Advanced Stochastic Operations Research

Stochastic Decision Theory

Lecture Notes

Fall, 2009

Stochastische Beslissingstheorie

In vele praktische problemen moeten strategische beslissingen genomen worden zonder dat het effect van die beslissingen op voorhand exact bekend zijn. Markov Beslissingstheorie biedt een raamwerk om problemen het hoofd te kunnen bieden waarbij de te nemen beslissingen het verloop van een *stochastisch proces* kunnen beïnvloeden. In dit vak zullen we naast de theorie ook een veelvoud aan (gestileerde) toepassingsvoorbeelden de revue laten passeren. Vereiste voorkennis is de elementaire theorie van Markov ketens en Markov processen. Aan de hand van voorbeelden wordt de theorie van (semi-)Markov processen met beslissingen ontwikkeld.

De volgende onderwerpen zullen worden behandeld: stochastische dynamische programmering voor problemen met een eindige planningshorizon; de optimaliteitsconditie van Bellman; maximalisatie van gemiddelde of verdisconteerde verwachte opbrengst (oneindige horizon); de methodes van successieve approximatie, policy iteratie, waarde-iteratie en lineaire programmering. Toepassingen van de theorie komen van voorraad-, productie- en wachtrijsystemen. Aanbevolen tekstboeken zijn Derman [1], Howard [2], Putterman [3], Ross [4, Hoofdstuk 6] en Tijms [5, Hoofdstuk 3].

Stochastic Decision Theory

In practice, decisions are often made without a precise knowledge of their impact on future behavior of systems under consideration. The field of Markov Decision Theory has developed a versatile approach to study and optimize the behavior of random processes by taking appropriate actions that influence future evolution. Besides theory, this course also contains many application examples. The course assumes knowledge of basic concepts from the theory of Markov chains and Markov processes. Guided by examples, the theory of (semi-)Markov processes with decisions is presented.

The following topics are covered: stochastic dynamic programming in problems with finite decision horizons; the optimality condition of Bellman; maximization of average or discounted expected reward (infinite horizon); the methods of successive approximation, policy iteration, value-iteration and linear programming. Applications are taken from inventory, production and queueing systems. Basic references to introductory textbooks are Derman [1], Howard [2], Putterman [3], Ross [4, Chapter 6] and Tijms [5, Chapter 3].

Rudesindo Núñez-Queija
November 1, 2009

Contents

Chapter 1

Finite horizon decision problems

In this chapter we shall treat stochastic decision problems defined over a finite period. A finite planning horizon arises naturally in many decision problems. Sometimes the planning period is exogenously pre-determined. We shall see examples of both cases.

This section also serves as an introduction into the basic concepts of Markov decision theory and into notation that shall be used in the remainder. We begin by examining an example.

1.1 Example: Investment

Suppose an investor has €10,000 available and must decide on how to invest it, so as to maximize his expected returns. The investor may choose between investing all of his capital either in stock from company A or in stock from company B. Investing in company A renders a profit of 100% (i.e., a doubling of the investment) after one year with probability 0.10. With probability 0.90, however, there is no profit after one year, and the investor will get his €10,000 back. Company B has a higher risk profile, but also renders higher expected returns. With probability 0.6 the investment is doubled, whereas with probability 0.4 the investment is completely lost. Indeed, the expected profit from investing €10,000 in company A is $0.10 \times 10,000 = 1,000$ euro and in company B it is $0.6 \times 10,000 + 0.4 \times (-10,000) = 2,000$ euro.

If not bankrupt, the investor can re-invest his money every year (each time €10,000 due to the popularity of both investments), what is the best strategy for the investor if his goal is to maximize the expected profit after five years?

In this example the planning horizon is exogenously given and equal to five decision epochs. Clearly, the decisions in later years depend on the profit made during the first year. A *decision rule* assigns a sequence of decisions (one for each year) for each possible outcome of the process. While not bankrupt, the investor must choose between the two possible investments. (In principle the investor could

choose not to invest, but this is not an interesting option in this example, since investment A implies no risk on the invested capital.) Every year the capital either remains unchanged, increases by €10,000 or decreases by €10,000 (the last two are only possible if the investor has not yet gone bankrupt). Thus, after T years the capital can be either 0, 10,000, ..., $(T+1) \times 10,000$ euro. In principle, a decision rule must return a decision at each stage *for every possible sequence of previous decisions and outcomes of the investments so far*. For this simple example this amounts to 640 possible combinations. Application of the technique of *dynamic programming*, however, can drastically reduce the number of relevant decision rules. This technique will be described more in detail in Section 1.3, but here we already illustrate it for this example.

