MARKOV DECISION PROCESSES

Ulrich Rieder
University of Ulm, Germany

Keywords: Markov decision problem, stochastic dynamic program, total reward criteria, average reward, optimal policy, optimality equation, backward induction algorithm, policy iteration, policy improvement, linear programming.

Contents

Summary

We consider Markov decision processes in discrete and continuous time with a finite or countable state space and an arbitrary action space. They are mathematical models for dynamic systems which can be controlled by sequential decisions and which contain stochastic elements. The main areas of applications are operations research, computer science, engineering and statistics. Typical problems arise, e.g. in queueing, inventory and investment models. At first, Markov decision processes are introduced and some examples are presented. In Section 3, we consider finite horizon decision problems and explain the backward induction algorithm. Section 4 is concerned with infinite horizon Markov decision problems. For each optimality criterion, results are formulated in as much generality as possible. Computational methods like policy iteration, policy improvement and linear programming are discussed. Section 5 introduces continuous time Markov decision processes. These problems can be reduced to discrete-time problems and thus can be solved in the same way. The last section identifies some

extensions and further stochastic decision processes.

1. Introduction

Markov decision processes are dynamical systems that can be controlled by sequential decisions. The successive state transitions are uncertain, but the probability distribution of the next state, as well as the immediate reward, depend only on the current state and the current decision. These models are also known as *stochastic dynamic programs* or *stochastic control problems*. They have been applied in various different subject areas, such as queueing systems, inventory models, and investment models (see *Queuing System*s, *Inventory Models*, *Investment Models*).

Markov decision processes serve as models for controlled dynamic systems of the following type. At decision time points, the controller of the system is allowed to choose an action according to rules given by the system. The choice is based on the knowledge of the current state and has two consequences: it generates an immediate reward for the decision-maker and determines a probability distribution for the next state of the system. The aim is to select a sequence of actions, also called a policy, in such a way as to maximize an overall performance criterion. Since the choice of actions influences the transition probabilities of the system, the controller has to be aware of all consequences of his chosen decisions. These stochastic control problems can roughly be classified in

- Continuous-time or discrete-time decision processes, depending on which decision time points are allowed.
- Finite or infinite horizon decision problems.
- Total discounted, or average expected reward decision problems when the time horizon is infinite.

In our exposition, we will restrict throughout to stochastic decision processes with finite or countable state spaces. The main objectives of the theory of Markov decision processes are to characterize optimal policies and the optimal value of the problem as well as to provide computational methods to identify optimal policies and the associate values. Basic to all these problems are the *optimality equations* or *Bellman equations*. Depending on the optimality criterion they have different forms (see *Dynamic Programming and Bellman's Principle*).

The theory of Markov decision processes has been extensively developed since the first books by Bellman (1957) and Howard (1960) and is still an active area of research with various open questions. In Sections 2 and 3, we will first deal with finite horizon problems. Some examples are presented and we explain the backward induction algorithm. Infinite horizon problems with discrete-time parameter are considered in Section 4, where we investigate both the expected total reward problem and the expected average reward problem. Finally, Section 5 is concerned with continuous-time Markov decision processes. Throughout the exposition, focus lies on computational methods. Thus, we discuss methods like policy iteration, policy improvement and linear programming.

2. Problem Definition and Examples

We start our investigations with discrete-time sequential decision problems over a finite time horizon. The decision-maker has to choose actions at time points 0, 1, 2, ..., $N-1$. These time points are called *decision epochs* or *stages* and N is the *planning horizon* of the problem. At stage n, the system is in some state $i \in S_n$. S_n is called the *state space* at stage n. Then the controller has to choose an action from the set A_n. However, depending on the current state $i \in S_n$, not all possible actions may be admissible. The *admissible actions* in state i at stage n are collected in a set $A_n(i) \subset A_n$. Upon the choice of an action $a \in A_n(i)$, the decision maker obtains the reward $r_n(i, a)$. Moreover, the action determines the transition law with which the system moves into the new state. $p_n(i, a, j)$ denotes the probability that the next state of the system is $j \in S_{n+1}$, given the system is in state $i \in S_n$ and action a is applied. $p_n(i, a, j)$ is called *transition probability* at stage n. When the final state $i \in S_N$ is reached at the planning horizon, the controller obtains a *final reward* $g_N(i)$. The collection of the objects $(S_n, A_n, p_n, r_n, g_N)$ is called an N-stage *stochastic dynamic program* or *Markov decision process.* It is assumed that all state spaces S_n are finite or countable and that all reward functions r_n and g_N are bounded from above.

A classical example for a Markov decision process is an *inventory control* problem. The situation is here as follows. Since the demand for a product is random, a warehouse will keep an inventory of this product to cope with this uncertainty. Every month, the manager has now to decide how much items of the product should be ordered, based on the knowledge of the current stock level. Of course, the order should not be too high, since keeping the inventory is expensive. On the other hand, the order should not be too small, since this could lead to lost sales or penalties for being unable to satisfy the customers demand. Thus, the manager faces the problem of finding an order policy that maximizes the profit.

