stationary policies. When $m = N$, the algorithm terminates with the set of N-discount optimal policies.

Since Blackwell optimality corresponds to ∞-discount optimality, the above suggests that an infinite number of passes throught the above policy iteration algorithm is necessary to obtain a Blackwell optimal stationary policy. Miller and Veinott (1968) showed that this is not the case.

Theorem 8.33. *Suppose d^* is N-discount optimal where N is the number of states in S. Then d^* is Blackwell optimal.*

Veinott (1974) and Lamond (1986) showed that the N in Theorem 8.33 can be replaced by $N - k$, where k is number of recurrent classes in an $(N - k)$-discount optimal policy. The following immediate corollary to the above theorem ties together many of the results in Section 8.

Corollary 8.34. *Suppose A_s for each $s \in S$ and S are finite. Then there exists a stationary Blackwell optimal policy which can be determined by the policy iteration algorithm in a finite number of iterations.*

Veinott (1966), Denardo and Miller (1968), Lippman (1968), Sladky (1974), Hordijk and Sladky (1977), Denardo and Rothblum (1979) and van der Wal (1981) have investigated the relationship between discount optimality, overtaking optimality and average optimality as defined in Section 5.

8.10. Computational results

This section illustrates policy iteration and value iteration in the undiscounted case by using these algorithms to find average optimal policies for an infinite horizon version of the stochastic inventory model of Section 3.2. The chain structure of stationary policies determines whether the single equation policy iteration scheme of Section 8.3 or the multiple equation method of Section 8.8 is appropriate and whether value iteration as defined in Section 8.4. is convergent. This structure is investigated first.

Since self transitions are possible under all actions, all policies are aperiodic. Consider the stationary policy $d = (0, 2, 1, 0)$ corresponding to the following ordering rule. If the inventory at the start of the period is 0 units, then no order is placed and if the inventory is 1 or more units, an order is placed which instantaneously raises the stock level to 3 units. Although such a policy is feasible, it is clearly impractical and non-optimal. This stationary policy is not unichain because it partitions the state space into two recurrent classes: $\{0\}$ and $\{1, 2, 3\}$. All remaining stationary policies are unichain.

Even though the set of stationary policies does not satisfy the unichain assumption, this problem is communicating since each state can be reached from each other state with positive probability in a finite number of iterations

under some policy. Consequently value iteration and modified policy iteration are convergent. Since not all policies are unichain the multichain version of policy iteration is required.

8.10.1. Policy iteration

The algorithm in Section 8.8 is applied with $d_0 = (0, 2, 1, 0)$. Solving the first evaluation equation yields

$$g_{d_0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad h^{\mathrm{B}}_{d_0} = \begin{bmatrix} 0 \\ -3 \\ -1 \\ 5 \end{bmatrix}.$$

Coincidentally, g_{d_0} is constant so that in the improvement stage of the algorithm, equation (8.56) is superfluous. If the data is perturbed slightly, then g_{d_0} will not be constant. This means that at the first pass through the algorithm an improved policy will be obtained through (8.56) so that the first improvement equation is necessary. Results (including the gain and bias) are reported in Table 8.1.

Table 8.1
Policy iteration results

n	g_{d_n} $(h^{\mathrm{B}}_{d_n}(s))$				$d_n(s)$			
	$s=0$	$s=1$	$s=2$	$s=3$	$s=0$	$s=1$	$s=2$	$s=3$
0	0 (0)	0 (−3.0)	0 (−1.0)	0 (5.0)	0	2	1	0
1	0 (0)	0 (6.6667)	0 (12.4444)	0 (17.1852)	0	0	0	0
2	1.6 (−5.08)	1.6 (−3.08)	1.6 (2.12)	1.6 (4.92)	3	2	0	0
3	2.2045 (−4.2665)	2.2045 (−0.5393)	2.2045 (3.2789)	2.2045 (5.7335)	3	0	0	0
4	×	×	×	×	3	0	0	0

Since the average optimal policy is unique, it is n-discount optimal for all n. Thus it is Blackwell optimal and discount optimal for all discount factors sufficiently close to 1. It agrees with that found in the discounted case with $\lambda = 0.9$.

8.10.2. Value iteration

Since all policies are aperiodic and communicating, value iteration and its bounds are convergent. Table 8.2 reports the results of applying this algorithm to the inventory example. The upper and lower bounds in (8.39) are the basis for a stopping rule; Δ^n denotes the difference between these two bounds. A 0.01-optimal solution is sought and v^0 is chosen to be $(0, 0, 0, 0)$.

Table 8.2
Value iteration results

n	$v^n(s)$				$d^n(s)$				Δ^n
	$s=0$	$s=1$	$s=2$	$s=3$	$s=0$	$s=1$	$s=2$	$s=3$	
0	0	0	0	0	0	0	0	0	
1	0	5.0	6.0	5.0	2	0	0	0	6.0000
2	2,0	6.25	10.0	10.50	3	0	0	0	4.2500
3	4.1875	8.0625	12.125	14.1875	3	0	0	0	1.8750
4	6.625	10.1563	14.1094	16.625	3	0	0	0	0.4531
5	8.75	12.5078	16.2617	18.75	3	0	0	0	0.1465
6	10.9453	14.6895	18.5068	20.9453	3	0	0	0	0.0635
7	13.1621	16.8813	20.7078	23.1621	3	0	0	0	0.0164
8	15.3647	19.0919	22.9081	25.3647	3	0	0	0	0.0103
9	17.5682	21.2965	25.1142	27.5685	3	0	0	0	0.0025