The key idea is to realize that we do know what to do at the last stage of the decision sequence. If the investor has not gone bankrupt before the last decision stage — let's say he has a capital $K_4 \geq 10,000$ euro — in order to maximize the expected returns he should invest €10,000 in B, making the expected final capital $K_4 + 2,000$ euro. With this information we can also determine the optimal decision at the previous to last decision stage. Suppose the capital at that point equals $K_3 \geq 10,000$ euro. Investing in A, leads to a capital of either K_3 or $K_3 + 10,000$ euro. In both cases we know that in the next stage it is optimal to invest in B which results in an expected profit of €2,000. Thus, the expected final capital if we invest in A equals $0.9 \times (K_3 + 2,000) + 0.1 \times (K_3 + 10,000 + 2,000) = K_3 + 3,000$ euro. Next, we evaluate the expected final capital if investment in B is chosen in the previous to last decision stage. The capital increases either to $K_3 + 10,000$ euro or decreases to $K_3 - 10,000$ euro. A distinction must be made between $K_3 = 10,000$ euro and $K_3 \geq 20,000$ euro, because with a capital of €10,000, investment in B may lead to bankruptcy, disabling any future revenues. By similar arguments as before we may conclude that the expected final capital after investing in B in the previous to last decision stage equals $0.6 \times (K_3 + 10,000 + 2,000) + 0.4 \times (K_3 - 10,000 + 2,000) = K_3 + 4,000$ euro if $K_3 \geq 20,000$ and it equals $0.6 \times (10,000 + 10,000 + 2,000) + 0.4 \times 0 = 13,200$ euro if $K_3 = 10,000$. In either case, the expected return is larger than that of investing in A, thus at the previous to last decision stage it is also optimal to invest in B, if not bankrupt.

Of course, we can repeat the same arguments to decide what to do yet one decision stage earlier, and so on. The results of this process are listed in Table 1.1. The table only reports information which is relevant to answer the question raised

capital	$\max E[K_T \mid K_{T-n}]$ (action)				
K_{T-n}	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
0	0 (-)	0 (-)	0 (-)	0 (-)	0 (-)
10,000	12,000 (B)	13,200 (B)	14,400 (B)	15,528 (A)	16,711.20 (A)
20,000	22,000 (B)	24,000 (B)	25,680 (B)	27,360 (B)	
30,000	32,000 (B)	34,000 (B)	36,000 (B)		
40,000	42,000 (B)	44,000 (B)			
50,000	52,000 (B)				

Figure 1.1: Optimal decisions and corresponding returns for $T = 5$

(i.e., what is a good strategy over a period of 5 years starting with €10,000). The n-th decision epoch before the evaluation moment is labeled with n (e.g., 1 corresponds to the last decision stage). The evaluation moment itself is marked with 0. With K_n we denote the capital at stage n. To illustrate how to read the table, let us focus on the following. If the investor has €20,000 with four decision epochs remaining, the maximal attainable expected final capital equals $27,360 = 0.6 \times 36,000 + 0.4 \times 14,400$ and this is (expectedly) achieved if the investor next chooses to invest in B. Indeed, with probability 0.6 there is a profit of $10,000$ after which the maximal expected final capital (over three years) is €36,000, and with probability 0.4 a loss of €10,000 is incurred leaving us with a maximal expected final capital of €14,400.

The answer to our problem can now be read from the table. The investor can maximally expect to end up with €16,711.20. This maximum can (expectedly) be achieved by investing €10,000 in A at the first stage.

1.2 The model

Before formalizing the technique illustrated in the example of the previous section, we introduce some notation. We shall assume there is a stochastic (discrete-time) process X_n, $n = 0, 1, 2, \ldots$ on a state space $\mathcal{I}$. The probabilistic law according to which the process evolves in time depends on a sequence of actions A_n, $n = 0, 1, 2, \ldots$, with $A_n \in \mathcal{A}$.

Assumption 1.2.1 *The state space $\mathcal{I}$ is countable and the action space $\mathcal{A}$ is finite.*

In general, it may be the case that, when $X_n = i$, only a subset of actions $A_n \in \mathcal{A}_i \subset \mathcal{A}$ are allowed. We further specifically assume that

$$\begin{aligned} &\mathbb{P}\{X_{n+1} = i_{n+1} \,|\, X_0 = i_0, \ldots, X_n = i_n,\, A_0 = a_0, \ldots, A_n = a_n\} \\ &= \mathbb{P}\{X_{n+1} = i_{n+1} \,|\, X_n = i_n,\, A_n = a_n\} \\ &=: p^{a_n}(i_n, i_{n+1}), \qquad i_0, \ldots, i_{n+1} \in \mathcal{I},\, a_0 \in \mathcal{A}_{i_0}, \ldots, a_n \in \mathcal{A}_{i_n}. \end{aligned} \tag{1.1}$$

This relation tells us that if the state at time n and the action taken at time n are known, then the state at time $n+1$ is *independent of the history*

$$H_{n-1} := (X_0, A_0, \ldots, X_{n-1}, A_{n-1})$$

of the process before time n. Note that if $p^a(i,j)$ are independent of the action $a \in \mathcal{A}_i$ for all $i, j \in \mathcal{I}$, then the process $X_1, X_2, \ldots$, is a *Markov chain*. If $p^a(i,j)$ does depend on a but the actions $A_0 = a_0$, $A_1 = a_1, \ldots$, are *deterministically* known a priory, then $X_1, X_2, \ldots$, is a *time dependent* Markov chain (with transition probabilities $p^{a_n}(i,j)$ at time n).