We will next give a formulation of this and further problems in the framework of stochastic dynamic programming.

Example 2.1 (Inventory Control Problem with Backlogging)

To get a simple mathematical formulation of the inventory control problem, we will make the following assumptions: order decisions take place at the beginning of each month and the delivery occurs instantaneously. All orders, which arrive during one month, are satisfied at the end of the month. The capacity of the warehouse is B units. In this example, we assume that customer orders, which could not be satisfied in one period, are backlogged for the next month. Thus, the inventory can get negative.

Let i_n be the inventory on hand at stage n which is the beginning of the $(n+1)$th month. i_n can take values in the set $S_n = \{..., -1, 0, 1, ..., B\}$. An order of a_n units costs $c_n a_n$. The action space is $A_n = \mathbb{N}_0$. Since the warehouse capacity is restricted to B, not more than $B - i_n$ items can be ordered. Thus $A_n(i_n) = \{0, 1, ..., B - i_n\}$. The holding and penalty costs are $L_n(i_n) \geq 0$ when the stock level is i_n. The immediate reward or negative cost of the system is therefore $r_n(i_n, a_n) := -(c_n a_n + L_n(i_n + a_n))$. The inventory after N months is worth d per unit, which gives a final reward of $d \cdot i_N$, when the final stock level is i_N. The demand for the items in each month is random. Let $q_n(x)$ be the probability that x units are ordered in the $(n+1)$th month. It is assumed that the sequential demands are stochastically

independent. The inventory at stage $n + 1$ is thus given by $i_{n+1} = i_n + a_n - x_n$. This implies that the transition probability at stage n is given by

$$p_n(i, a, j) = \begin{cases} q_n(i + a - j), & j \le i + a \\ 0, & j > i + a \end{cases} \tag{1}$$

$p_n(i, a, j)$ is the probability that the inventory is equal to j, given the inventory one month ago is i and a units have been ordered in the $(n + 1)$th month.

Example 2.2 (Inventory Control Problem without Backlogging)

In this example, we assume that demands, which cannot be satisfied immediately, are lost. The difference in the mathematical formulation to the preceding example is as follows. The possible inventories at the beginning of the $(n + 1)$th month are given by $S_n = \{0, 1, ..., B\}$. If the demand in the $(n + 1)$th month is equal to x_n, then the inventory at the beginning of the next month is given by $i_{n+1} = \max(0, i_n + a_n - x_n)$. Hence, the transition probability is given by

$$p_n(i, a, j) = \begin{cases} \sum_{x \ge i+a} q_n(x), & j = 0 \\ q_n(i + a - j), & 1 \le j \le i + a. \\ 0, & j > i + a \end{cases} \tag{2}$$

Example 2.3 (Replacement Problem)

Replacement problems are another class of typical examples for Markov decision processes. The problem is as follows. A technical system (e.g. a machine) is in use over N years and its condition deteriorates randomly. The reward of the machine depends on its condition. Each year the manager has to decide whether or not the system should be replaced by a new one. It is assumed that a replacement occurs without time delay. Suppose the state i_n gives the condition of the system at the beginning of the $(n+1)$th year. S_n is a countable set with $0 \in S_n$. If $i_n = 0$, then the system is new. At the beginning of the $(n+1)$th year, the manager has to decide upon the replacement. The action space can be defined by $A_n = A = \{0, 1\}$ with the interpretation that if $a_n = 1$, the system is replaced by a new one and if $a_n = 0$, no replacement is undertaken. Obviously a replacement is always allowed, hence $A_n(i) = A$. Now suppose $q_n(i, j)$ gives the conditional probability of deterioration of the machine condition in the $(n + 1)$th year. The transition probability is thus defined by

$$p_n(i, a, j) := \begin{cases} q_n(0, j), & a = 1 \\ q_n(i, j), & a = 0 \end{cases} \tag{3}$$

The reward function is given by r_n. At the end of the planning horizon N, the machine can be sold and a reward $g_N(i_N)$, depending on the state i_N is obtained. The aim of the management is to find a replacement policy in order to maximize the companies' profit.

A *decision rule* is a function f_n: $S_n \to A_n$ with $f_n(i) \in A_n(i)$ for all $i \in S_n$. It chooses an

admissible action at stage n depending on the state of the system. A decision rule of this type is called *Markovian*, since the decision depends only on the current state and not on the whole history. Moreover, the actions are chosen with certainty. Because of this, the decision rule is called *deterministic*. There are cases where it is necessary to work with more general decision rules, which are history dependent and/or randomized. A randomized decision rule specifies in a given state a probability distribution on the set of admissible actions. Thus, a random experiment has to be carried out, in order to determine the action. There are certain problems where only randomized decision rules are optimal (see also *Stochastic Games*). In our exposition, we will restrict to deterministic and Markovian decision rules.