Observe that after 9 iterations, the difference between the bounds is 0.0025 so that the decision rule $(3, 0, 0, 0)$ (which is the unique optimal policy identified by policy iteration) is guaranteed to have a gain that is within 0.0025 of optimum. That is, $0 \leq g^* - g_{d_9} \leq 0.0025$. Estimates of g^* and h^* are obtained using the method described in Section 8.4.1. That is

$$g^* \approx v^9 - v^8 = \begin{bmatrix} 2.2035 \\ 2.2046 \\ 2.2060 \\ 2.2035 \end{bmatrix} \quad \text{and} \quad h^* \approx v^9 - 9g^* = \begin{bmatrix} -2.2633 \\ 1.4551 \\ 5.2602 \\ 7.7370 \end{bmatrix}.$$

These calculations imply that $2.2035 \leq g^* \leq 2.2060$ which agrees with the precise value of $g^* = 2.2045$ obtained by solving the policy evaluation equations in the policy iteration algorithm. The values of h^* are considerably different those of policy iteration, but the differences of h^* between any two states are accurately estimated and agree with the exact values. For example, comparing h^* to its value at state 0 yields $h^*(1) - h^*(0) = 4.7184$, $h^*(2) - h^*(0) = 7.5235$ and $h^*(3) - h^*(0) = 10.0000$ which is identical to the exact values. Thus value iteration provides an accurate way of finding relative values. If the precise bias is desired, the policy evaluation equations may be solved.

9. Semi-Markov decision processes and Markov renewal programming

In the MDP's considered in the previous sections, the decision maker chooses actions at a fixed discrete set of time points. Semi-Markov decision processes (SMDP's) generalize MDP's by allowing the decision maker to choose actions at *any* epoch necessitating observing the system in *continuous* time. The effect of choosing an action is to determine the probability distribution of both the subsequent state and the remaining time in the current state.

For a specified policy the system evolves by remaining in a state for a random amount of time and then jumping to a different state. These models are called semi-Markov because for fixed Markov policies the system states evolve according to a semi-Markov process.

Analysis of SMDP's depends on the set of admissible policies or controls. When actions are allowed at any time, continuous time control theory methods are appropriate. If actions can be chosen *only* immediately following transitions and the time horizon is infinite, or a fixed number of transitions, discrete time MDP methods can be adapted. When actions are allowed only after transitions, the models are often referred to as Markov renewal programs (MRP's), however not all authors distinguish MRP's and SMDP's in this way. When the time horizon is finite and fixed, MRP's also require control theory methods for analysis. This section presents results for infinite horizon Markov renewal programs.

MRP's are mostly widely used to model equipment replacement, queueing control and inventory control problems. Notable special cases are continuous time MDP's (CTMDP's) in which the transition times are exponentially distributed and MDP's in which all transition times are constant and equal.

Semi-Markov decision processes were introduced by Jewell (1963), other contributors to the theory include Howard (1963), de Cani (1964), Schweitzer (1965, 1971), Fox (1966) and Denardo (1971). They are also treated in the books of Kallenberg (1983) and Heyman and Sobel (1984). Numerous papers, most notably in the infinite horizon average reward case, have been based on applying a transformation (Schweitzer, 1971) to convert SMDP's to MDP's. A good reference on semi-Markov processes is Çinlar (1975).

9.1. Problem formulation

Let S be either a finite or countable set of states. Whenever a transition occurs into state s, the decision maker chooses an action from the set A_s. As a consequence of choosing action a in state s, the probability that the system is in state j at the next transition is $p(j|s, a)$, and the probability that a transition occurs within t units of time is $F(t|s, a)$. It is assumed that there exists a constant γ, $0 < \gamma < 1$ such that

$$F(0|s, a) \leqslant \gamma \quad \text{for all } a \in A_s \text{ and } s \in S\,. \tag{9.1}$$

The transition structure can be expressed in terms of the joint probability $q(t, j|s, a) = p(j|s, a)F(t|s, a)$. Alternatively the problem can be defined in terms of q in which case p and F can be derived.

The economic consequence of choosing action a in state s is that the decision maker receives a lump sum reward $k(s, a)$, and a continuous reward at rate $c(s, a)$ per unit time as long as the system is in state s.

A Markov decision rule d is a function such that $d(s) \in A_s$ for each $s \in S$. For a specified d let $p_d(j|s) = p(j|s, d(s))$, $F_d(t|s) = F(t|s, d(s))$, $q_d(t|s, j) =$

$q(t|s, d(s), j)$, $k_d(s) = k(s, d(s))$ and $c_d(s) = c(s, d(s))$. To avoid technicalities, it will be assumed that the decision making period begins at the time of the first transition and no rewards are received until that time. Let $\pi = (d_1, d_2, \ldots)$ be an arbitrary policy which corresponds to using d_n at the time of the nth transition. Corresponding to this policy is a semi-Markov process $\{Y_t^\pi; t \geq 0\}$ which gives the state of the system at time t and a process $\{U_t^\pi; t \geq 0\}$ which gives the action chosen at time t. This process can be further described in terms of a sequence of jointly distributed random variables $\{(X_n^\pi, \tau_n^\pi); n = 1, 2, \ldots\}$ where X_n^π is the state of the system immediately following the nth transition and τ_n^π is length of time the system is in state X_n^π. It is convenient to define σ_n^π as the total time until the nth transition starting at the time of the first transition, that is

$$\sigma_n^\pi = \sum_{j=1}^{n-1} \tau_n^\pi$$

and $\sigma_1^\pi = 0$.