Remark 1.2.1 *The notation used for the probability $p^a(i,j)$ and the random variables X_n and A_n shall also be adopted in the sequel. In general, superscripts refer*

to actions (or collections of actions), subscripts are time indexes and states are function arguments. Note however that, e.g., $p^{a_n}(i,j)$ depends on time through a_n, the action taken at time n.

Suppose a reward $r^a(i,j)$ is earned whenever the process X_n is in state i, action a is taken and the process moves to state j. Then

$$r^a(i) := \sum_{j \in \mathcal{I}} p^a(i,j)\, r^a(i,j), \tag{1.2}$$

represents the expected reward if action a is taken while in state i. In many problems, the $r^a(i)$ may be specified without knowledge of $r^a(i,j)$.

To facilitate the analysis, we shall make the following technical assumption, that is commonly satisfied in applications.

Assumption 1.2.2 *The expected rewards are uniformly bounded, i.e., there exists an $R > 0$ such that $\|r^a(i)\| < R$ for all $i \in \mathcal{I}$, $a \in \mathcal{A}_i$.*

In this chapter we shall be interested in choosing the actions such that the expected total reward over a finite period, say $T \in \{1, 2, \ldots\}$ time units, is maximized. In addition to $r^a(i,j)$, suppose a final (expected) reward $q(i)$ is incurred at time T if $X_T = i$. We shall consider a large class of possible strategies for choosing the subsequent actions $A_0, \ldots, A_{T-1}$. Letting $h_{n-1} = (i_0, a_0, \ldots, i_{n-1}, a_{n-1})$ denote a particular history at time n, we define a *strategy* $\mathbf{s}$ as a sequence of functions

$$s_n^{a_n}(h_{n-1}, i_n) \in (0,1), \qquad n = 0, 1, \ldots.$$

(For $n = 0$ this reads $s_0^{a_0}(i_0)$.) If strategy $\mathbf{s}$ is used, then if $X_n = i$ and $H_{n-1} = h_{n-1}$ then at time n action a is taken with probability $s_n^a(h_{n-1}, i)$. Naturally,

$$\sum_{a \in \mathcal{A}_i} s_n^a(h_{n-1}, i) = 1,$$

for all n, h_{n-1} and i.

Remark 1.2.2 *In general, the actions prescribed by a strategy are allowed to depend on the entire history and they may be non-deterministic. So if H_{n-1} and X_n are known, for instance $H_{n-1} = h_{n-1}$ and $X_n = i_n$, but A_n is not yet known, then X_{n+1} may depend on the entire history too. By (1.1), as soon as A_n is known, for instance $A_n = a_n$, the next state X_{n+1} only depends on a_n and i_n.*

If a strategy prescribes a deterministic action to be taken at all times for each possible state and history then we call it a *decision rule* rather than a strategy. More precisely, if for all n, h_{n-1} and i_n there exists precisely one $a \in \mathcal{A}_{i_n}$ such that

$$s_n^a(h_{n-1}, i_n) = 1,$$

then we define the equivalent (non-stationary) *decision rule* $\mathbf{f} = (f_0, f_1)$, where $f_0, f_1, \ldots$, is a sequence of functions such that

$$f_n(h_{n-1}, i_n) = a,$$

precisely for that choice of a for which $s_n^a(h_{n-1}, i_n) = 1$. The decision rule $\mathbf{f}$ is called a Markov decision rule if f_n does not depend on h_{n-1}, in which case we write $f_n(h_{n-1}, i_n) = f_n(i_n)$. Additionally, a Markov decision rule is called *stationary* if $f_0 = f_1 = f_2 = \ldots$.

Let us denote the expected total reward up to time T when starting at time 0 in state i and using strategy $\mathbf{s}$ by

$$V_T^{\mathbf{s}}(i) := \sum_{n=0}^{T-1} \mathbb{E}^{\mathbf{s}} \left[r^{A_n}(X_n, X_{n+1}) \,\middle|\, X_0 = i \right] + \mathbb{E}^{\mathbf{s}} \left[q(X_T) \,\middle|\, X_0 = i \right]. \tag{1.3}$$

The superscript in the expectation $\mathbb{E}^{\mathbf{s}}[\cdot]$ reflects the fact that the strategy determines the probability law according to which the process (X_n, A_n) evolves. Because of (1.2) we may write equivalently

$$V_T^{\mathbf{s}}(i) = \sum_{n=0}^{T-1} \mathbb{E}^{\mathbf{s}} \left[r^{A_n}(X_n) \,\middle|\, X_0 = i \right] + \mathbb{E}^{\mathbf{s}} \left[q(X_T) \,\middle|\, X_0 = i \right]. \tag{1.4}$$

1.3 Bellman's optimality condition

Suppose our goal is to maximize $V_T^s(i)$ over all strategies $\mathbf{s}$. Bellman's optimality condition — or, equivalently, the stochastic dynamic programming optimality condition — given in (1.5) paves the way to determining an optimal strategy, which turns out to be a (non-stationary) decision rule!