A *policy* specifies the decision rules that have to be applied at the different stages. It is a sequence of decision rules and denoted by $\pi = (f_0, f_1, ..., f_{N-1})$. If policy $\pi = (f_0, ..., f_{N-1})$ is given and if the system is in state $i_n \in S_n$, the controller has to select action $f_n(i_n) \in D_n(i_n)$ and the system moves to a new state i_{n+1} with probability $p_n(i_n, f_n(i_n), i_{n+1})$. Formally, the evolution of the system under a given policy π is described by a sequence of random variables $X_0, X_1, ..., X_N$ which forms a (non-stationary) Markov chain with transition probabilities

$$p_{f_n}(i_n, i_{n+1}) := p_n(i_n, f_n(i_n), i_{n+1}) \tag{4}$$

for $n = 0, 1, ..., N - 1$.

A stochastic dynamic program is called *stationary*, if the state spaces, the action spaces, the set of admissible actions and the transition probabilities are independent of n and the reward functions r_n and g_N have the form

$$r_n(i\,a) = \beta^n r(i,a) \text{ and } g_N(i) = \beta^N g(i) \tag{5}$$

for some factor $ß \in (0, \infty)$. In this case, we drop the subscript n and the problem is determined by the objects $(S, A, p, r, g, ß)$. These models play an important role, when infinite horizon decision problems are considered.

3. Finite Horizon Decision Problems

In this section, we consider finite horizon Markov decision problems. We explain the important role of the Bellman equation and show how optimal policies can be computed with the backward induction algorithm. Under some assumptions, the monotonicity of optimal policies is studied.

Given an initial state and a policy, a random stream of rewards is generated. As usual in the literature, we suppose that the objective is to maximize the expected sum of the rewards. More precisely, let $i \in S_0$ be the initial state, π a given policy and N the planning horizon. $X_0, X_1, ..., X_N$ is the sequence of random states which are visited. The *expected total reward* over the planning horizon if policy π is used and the system starts in state i is given by

$$v_{N\pi}(i) = \mathbb{E}_{\pi i}\left[\sum_{k=0}^{N-1} r_k(X_k, f_k,(X_k)) + g_N(X_N)\right], \tag{6}$$

where $\mathbb{E}_{\pi i}$ denotes the expectation with respect to the probability measure which is defined by policy π and initial state i. Of course, the aim is to find a policy which maximizes the expected reward. This maximal expected reward is given by

$$V_N(i) := \sup_{\pi} V_{N\pi}(i), i \in S_0 . \tag{7}$$

The function $V_N(i)$ is called value function or optimal value of the N-stage Markov decision process. A policy π^* is called optimal if $V_{N\pi^*}(i) = V_N(i)$ for all states $i \in S_0$. If S_n and A_n are finite for every stage n, such an optimal policy obviously exists and can be found by enumeration. In general, such an optimal policy does not necessarily exist. In this case, one might be content with an ε-optimal policy. This is a policy that satisfies $V_{N\pi}(i) \geq V_N(i) - \varepsilon$ for all states $i \in S_0$.

-
-
-

Bibliography

Bellman R.E. (1957). *Dynamic Programming*, 342 pp. Princeton, NJ: Princeton University Press. [First textbook on deterministic and stochastic dynamic models.]

Bertsekas D.P. (1995). *Dynamic Programming and Optimal Control.* Belmont, MA: Athena Scientific. [Textbook on deterministic and stochastic dynamic models.]

Hernández-Lerma O. and Lasserre J.B. (1996) *Discrete-Time Markov Control Processes*, 216 pp. New York: Springer-Verlag. [Recent book on Markov decision processes with general state spaces.]

Howard, R.A. (1960). *Dynamic Programming and Markov Processes*, 136 pp. Cambridge, MA: MIT Press. [Early textbook on Markov decision processes].

Puterman M. (1994) *Markov Decision Processes*, 649 pp. New York: Wiley. [Textbook on Markov decision processes with countable state spaces.]

Sennott L.I. (1999) *Stochastic Dynamic Programming and the Control of Queues*, 328 pp. New York: John Wiley & Sons. [This book considers Markov decision processes with queueing applications and has a strong computational emphasis.]

Biographical Sketch

Ulrich Rieder, born in 1945, received the Ph.D. degree in mathematics in 1972 (University of Hamburg) and the Habilitation in 1979 (University of Karlsruhe). Since 1980, he has been Full Professor of Mathematics and head of the Department of Optimization and Operations Research at the University of Ulm. His research interests include the analysis and control of stochastic processes with applications in

telecommunication, logistics, applied probability, finance and insurance. Dr. Rieder is Editor-in-Chief of the journal *Mathematical Methods of Operations Research.*