9.2. *The discounted case*

Let $\alpha > 0$ denote the continuous time interest rate. That is, the present value of one dollar received at time t is $e^{-\alpha t}$. For policy π, the expected infinite horizon discounted reward given that the first transition is into state s is denoted by $v_\alpha^\pi(s)$ and given by

$$v_\alpha^\pi(s) = E_s^\pi\left[\int_0^\infty e^{-\alpha t} c(Y_t^\pi, U_t^\pi)\, dt + \sum_{n=1}^{\infty} e^{-\alpha\sigma_n^\pi} k(Y_{\sigma_n^\pi}, U_{\sigma_n^\pi})\right]. \tag{9.2}$$

The first term in (9.2) corresponds to the continuous portion of the reward and the second term to the fixed reward received only at decision epochs.

The objective in this problem is to characterize

$$v_\alpha^*(s) = \sup_{\pi \in \Pi} v_\alpha^\pi(s)$$

for all $s \in S$ and to find a policy π^* with the property that

$$v_\alpha^{\pi^*}(s) = v_\alpha^*(s)$$

for all $s \in S$.

This problem is transformed to a discrete time problem by allowing the discount factor to be state and action dependent and analyzing the problem in terms of its embedded chain. Define $r_d(s)$ to be the expected total discounted reward until the next transition if the system just entered state s and decision rule d is selected. It is given by

$$r_d(s) = k_d(s) + c_d(s)E_s^d\left[\int_0^{\tau^d} e^{-\alpha t}\,dt\right]$$

$$= k_d(s) + c_d(s)\{\alpha^{-1}(1 - E_s^d[e^{-\alpha\tau^d}])\} \tag{9.3}$$

where τ^d is the time until the first transition given that the system just entered state s and decision rule d is used. Define the expected discounted holding time in state s if action a is selected by

$$\lambda(s, a) = \int_0^{\infty} e^{-\alpha t}\,dF(t|s, a)\,. \tag{9.4}$$

For $d \in D$, define $\lambda_d(s) = \lambda(s, d(s))$. Note that $\lambda_d(s)$ is the Laplace transform of τ^d. Thus

$$r_d(s) = k_d(s) + c_d(s)\{\alpha^{-1}[1 - \lambda_d(s)]\}\,.$$

Using this quantity, $v_\alpha^\pi(s)$ can be re-expressed as

$$v_\alpha^\pi(s) = E_s^\pi\left[\sum_{n=1}^{\infty} e^{-\alpha\sigma_n^\pi} r_{d_n}(X_n^\pi)\right]$$

$$= r_{d_1}(s) + E_s^\pi[e^{-\alpha\tau^{d_1}} v_\alpha^{\pi'}(X_2^\pi)]$$

$$= r_{d_1}(s) + \sum_{j\in S} \lambda_{d_1}(s)p_{d_1}(j|s)v_\alpha^{\pi'}(j)$$

where $\pi' = (d_2, d_3, \ldots)$. For a stationary policy d, the infinite horizon expected total discounted reward can be obtained by solving the equation

$$v_\alpha^d(s) = r_d(s) + \sum_{j\in S} \lambda_d(s)p_d(j|s)v_\alpha^d(j)\,.$$

This can be expressed in matrix terms as

$$v = r_d + M_d v \tag{9.5}$$

where M_d is the matrix with entries $\lambda_d(s)p_d(j|s)$. This differs from the discrete time evaluation equation (6.8) by the state and action dependent discount rate (9.4). From a computational point of view this causes the matrix M_d to have unequal row sums bounded by $\lambda^* = \sup_{a,s}\lambda(s, a)$. The efficiency of numerical methods for solution of (9.5) has been investigated by Porteus (1980a, 1983).

The optimality equation for Markov renewal programs is given by

$$v = \max_{d\in D}\{r_d + M_d v\} \equiv Tv\,. \tag{9.6}$$

If (9.1) holds, $\lambda^* < 1$ and if in addition $\|r_d\| \leq M < \infty$ for all $d \in D$, T defined in (9.6) is a contraction operator on the space of bounded real valued functions on S, so consequently all results of Section 6 apply. This means that for MRP's:

(1) The optimality equation has a unique solution v_α^*.

(2) There exists a stationary policy which is optimal.

(3) The problem can be solved by value iteration, policy iteration, modified policy iteration, linear programming and their variants.

(4) Bounds and action elimination methods are valid.

(5) Extensions to unbounded rewards are possible.

9.3. The average reward case

Let T be the time the system has been under observation since the first decision epoch. For a fixed $\pi \in \Pi$, define the expected total reward up to time T by

$$v_T^\pi(s) = E_s^\pi\left[\int_0^T c(Y_t^\pi, U_t^\pi)\,\mathrm{d}t + \sum_{n=1}^{\nu_T^\pi} k(Y_{\pi_{\sigma_n}}, U_{\pi_{\sigma_n}})\right], \tag{9.7}$$

where ν_T^π is the random variable representing the number of decisions made up to time T using policy π. For each $\pi \in \Pi$, define the average expected reward or gain by

$$g^\pi(s) = \liminf_{T\to\infty} \frac{1}{T}\, v_T^\pi(s)\,, \quad s \in S\,.$$

The objective in the average reward case is to characterize the optimal average expected reward

$$g^*(s) = \sup_{\pi\in\Pi} g^\pi(s)\,, \quad s \in S\,,$$

and determine a policy π^* with the property that

$$g^{\pi^*}(s) = g^*(s)\,, \quad s \in S\,.$$

Let $r_d(s)$ be the expected total reward until the next transition when the system is in state s and decision rule d is used. It is given by

$$r_d(s) = k_d(s) + c_d(s)E_s^d[\tau^d] = k_d(s) + c_d(s)H_d(s) \tag{9.8}$$

where $H_d(s)$ is defined as follows. For each $a \in A_s$ and $s \in S$, the expected time in state s until the next transition, $H(s, a)$ is given by

$$H(s, a) \equiv \int_0^\infty t\,\mathrm{d}F(t\,|\,s, a)\,. \tag{9.9}$$

For $d \in D$, define $H_d(s) \equiv H(s, d(s))$. Under (9.1), $\eta \equiv \inf_{s,a} H(s, a) > 0$.