Theorem 1.3.1 *Let $V_0^*(i) := q(i)$ and $V_n^*(i)$, $n = 1, 2, \ldots$, be recursively given by*

$$V_n^*(i) := \max_{a \in \mathcal{A}_i} \left\{ r^a(i) + \sum_{j \in \mathcal{I}} p^a(i,j) V_{n-1}^*(j) \right\},$$

then,

$$V_n^*(i) = \sup_s V_n^s(i), \qquad n = 0, 1, \ldots, \tag{1.5}$$

and any (there may be more than one) decision rule $\mathbf{f}_n = (f_n, f_{n-1}, \ldots, f_1)$, determined by

$$f_n(i) = \mathrm{argmax}_{a \in \mathcal{A}_i} \left\{ r^a(i) + \sum_{j \in \mathcal{I}} p^a(i,j) V_{n-1}^*(j) \right\},$$

attains this optimal reward over the first n periods.

Proof A proof by induction on n is given in Appendix A.1. □

Remark 1.3.1 *We emphasize that, given an optimal strategy for n periods, to determine an optimal strategy for the $n+1$-period maximization, we only need to compute f_{n+1} and then use the optimal strategy for the n-period optimization.*

Example 1.3.1 Consider a Markov Decision problem with two states (0 and 1) and two decisions (1 and 2) in each state. The direct reward function is given by $r^1(0) = 1$, $r^2(0) = 0$ and $r^1(1) = r^2(1) = 2$ and the transition probabilities by $p^1(0,0) = \frac{1}{2}$, $p^1(1,0) = \frac{2}{3}$, $p^2(0,0) = \frac{1}{4}$ en $p^2(1,0) = \frac{1}{3}$. We assume a finite planning horizon and final costs $q(0) = 2$ and $q(1) = 1$.
We determine the *minimal* costs over a period with two decision epochs and the corresponding optimal strategy:

$V_0(i) = q(i) \Rightarrow V_0(0) = 2,\ V_0(1) = 1.$
$V_1(0) = \min\left\{1 + \frac{1}{2}\cdot 2 + \frac{1}{2}\cdot 1, 0 + \frac{1}{4}\cdot 2 + \frac{3}{4}\cdot 1\right\} = \frac{5}{4};$
$V_1(1) = \min\left\{2 + \frac{2}{3}\cdot 2 + \frac{1}{3}\cdot 1, 2 + \frac{1}{3}\cdot 2 + \frac{2}{3}\cdot 1\right\} = \frac{10}{3};$
$V_2(0) = \min\left\{1 + \frac{1}{2}\cdot \frac{5}{4} + \frac{1}{2}\cdot \frac{10}{3}, 0 + \frac{1}{4}\cdot \frac{5}{4} + \frac{3}{4}\cdot \frac{10}{3}\right\} = 2\frac{13}{16};$
$V_2(1) = \min\left\{2 + \frac{2}{3}\cdot \frac{5}{4} + \frac{1}{3}\cdot \frac{10}{3}, 2 + \frac{1}{3}\cdot \frac{5}{4} + \frac{2}{3}\cdot \frac{10}{3}\right\} = 3\frac{17}{18};$

Thus, the optimal strategy for two periods is: $f_1(0) = 2$, $f_1(1) = 2$, $f_2(0) = 2$, $f_2(1) = 1$.

Example 1.3.2 Inventory control
A storage depot is used to keep production items in stock. At most 2 items can be stored at the same time. At the end of each week, the inventory level (i.e., the number of items in stock) is monitored and a decision is made about the number of new items to be ordered from the production facility. An order that is placed on Friday is delivered on Monday at 7.30 a.m. The cost of an order consist of a fixed amount of €100 and an additional €100 per ordered item. Requests for items arrive randomly at the storage depot: With probability $\frac{1}{4}$ there is no demand during a week, with probability $\frac{1}{2}$ exactly one item is requested during a week and with probability $\frac{1}{4}$ the weekly demand equals 2 items. If the weekly demand exceeds the inventory stock, it is fulfilled directly from the production facility at the expense of €300 per item. The depot manager wishes to minimize the expected ordering costs over a pre-determined finite horizon planning period. The items in stock at the end of the planning period render no value.

a) *Formulate the above problem as a Markov Decision problem. (What are the state space, action space, direct rewards, final rewards and transition probabilities?)*

b) *Determine (for each possible initial state) the minimal expected cost over a period of 2 weeks.*

c) *Suppose the value of each item in stock at the end of the planning period of 2 weeks equals q euro. For which value(s) of q does the optimal strategy change?*