The gain of a fixed, stationary policy d is uniquely determined by the equations

$$(P_d - I)g = 0 \quad \text{and} \quad r_d - gH_d + (P_d - I)h = 0\,. \tag{9.10}$$

The second equation in (9.10) uniquely determines h up to an element in the null space of $P_d - I$. The derivation of (9.10) is based on the partial Laurent series expansion of v_α^d (Denardo, 1971).

The corresponding optimality equations are

$$\max_{d \in D} \{(P_d - I)g\} = 0 \quad \text{and} \quad \max_{d \in E} \{r_d - gH_d + (P_d - I)h\} = 0 \tag{9.11}$$

where $E = \{d \in D\colon P_d g = g\}$.

Both the evaluation equations and the optimality equations differ from the MDP case by the inclusion of the term H_d in the second equation. These equations can be solved using the policy iteration methods of Section 8. The only modification is that $H(s, a)$ must be evaluated for each state-action pair. To obtain other theoretical results and establish the convergence of value iteration, the problem can be converted into an 'equivalent' MDP by applying the following transformation (Schweitzer, 1971).

Define a transformed MDP indicated by "$\sim$" as follows:

$$\tilde{r}(s, a) = r(s, a)/H(s, a)$$

and

$$\tilde{p}(j\,|\,s, a) = \eta'[p(j\,|\,s, a) - \delta(j\,|\,s)]/H(s, a) + \delta(j\,|\,s)$$

where $\delta(j\,|\,s) = 1$ if $j = s$ and 0 otherwise and

$$0 < \eta' < H(s, a)/(1 - p(s\,|\,s, a))$$

for all $a \in A_s$ and $s \in S$ for which $p(s\,|\,s, a) < 1$.

The choice of η' ensures that $\tilde{p}(s\,|\,s, a) > 0$ so that all stationary policies have aperiodic chains. The sets of optimal policies for the original and transformed problems are identical and $\tilde{g}^* = g^*$ and $\tilde{h} = \eta' h$.

Because of this, the following results hold for the Markov renewal programming problem with average reward criteria:

(1) Whenever (8.23) holds,

(a) the first optimality equation is redundant and there exists a solution to the second optimality equation, and

(b) there exists an optimal stationary policy.

(2) When S is finite, value iteration converges in the sense of Section 8.4, it can be used to determine optimal policies, and bounds are available.

(3) Linear programming can be used to determine optimal policies in the finite state and action case.

Bibliography

Bather, J. (1975). Optimal decision procedures for finite Markov chains. *Adv. Appl. Probab.* **5**, 328–339, 521–540, 541–553.

Bellman, R.E. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ.

Bertsekas, D.P. (1987). *Dynamic Programming, Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, NJ.

Blackwell, D. (1961). On the functional equation of dynamic programming. *J. Math. Anal. Appl.* **2**, 273–276.

Blackwell, D. (1962). Discrete dynamic programming. *Ann. Math. Statist.* **35**, 719–726.

Blackwell, D. (1965). Discounted dynamic programming. *Ann. Math. Statist.* **36**, 226–235.

Blackwell, D. (1967). Positive dynamic programming. *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability* **1**, 415–418.

Brown, B.W. (1965). On the iterative method of dynamic programming on a finite space discrete Markov process. *Ann. Math. Statist.* **36**, 1279–1286.

Çinclar, E. (1975). *Introduction to Stochastic Processes*. Prentice-Hall, Englewood Cliffs, NJ.

De Cani, J.S. (1964). A dynamic programming algorithm for embedded Markov chains when the planning horizon is at infinity. *Management Sci.* **10**, 716–733.

Dekker, R. (1985). Denumerable Markov decision chains: Optimal policies for small interest rates. Unpublished Ph.D. Dissertation, University of Leiden.

Dembo, R. and M. Haviv (1984). Truncated policy iteration methods. *Oper. Res. Lett.* **3**, 243–246.

Demko, S. and T.P. Hill (1981). Decision processes with total cost criteria. *Ann. Probab.* **9**, 293–301.

Denardo, E.V. (1967). Contraction mappings in the theory underlying dynamic programming. *SIAM Rev.* **9**, 169–177.

Denardo, E.V. and B. Fox (1968). Multichain Markov renewal programming. *SIAM J. Appl. Math.* **16**, 468–487.

Denardo, E.V. and B.L. Miller (1968). An optimality condition for discrete dynamic programming with no discounting. *Ann. Math. Statist.* **39**, 1220–1227.

Denardo, E.V. (1970). Computing a bias-optimal policy in a discrete-time Markov decision problem. *Oper. Res.* **18**, 279–289.

Denardo, E.V. (1971). Markov renewal programs with small interest rates. *Ann. Math. Statist.* **42**, 477–496.

Denardo, E.V. (1973). A Markov decision problem. In: T.C. Hu and S.M. Robinson (Eds.), *Mathematical Programming*. Academic Press, New York.

Denardo, E.V. and U.G. Rothblum (1979). Overtaking optimality for Markov decision chains. *Math. Oper. Res.* **4**, 144–152.

Denardo, E.V. (1982). *Dynamic Programming, Models and Applications*. Prentice-Hall, Englewood Cliffs, NJ.

D'Epenoux, F. (1963). Sur un problème de production et de stockage dans l'aleatoire. *Rev. Francaise Automat. Informat. Rech. Oper.* **14** (English Transl.: *Management Sci.* **10**, 98–108).

Deppe, H. (1984). On the existence of average optimal policies in semi-regenerative decision models. *Math. Oper. Res.* **9**, 558–575.

Derman, C. (1966). Denumerable state Markovian decision processes—Average cost criterion. *Ann. Math. Statist.* **37**, 1545–1554.

Derman, C. and R. Strauch (1966). A note on memoryless rules for controlling sequential decision processes. *Ann. Math. Statist.* **37**, 276–278.

Derman, C. (1970). *Finite state Markovian decision processes*. Academic Press, New York.

Dirickx, Y.M.J. and M.R. Rao (1979). Linear programming methods for computing gain-optimal policies in Markov decision models. *Cah. Centre d'Etudes Rech. Oper.* **21**, 133–142.

Dubins, L.E. and L.J. Savage (1965). *How to Gamble if You Must: Inequalities for Stochastic Processes*. McGraw-Hill, New York.

Eagle, J.E. (1975). A Utility Criterion for the Markov Decision Process. Unpublished Ph.D. Dissertation, Dept. of Engineering–Economic Systems, Stanford University.

Federgruen, A., A. Hordijk and H.C. Tijms (1978). Recurrence conditions in denumerable state Markov decision processes. In: M.L. Puterman (Ed.), *Dynamic Programming and Its Applications*. Academic Press, New York, 3–22.

Federgruen, A. and P.J. Schweitzer (1978). Discounted and undiscounted value-iteration in Markov decision problems: A survey. In: M.L. Puterman (Ed.), *Dynamic Programming and Its Applications*. Academic Press, New York, 23–52.

Federgruen, A. and H.C. Tijms (1978). The optimality equation in average cost denumerable state semi-Markov decision problems, recurrency conditions and algorithms. *J. Appl. Probab.* **15**, 356–373.

Federgruen, A., A. Hordijk and H.C. Tijms (1979). Denumerable state semi-Markov decision processes with unbounded costs, average cost criteria. *Stochastic Process. Appl.* **9**, 223–235.

Federgruen, A. and P.J. Schweitzer (1980). A survey of asymptotic value-iteration for undiscounted Markovian decision processes. In: R. Hartley, L.C. Thomas and D.J. White (Eds.), *Recent Developments in Markov Decision Processes*. Academic Press, New York, 73–109.

Federgruen, A., P.J. Schweitzer and H.C. Tijms (1983). Denumerable undiscounted semi-Markov decision processes with unbounded rewards. *Math. Oper. Res.* **8**, 298–313.

Federgruen, A. and J.P. Schweitzer (1984a). Successive approximation methods for solving nested functional equations in Markov decision problems. *Math. Oper. Res.* **9**, 319–344.

Federgruen, A. and J.P. Schweitzer (1984b). A fixed point approach to undiscounted Markov renewal programs. *SIAM J. Algebraic Discrete Methods* **5**, 539–550.

Fisher, L. and S.M. Ross (1968). An example in denumerable decision processes. *Ann. Math. Statist.* **39**, 674–676.

Fleming, W.H. and R. Rishel (1975). *Deterministic and Stochastic Optimal Control*. Springer-Verlag, New York.

Flynn, J. (1976). Conditions for the equivalence of optimality criteria in dynamic programming. *Ann. Statist.* **4**, 936–953.

Fox, B.L. (1966). Markov renewal programming by linear fractional programming. *SIAM J. Appl. Math.* **14**, 1418–1432.

Grinold, R. (1973). Elimination of suboptimal actions in Markov decision problems. *Oper. Res.* **21**, 848–851.

Harrison, J.M. (1972). Discrete dynamic programming with unbounded rewards. *Ann. Math. Statist.* **43**, 636–644.

Hartley, R., L.C. Thomas and D.J. White (Eds.) (1980). *Recent Developments in Markov Decision Processes*. Academic Press, New York.

Hartley, R. (1980). A simple proof of Whittle's bridging condition in dynamic programming. *J. Appl. Probab.* **17**, 1114–1116.

Hastings, N.A.J. (1968). Some note son dynamic programming and replacement. *Oper. Res. Quart.* **19**, 453–464.

Hastings, N.A.J. (1969). Optimization of discounted Markov decision problems. *Oper. Res. Quart.* **20**, 499–500.

Hastings, N.A.J. (1976). A test for suboptimal actions in undiscounted Markov decision chains. *Management Sci.* **23**, 87–91.

Hastings, N.A.J. and J.A.E.E. van Nunen (1977). The action elimination algorithm for Markov decision processes. In: H.C. Tijms and J. Wessels (eds.), *Markov Decision Theory*, Mathematical Centre Tract No. 93. Mathematical Centre, Amsterdam, 161–170.

Haviv, M. and M.L. Puterman (1990). Improved policy iteration methods for communicating Markov decision processes. *Annals of Operations Research*, Special Issue on Markov Decision Processes, to appear.

Hernández-Lerma, O. (1989). *Adaptive Markov Control Processes*. Springer-Verlag, New York.