1.4 Maximizing an entrance probability

So far we have been concerned with maximizing expected rewards. In many applications it is required that the probability of reaching some particular state i' within T time units be maximized. We shall see such an example below. If we define the *first entrance time* into the state i' by

$$\tau(i') = \inf_n \{n \geq 1 : X_n = i'\},$$

our objective is thus to maximize

$$P_T^{\mathbf{s}}(i) := \mathbb{P}^{\mathbf{s}}\{\tau(i') \leq T \,\big|\, X_0 = i\} \tag{1.6}$$

over all strategies $\mathbf{s}$ and for all initial states i. At first sight it may not be obvious that such a problem fits the framework described in Section 1.2. In principle, after having entered the state i', the process X_n may again move to other states. Since the posterior evolution after having visited i' does not alter the criterion function $P_T^s(i)$, we may as well require that the process is *absorbed* in state i' after having visited it. More precisely, we may set $p^a(i', i') = 1$ and $p^a(i', i) = 0$ for all actions $a \in \mathcal{A}_{i'}$ and all initial states i. Now take, for all $i \in \mathcal{I}$, $i \neq i'$,

$$r^a(i, i') = 1, \qquad a \in \mathcal{A}_i,$$

and $r^a(i, j) = 0$ for all other choices of i, j and a. In particular $r^a(i', i') = 0$ for all $a \in \mathcal{A}_{i'}$. Obviously,

$$\sum_{n=0}^{T-1} r^{A_n}(X_n, X_{n+1})$$

equals 1 if $X_n = i'$ for at least one $n = 1, \ldots, T$, and it equals 0 otherwise. Therefore, we have

$$P_T^s(i) = \mathbb{E}^s \left[\sum_{n=0}^{T-1} r^{A_n}(X_n, X_{n+1}) \,\big|\, X_0 = i \right] = \sum_{n=0}^{T-1} \mathbb{E}^s \left[r^{A_n}(X_n, X_{n+1}) \,\big|\, X_0 = i \right].$$

Taking $q(i) = 0$ for all $i \in \mathcal{I}$ and using the definition in (1.3) we have $P_T^s(i) = V_T^s(i)$. We can thus maximize $P_T^s(i)$ using Theorem 1.3.1.

Example 1.4.1 In the lectures, it is shown how to maximize the probability of ending up with at least €20,000 for the investment problem described in Section 1.1.

Example 1.4.2 Roulette
An amateur gambler goes to the casino to play roulette with a budget of €75. In each round, he chooses to play either black or red. Therefore, in each round, the probability of doubling the bet is 18/37 and the probability of losing the bet is 19/37. Each round, the gambler places a bet with an (integer) amount of euros. The goal is to maximize the probability of taking home at least €200.

a) Formulate this game as a Markov Decision Problem, assuming a finite planning horizon T. (What are the state space, the action space, the direct rewards, the final rewards and the transition probabilities?)

b) Formulate the optimality equation for the probability of ending up with at least €200.

c) Determine the optimal strategy for $T = 2$.

d) Write a computer program that computes the optimal strategies and the corresponding maximal probabilities $P(X_T \geq 200)$ for arbitrary values of $T \in \{1, 2, 3, \ldots\}$. (Also hand in a copy of the code.)

e) What is the optimal first action (for each possible initial bet) if $T = 60$? (Print using computer program.)

f) Determine the (minimal) value of T such that, beyond this value the optimum does not alter by further increasing T. (Indication: this value does not exceed 100.)

g) What is the maximum probability of going home with at least €200 (assuming $T = \infty$)?

1.5 Towards an infinite horizon

If we let $T \to \infty$ in (1.6) the limit represents the probability of ever reaching state i' by using strategy $\mathbf{s}$. Letting $T \to \infty$ in (1.4), however, the limit in general may not be well-defined. To start with, it is not clear what the "final" rewards $q(\cdot)$ represent. But even if we take $q(i) \equiv 0$, there may be other problems. For instance, if $r(i) \geq r$ for all $i \in \mathcal{I}$ and some $r > 0$. In that case $V_T^{\mathbf{s}}(i) \to \infty$ when $T \to \infty$ for all initial states i and all strategies $\mathbf{s}$. In such cases the *total reward* criterion is not appropriate. In Chapters 2 and 3 we shall investigate two alternative criterions: that of maximization of the *average reward* and the *total discounted rewards*, respectively.

Chapter 2

Average reward criterion

One way of dealing with an infinite planning horizon is to maximize the *average reward*. For a fixed strategy $\mathbf{s}$ we define the average reward after starting in state i as

$$g^{\mathbf{s}}(i) := \limsup_{T\to\infty} \frac{V_T^{\mathbf{s}}(i)}{T} = \limsup_{T\to\infty} \frac{1}{T} \sum_{n=0}^{T-1} \mathbb{E}^{\mathbf{s}} \left[r^{A_n}(X_n) \,\middle|\, X_0 = i \right]. \tag{2.1}$$

(As argued in Section 1.5 we shall take $q(\cdot) \equiv 0$.) In (2.1) we take the $\limsup$ to avoid technicalities regarding the existence of the limit. We further define

$$g^*(i) := \sup_{\mathbf{s}} g^{\mathbf{s}}(i), \tag{2.2}$$

and we are interested in finding a strategy (if it exists) that attains this maximum average reward. We shall see that for a broad class of models not only such a strategy exists, but that we can even find an *optimal stationary decision rule*. It is also convenient to define

$$g^* := \sup_{i\in\mathcal{I}} g^*(i), \tag{2.3}$$

which is the maximally attainable average reward if we are also allowed to choose the initial state. As we shall see, often the average reward does not depend on the initial state and we simply have $g^* \equiv g^*(i)$.