Heyman, D.P. and M.J. Sobel (1984). *Stochastic Models in Operations Research*, Vol. II. McGraw-Hill, New York.

Hille, E. and R.S. Phillips (1957). *Functional Analysis and Semi-Groups*, American Mathematical Society Colloquim Publications, Vol. 31. AMS, Providence, RI.

Hinderer, K. (1970). *Foundations of Non-Stationary Dynamic Programming with Discrete Time Parameter*. Springer-Verlag, New York.

Hopp, W.J., J.C. Bean and R.L. Smith (1988). A new optimality criterion for non-homogeneous Markov decision processes. *Oper. Res.* **35**, 875,–883.

Hordijk, A. (1974). *Dynamic Programming and Markov Potential Theory*. Mathematical Centre Tract No. 51. Mathematical Centre, Amsterdam.

Hordijk, A. and K. Sladky (1977). Sensitive optimality criteria in countable state dynamic programming. *Math. Oper. Res.* **2**, 1–14.

Hordijk, A. and L.C.M. Kallenberg (1979). Linear programming and Markov decision chains. *Management Sci.* **25**, 352–362.

Hordijk, A. and L.C.M. Kallenberg (1980). On solving Markov decision problems by linear programming. In: R. Hartley, L.C. Thomas and D.J. White (Eds.), *Recent Developments in Markov Decision Processes*. Academic Press, New York, 127–143.

Hordijk, A. and M.L. Puterman (1987). On the convergence of policy iteration in undiscounted finite state Markov decision processes; the unichain case. *Math. Oper. Res.* **12**, 163–176.

Howard, R. (1960). *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA.

Howard, R.A. (1963). Semi-Markovian decision processes. *Proc. Internat. Statist. Inst.*, Ottawa, Canada.

Howard, R.A. and J.E. Matheson (1972). Risk sensitive Markov decision processes. *Management Sci.* **8**, 356–369.

Hubner, G. (1977). Improved procedures for eliminating suboptimal actions in Markov programming by the use of contraction properties. *Transactions of the Seventh Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*, 257–263.

Hubner, G. (1988). A unified approach to adaptive control of average reward decision processes. *OR Spektrum* **10**, 161–166.

Jacquette, S.C. (1973). Markov decision processes with a new optimality condition: Discrete time. *Ann. Statist.* **3**, 496–505.

Jewell, W.S. (1963). Markov-renewal programming I: Formulation, finite return models; II: Infinite return models, example. *Oper. Res.* **11**, 938–971.

Kallenberg, L.C.M. (1983). *Linear Programming and Finite Markov Control Problems*, Mathematical Centre Tract No. 148. Mathematical Centre, Amsterdam.

Kantorovich, L.V. (1952). *Functional Analysis and Applied Mathematics*, Translated by C.D. Benster, NBS Report 1509, National Bureau of Standards, Los Angeles, CA.

Kemeny, J.G. and J.L. Snell (1960). *Finite Markov Chains*. Van Nostrand-Reinhold, New York.

Kreps, D.M. and E. Porteus (1977). On the optimality of structured policies in countable stage decision processes, II: Positive and negative problems. *SIAM J. Appl. Math.* **32**, 457–466.

Lamond, B.L. (1984). MDPLAB, an interactive computer program for Markov dynamic programming. Working Paper 1068, Faculty of Commerce, University of British Columbia.

Lamond, B.L. (1986). Matrix methods in queueing and dynamic programming. Unpublished Ph.D. Dissertation, Faculty of Commerce, University of British Columbia.

Lamond, B.L. and M.L. Puterman (1989). Generalized inverses in discrete time Markov decision processes. *SIAM J. Mat. Anal. Appl.* **10**, 118–134.

Lanery, E. (1967). Etude asymptotique des systèmes Markovien à commande. *Rev. Française Inform. Rech. Oper.* **1**, 3–56.

Lippman, S.A. (1968). Criterion equivalence in discrete dynamic programming. *Oper. Res.* **17**, 920–923.

Lippman, S.A. (1975). On Dynamic Programming with Unbounded Rewards. *Management. Sci.* **21**, 1225–1233.

Liusternik, L. and V. Sobolev (1961). *Elements of Functional Analysis*. Ungar, New York.

MacQueen, J. (1966). A modified dynamic programming method for Markov decision problems. *J. Math. Anal. Appl.* **14**, 38–43.

Mandl, P. (1967). An iterative method for maximizing the characteristic root of positive matrices. *Rev. Roumaine Math. Pures Appl.* **12**, 1312–1317.

Mandl, P. (1974). Estimation and control in Markov chains. *Adv. in Appl. Probab.* **6**, 40–60.

Mann, E. (1983). Optimality equations and bias optimality in bounded Markov decision processes. Preprint No. 574, University of Bonn.

Manne, A. (1960). Linear programming and sequential decisions. *Management Sci.* **6**, 259–267.

Miller, B.L. and A.F. Veinott, Jr. (1969). Discrete dynamic programming with a small interest rate. *Ann. Math. Statist.* **40**, 366–370.

Mine, H. and S. Osaki (1968). Some remarks on a Markovian decision process with an absorbing state. *J. Math. Anal. Appl.* **23**, 327–333.

Monahan, G.E. (1982). A survey of partially observabel Markov decision processes: Theory, models, and algorithms. *Management Sci.* **28**, 1–16.

Morton, T.E. (1971). On the asymptotic convergence rate of cost differences for Markovian decision processes. *Oper. Res.* **19**, 244–248.