From Theorem 1.3.1 we know how to maximize the *total reward* over a finite period. Naturally, this implies that we also know how to maximize the average reward over a finite pre-determined period, since the two criterions are equivalent. The trouble with determining what to do in the case of an infinitely long planning period is that the decision rule $(f_n, f_{n-1}, \ldots, f_1)$ is — in general — *not stationary*, i.e., in general the decision rules f_n and f_m do not coincide when $m \neq n$. Still, for n large enough, it seems that maximizing the total reward over n periods is a natural approximation to maximize the eventual average reward. But this is true for any decision epoch! Hence, *repeatedly* using f_n, i.e., using the decision rule $(f_n, f_n, f_n, \ldots)$ seems a good thing to do in order to approximately attain the maximum average reward.

Indeed, if $\lim_{n\to} f_n$ exists, it can be shown that this limiting *stationary decision rule* attains the maximal average reward. But why should such a rule exist? Indeed, in general this may not be the case. The subject of this chapter is to show that for a large class of problems there does exists an optimal stationary rule and to develop techniques to compute that rule. In the first section we recall basic properties of (stationary) Markov chains that have an important implication for the average reward of stationary decision rules. Then — in Section 2.2 — we develop some more intuition to guide us through the technical derivations in later sections.

2.1 Average reward using stationary decision rules

What can we say about the average reward when using the stationary decision rule $\mathbf{f} = (f, f, f, \ldots)$? The stationarity of the decision rule implies that X_n, $n = 0, 1, 2, \ldots$, is a Markov chain[1] with transition probabilities

$$p^{\mathbf{f}}(i,j) := \mathbb{P}^{\mathbf{f}}\{X_{n+1} = j \,\big|\, X_n = i\} = \mathbb{P}\{X_{n+1} = j \,\big|\, X_n = i, A_n = f(i)\}.$$

It shall be useful to also define the n-step transition probabilities

$$p_n^{\mathbf{f}}(i,j) := \mathbb{P}^{\mathbf{f}}\{X_n = j \,\big|\, X_0 = i\}.$$

By definition $p_1^{\mathbf{f}}(i,j) = p^{\mathbf{f}}(i,j)$ and by conditioning on the state after one step we have for $n = 2, 3, \ldots$,

$$p_n^{\mathbf{f}}(i,j) = \sum_{k\in\mathcal{I}} p^{\mathbf{f}}(i,k)\, p_{n-1}^{\mathbf{f}}(k,j). \tag{2.4}$$

(These recursive equations are known as the Chapman-Kolmogorov equations.)

Let $T^{\mathbf{f}}(i, i_0)$ be the expected time it takes X_n to reach some fixed reference state $i_0 \in \mathcal{I}$ starting from state i:

$$T^{\mathbf{f}}(i, i_0) = \mathbb{E}^{\mathbf{f}}\left[\inf\{n \geq 1 : X_n = i_0\} \,\big|\, X_0 = i\right].$$

Assumption 2.1.1 *The Markov chain with transition probabilities $p^{\mathbf{f}}(i,j)$ is aperiodic and the reference state i_0 and $T_0^{\mathbf{f}} < \infty$ can be chosen such that $T^{\mathbf{f}}(i, i_0) < T_0^{\mathbf{f}}$ for all $i \in \mathcal{I}$. In particular, the reference state i_0 is positive recurrent.*

If this assumption is satisfied, there exists a unique stationary distribution $\pi^{\mathbf{f}}(j)$, $j \in \mathcal{I}$ and we may write

$$\pi^{\mathbf{f}}(j) = \lim_{n\to\infty} \mathbb{P}^{\mathbf{f}}\{X_n = j \,\big|\, X_0 = i\},$$

independent of i. In particular, we have for the average reward that

$$g^{\mathbf{f}}(i) \equiv g^{\mathbf{f}} := \sum_{j\in\mathcal{I}} \pi^{\mathbf{f}}(j) r^{\mathbf{f}}(j),$$

for all $i \in \mathcal{I}$.

[1] We shall use results that were developed in the course Stochastic Processes 1 (*Stochastische Processen* 1).

2.2 Heuristics

Let the sequence f_n again be generated from Bellman's optimality equation. Suppose that we are in the case where $f^* := \lim_{n\to} f_n$ exists and that $\mathbf{f}^* = (f^*, f^*, f^*, \ldots)$ satisfies Assumption 2.1.1. Hence, $g^{\mathbf{f}^*}(i) \equiv g^{\mathbf{f}^*}$ is indeed independent of i. This has interesting consequences on the rewards generated by using $\mathbf{f}^*$ over finite periods. Since the action space is finite (Assumption 1.2.1) and $f_n(i) \to f^*(i)$, it must be that there exists some N_i such that $f_n(i) = f^*(i)$ for all $n > N_i$. Let us assume something stronger: there exists an N such that $f_n(i) = f^*(i)$ for all i and $n > N$. So after N steps of Bellman's optimization procedure, the policy maximizing the right hand side of (1.5) is fixed at f^*. For $n \geq 1$ we therefore have

$$\begin{aligned} V^*_{N+n}(i) &= r^{\mathbf{f}^*}(i) + \sum_{j\in\mathcal{I}} p^{\mathbf{f}^*}(i,j) V^*_{N+n-1}(j) \\ &= \max_{a\in\mathcal{A}_i} \left\{ r^a(i) + \sum_{j\in\mathcal{I}} p^a(i,j) V^*_{N+n-1}(j) \right\}. \end{aligned} \tag{2.5}$$

Moreover,

$$V^*_{N+n}(i) = V^{\mathbf{f}^*}_n(i) + \sum_{j\in\mathcal{I}} p^{\mathbf{f}^*}_n(i,j) V^*_N(j), \tag{2.6}$$

where $p^{\mathbf{f}^*}_n(i,j)$ are the n-step transition probabilities that are recursively given by the Chapman-Kolmogorov equations (2.4). In words, (2.6) says that the maximal revenue over $N+n$ periods is the sum of the revenue of the stationary rule $\mathbf{f}^*$ over the first n periods and the maximal revenue over the last N periods.

We know that $V^{\mathbf{f}^*}_n(i)$ will grow approximately as $n\, g^{\mathbf{f}^*}$, for all i. For $n \geq 1$, let us therefore look at the *relative rewards* over finite periods defined by

$$d^{\mathbf{f}^*}_n(i, i_0) := V^{\mathbf{f}^*}_n(i) - V^{\mathbf{f}^*}_n(i_0), \tag{2.7}$$

where $i_0 \in \mathcal{I}$ is an arbitrary chosen reference state. Substituting (2.6) and (2.7) into (2.5) gives

$$\begin{aligned} &d^{\mathbf{f}^*}_n(i,i_0) + V^{\mathbf{f}^*}_n(i_0) - V^{\mathbf{f}^*}_{n-1}(i_0) + \sum_{j\in\mathcal{I}} p^{\mathbf{f}^*}_n(i,j) V^*_N(j) \\ &= \max_{a\in\mathcal{A}_i} \left\{ r^a(i) + \sum_{j\in\mathcal{I}} p^a(i,j) \left(d^{\mathbf{f}^*}_{n-1}(j,i_0) + \sum_{k\in\mathcal{I}} p^{\mathbf{f}^*}_{n-1}(j,k) V^*_N(k) \right) \right\} \end{aligned} \tag{2.8}$$

and taking $a = f^*(i)$ attains the maximum on the right hand side. We shall now let $n \to \infty$ in this equation. Note that (see Section 2.1)

$$\lim_{n\to\infty} \sum_{j\in\mathcal{I}} p^{\mathbf{f}^*}_n(i,j) V^*_N(j) = \sum_{j\in\mathcal{I}} \pi^{\mathbf{f}^*}(j) V^*_N(j),$$

independent of i (the interchange of the limit and the summation is allowed because of Assumption 1.2.2). The corresponding terms on the left and right hand side of (2.8) will therefore cancel out when $n \to \infty$.

Let us then concentrate on the remaining terms. While $V_n^{\mathbf{f}^*}(i)$ may grow unboundedly, the same need not hold for $d_n^{\mathbf{f}^*}(i, i_0)$. In fact, we shall see in Section 2.4, under the assumptions made so far, that

$$d^{\mathbf{f}^*}(i, i_0) := \lim_{n\to\infty} d_n^{\mathbf{f}^*}(i, i_0), \tag{2.9}$$

is well defined and that, as one would expect intuitively (ultimately, each period renders a reward equal to the average),

$$\lim_{n\to\infty} V_n^{\mathbf{f}^*}(i) - V_{n-1}^{\mathbf{f}^*}(i) = g^{\mathbf{f}^*},$$

for all $i \in \mathcal{I}$, in particular for $i = i_0$. Let us assume that this is true and that, when passing $n \to \infty$ in (2.8), we may interchange the order of the limit and the summation (in the right hand side). We then have

$$d^{\mathbf{f}^*}(i, i_0) + g^{\mathbf{f}^*} = \max_{a\in\mathcal{A}_i} \left\{ r^a(i) + \sum_{j\in\mathcal{I}} p^a(i,j) d^{\mathbf{f}^*}(j, i_0) \right\}, \tag{2.10}$$

and $a = f^*(i)$ attains the maximum in the right hand side.

We have thus argued heuristically that if Bellman's recursive procedure ultimately gives a stationary decision rule, then the average revenue and the relative rewards corresponding to this rule satisfy the functional equation (2.10). All entities in this equation have a clear probabilistic interpretation and the relation itself says that the optimal decision rule (assuming there exists one) has the property that if one is forced to use rule f^* in all subsequent steps, then one can not do better than when using rule f^* in the first step too.