Morton, T.E. and W.E. Wecker (1977). Discounting ergodicity and convergence for Markov decision processes. *Management Sci.* **23**, 890–900.

Morton, T. (1978). The non-stationary infinite horizon inventory problem. *Management Sci.* **24**, 1474–1482.

Odoni, A.R. (1969). On finding the maximal gain for Markov decision processes. *Oper. Res.* **17**, 857–860.

Ohno, K. (1985). Modified policy iteration algorithm with nonoptimality tests for undiscounted Markov decision processes. Working Paper, Dept. of Information System and Management Science, Konan University, Japan.

Ohno, K. and K. Ichiki (1987). Computing optimnal policies for tandem queueing systems. *Oper. Res.* **35**, 121–126.

Ornstein, D. (1969). On the existence of stationary optimal strategies. *Proc. Amer. Math. Soc.* **20**, 563–569.

Ortega, J.M. and W.C. Rheinboldt (1970). *Iterative Solutions of Nonlinear equations in Several Variables*. Academic Press, New York.

Platzman, L. (1977). Improved conditions for convergence in undiscounted Markov renewal programming. *Oper. Res.* **25**, 529–533.

Pliska, S.R. (1976). Optimization of multitype branching processes. *Management Sci.* **23**, 117–125.

Pliska, S.R. (1978). On the transient case for Markov decision processes with general state spaces. In: M.L. Puterman (Ed.), *Dynamic Programming and Its Application*. Academic Press, New York, 335–350.

Porteus, E. (1971). Some bounds for discounted sequential decision processes. *Management Sci.* **18**, 7–11.

Porteus, E. and J. Totten (1978). Accelerated computation of the expected discounted return in a Markov chain. *Oper. Res.* **26**, 350–358.

Porteus, E. (1980a). Improved iterative computation of the expected discounted return in Markov and semi-Markov chains. *Z. Oper. Res.* **24**, 155–170.

Porteus, E. (1980b). Overview of iterative methods for discounted finite Markov and semi-Markov decision chains. In: R. Hartley, L.C. Thomas and D.J. White (Eds.), *Recent Developments in Markov Decision Processes*. Academic Press, New York, 1–20.

Porteus, E. (1981). Computing the discounted return in Markov and semi-Markov chains. *Naval Res. Logist. Quart.* **28**, 567–578.

Porteus, E. (1983). Survey of numerical methods for discounted finite Markov and semi-Markov chains. Presented at *Twelfth Conference on Stochastic Processes and Their Applications*, Ithaca, NY.

Puterman, M.L. (Ed.) (1978). *Dynamic Programming and Its Applications*. Academic Press, New York.

Puterman, M.L. and S.L. Brumelle (1978). The analytic theory of policy iteration. In: M.L. Puterman (ed.), *Dynamic Programming and Its Application*. Academic Press, New York.

Puterman, M.L. and M.C. Shin (1978). Modified policy iteration algorithms for discounted Markov decision problems. *Management Sci.* **24**, 1127–1137.

Puterman, M.L. and S.L. Brumelle (1979). On the convergence and policy iteration in stationary dynamic programming. *Math. Oper. Res.* **4**, 60–69.

Puterman, M.L. and M.C. Shin (1982). Action elimination procedures for modified policy iteration algorithms. *Oper. Res.* **30**, 301–318.

Puterman, M.L. (1991). *Markov Decision Processes*. Wiley, New York.

Ross, S. (1968a). Non-Discounted denumerable Markovian decision models. *Ann. Math. Statist.* **39**, 412–423.

Ross, S.M. (1968b). Arbitrary state Markovian decision processes. *Ann. Math. Statist.* **39**, 2118–2122.

Ross, S.M. (1983). *Introduction to Stochastic Dynamic Programming*. Academic Press, New York.

Rothblum, U.G. and A.F. Veinott, Jr. (1975). Cumulative average optimality for normalized Markov decision chains. Working Paper, Dept. of Operations Research, Stanford University.

Rothblum, U.G. (1979). Iterated successive approximation for sequential decision processes. In: J.W.B. van Overhagen and H.C. Tijms, (Eds.), *Stochastic Control and Optimization*. Vrije Universiteit, Amsterdam, 30–32.

Rothblum, U.G. (1984). Multiplicative Markov decision chains. *Math. Oper. Res.* **9**, 6–24.

Schal, M. (1975). Conditions for optimality in dynamic programming and for the limit of n-stage optimal policies to be optimal. *Z. Wahrsch. Verw. Gebiete* **32**, 179–196.

Schweitzer, P.J. (1965). Perturbation theory and Markov decision chains. Unpublished Ph.D. Dissertation, Massachusetts Institute of Technology.

Schweitzer, P.J. (1971). Iterative solution of the functional equations of undiscounted Markov renewal programming. *J. Math. Anal. Appl.* **34**, 495–501.

Schweitzer, P.J. and A. Federgruen (1977). The asymptotic behavior of undiscounted value iteration in Markov decision problems. *Math. Oper. Res.* **2**, 360–381.

Schweitzer, P.J. and A. Federgruen (1978). The functional equations of undiscounted Markov renewal programming. *Math. Oper. Res.* **3**, 308–321.

Schweitzer, P.J. and A. Federgruen (1979). Geometric convergence of value iteration in multichain Markov decision problems. *Adv. in Appl. Probab.* **11**, 188–217.

Schweitzer, P.J. (1985). On undiscounted Markovian decision processes with compact action spaces. *Rev. RAIRO Rech. Oper.* **19**, 71–86.

Seneta, E. (1981). *Non-negative Matrices and Markov Chains*. Springer-Verlag, New York.

Shapiro, J. (1968). Turnpike planning horizons for a Markovian decision model. *Management Sci.* **14**, 292–300.

Shapley, L.S. (1953). Stochastic games. *Proc. Nat. Acad. Sci. U.S.A.* **39**, 1095–1100.

Sheu, S.S. and K.-J. Farn (1980). A sufficient condition for the existence of a stationary 1-optimal plan in compact action Markovian decision processes. In: R. Hartley, L.C. Thomas and D.J. White (Eds.), *Recent Developments in Markov Decision Processes*. Academic Press, New York, 111–126.

Sladky, K. (1974). On the set of optimal controls for Markov chains with rewards. *Kybernetika* **10**, 350–367.

Smallwood, R. and E. Sondik (1973). The optimal control of partially observable Markov processes over a finite horizon. *Oper. Res.* **21**, 1071–1088.

Sobel, M.J. (1982). The variance of discounted Markov decision processes. *J. Appl. Probab.* **19**, 794–802.

Sondik, E.J. (1971). The optimal control of partially observable Markov processes. Ph.D. Dissertation, Department of Engineering–Economic Systems, Stanford University.

Sondik, E. (1978). The optimal control of Partially observable Markov processes over the infinite horizon: Discounted costs. *Oper. Res.* **26**, 282–304.

Strauch, R. (1966). Negative dynamic programming. *Ann. Math. Statist.* **37**, 871–890.

Taylor, H.M. (1965). Markovian sequential replacement processes. *Ann. Math. Statist.* **36**, 1677–1694.

Tijms, H.C. and J. Wessels (eds.) (1977). *Markov Decision Theory*. Tract 93, Mathematical Centre, Amsterdam.

van Dawen, R. (1986a). Finite state dynamic programming with the total reward criterion. *Z. Oper. Res.* **30**, A1–A14.

van Dawen, R. (1986b). Pointwise and uniformly good stationary strategies in dynamic programming models. *Math. Oper. Res.* **11**, 521–535.

van der Wal, J. and J.A.E.E. van Nunen (1977). A note on the convergence of the value oriented successive approximations method. COSO Note R 77-05, Department of Mathematics, Eindhoven University of Technology.

van der Wal, J. (1984). *Stochastic Dynamic Programming*. Tract 139, Mathematical Centre, Amsterdam.

van der Wal, J. (1984). On stationary strategies in countable state total reward Markov decision processes. *Math. Oper. Res.* **9**, 290–300.

van Hee, K. (1978). Markov strategies in dynamic programming. *Math. Oper. Res.* **3**, 37–41.

van Nunen, J.A.E.E. (1976a). A set of successive approximation methods for discounted Markovian decision problems. *Z. Oper. Res.* **20**, 203–208.

van Nunen, J.A.E.E. (1976b). *Contracting Markov Decision Processes*. Tract 71, Mathematical Centre, Amsterdam.

van Nunen, J.A.E.E. and J. Wessels (1978). A note on dynamic programming with unbounded rewards. *Management Sci.* **24**, 576–560.

Veinott, Jr., A.F. (1966). On finding optimal in discrete dynamic programming with no discounting. *Ann. Math. Statist.* **37**, 1284–1294.

Veinott, Jr., A.F. (1968). Extreme points of Leontief substitution systems. *Linear Algebra Appl.* **1**, 181–194.

Veinott, Jr., A.F. (1969). On discrete dynamic programming with sensitive discount optimality criteria. *Ann. Math. Statist.* **40**, 1635–1660.

Veinott, Jr., A.F. (1974). Markov decision chains. In: G.B. Dantzig and B.C. Eaves (Eds.), *Studies in Optimization*. American Mathematical Association, Providence, RI.

White, D.J. (1963). Dynamic programming, Markov chains, and the method of successive approximations. *J. Math. Anal. Appl.* **6**, 373–376.

White, D.J. (1978). Elimination of non-optimal actions in Markov decision processes. In: M.L. Puterman (Ed.), *Dynamic Programming and Its Applications*. Academic Press, New York, 131–160.

White, D.J. (1985a). Monotone value iteration for discounted finite Markov decision processes. *J. Math. Anal. Appl.* **109**, 311–324.

White, D.J. (1985b). Real applications of Markov decision processes. *Interfaces* **15**, 73–83.

White, D.J. (1988). Mean, variance, and probabilistic criteria in finite Markov decision processes: A review. *J. Optim. Theory Appl.* **56**, 1–29.

Whittle, P. (1979). A simple condition for regularity in negative programming. *J. Appl. Probab.* **16**, 305–318.

Whittle, P. (1980a). Stability and characterisation condition in negative programming. *J. Appl. Probab.* **17**, 635–645.

Whittle, P. (1980b). Negative programming with unbounded costs: A simple condition for regularity. In: R. Hartley, L.C. Thomas, D.J. White (Eds.), *Recent Developments in Markov Decision Processes*. Academic Press, New York, 23–34.

Whittle, P. (1983). *Optimization Over Time, Dynamic Programming and Stochastic Control*, Vol. II. J. Wiley and Sons, New York.

Wijngaard, J. (1977). Sensitive optimality in stationary Markov decision chains on a general state space. In: H.C. Tijms and J. Wessels (Eds.), *Markov Decision Theory*, Mathematical Centre Tracts No. 93. Mathematical Centre, Amsterdam, 85–94.

Yosida, K. (1968). *Functional Analysis*. Springer-Verlag, New York.