In the remainder of this chapter we shall rigorously prove the assertions that were loosely stated in this section. In all that follows, the intuition provided here will play an important role.

2.3 Optimality condition

If one can find a solution to the functional equation (2.10), then this solution actually determines a stationary decision rule that renders the maximal average reward. This statement is formalized in the next theorem.

Theorem 2.3.1 *[4, Thm. 6.17] If there exists a bounded function $d(i)$, $i \in \mathcal{I}$, and a constant g such that, for all $i \in \mathcal{I}$,*

$$d(i) + g = \max_{a\in\mathcal{A}_i} \left\{ r^a(i) + \sum_{j\in\mathcal{I}} p^a(i,j)\, d(j) \right\}, \tag{2.11}$$

then $g = g^*(i)$ for all $i \in \mathcal{I}$ and any stationary decision rule $\mathbf{f} = (f, f, f, \ldots)$ that satisfies

$$f(i) \in \operatorname{argmax}_{a\in\mathcal{A}_i} \left\{ r^a(i) + \sum_{j\in\mathcal{I}} p^a(i,j)\, d(j) \right\},$$

renders the maximally attainable average reward: $g^{\mathbf{f}}(i) = g^*$.

Proof See Appendix A.2. □

So, if we can find a solution to (2.11) then we are done. From Section 2.2 we also have an idea how to go about determining such a solution: by repeatedly applying Bellman's optimality equation. Doing exactly what is suggested in Section 2.2 is, however, not feasible: when can we conclude that the decision rules will not change anymore? In principle, even after many iterations giving the same rule, future rules may still change at some point. It may even be the case that Bellman's equation does not give a converging sequence of rules, however many iterations we do. And do the bounded function $d(i)$ and the constant g of Theorem 2.3.1 exist at all? These matters will be resolved in what follows.

2.4 Relative rewards of stationary decision rules

Before investigating the problem of finding an optimal strategy, we first study the Markov chain induced by a stationary decision rule $\mathbf{f}$ in more detail. In Section 2.1 we saw that under Assumption 2.1.1 the average reward does not depend on the initial state: $g^{\mathbf{f}}(i) \equiv g^{\mathbf{f}}$. In Section 2.2 we informally introduced the concept of *relative rewards*. We shall now give a rigorous treatment of this important notion. Again, suppose that Assumption 2.1.1 is satisfied. Let $R^{\mathbf{f}}(i, i_0)$ be the expected reward until the first visit to the reference state i_0, starting from state i. Because of Assumptions 1.2.2 and 2.1.1 we have $\|R^{\mathbf{f}}(i, i_0)\| \leq R\,T_0^{\mathbf{f}} < \infty$. Furthermore, by the Renewal Reward Theorem[2] (consecutive visits to the reference state i_0 constitute regeneration points):

$$g^{\mathbf{f}} = \frac{R^{\mathbf{f}}(i_0, i_0)}{T^{\mathbf{f}}(i_0, i_0)}. \tag{2.12}$$

The *relative reward* of state i over state i_0 is now defined as

$$d^{\mathbf{f}}(i, i_0) := R^{\mathbf{f}}(i, i_0) - g^{\mathbf{f}}\,T^{\mathbf{f}}(i, i_0). \tag{2.13}$$

We shall see later that this definition is equivalent with that in (2.9), i.e., $d^{\mathbf{f}}(i, i_0) = \lim_{n\to\infty} d_n^{\mathbf{f}}(i, i_0)$, where $d_n^{\mathbf{f}}(i, i_0)$ is defined similarly as in (2.7).

So $d^{\mathbf{f}}(i, i_0)$ is the difference between the expected total reward that can be attained while moving from i to i_0 and the expected accumulated reward over a period of equal (expected) length if exactly $g^{\mathbf{f}}$ is incurred every time unit. The relative reward of state i over an arbitrary state j can now be defined as[3] $d^{\mathbf{f}}(i, i_0) - d^{\mathbf{f}}(j, i_0)$. Note that from (2.12) it follows that $d^{\mathbf{f}}(i_0, i_0) = 0$.

[2] See the course Stochastic Processes 1 (*Stochastische Processen* 1).

[3] It is left to the reader to verify that if j is a positive recurrent state and j would have been chosen as the reference state, then the relative reward of state i over state j would have been the same: $d^{\mathbf{f}}(i, j) = d^{\mathbf{f}}(i, i_0) - d^{\mathbf{f}}(j, i_0)$.

By conditioning on the state of the process after one transition, we may write

$$T^{\mathbf{f}}(i,i_0) = 1 + \sum_{j \in \mathcal{I}\setminus\{i_0\}} p^{\mathbf{f}}(i,j) T^{\mathbf{f}}(j,i_0), \tag{2.14}$$

and similarly,

$$R^{\mathbf{f}}(i,i_0) = r^{\mathbf{f}}(i) + \sum_{j \in \mathcal{I}\setminus\{i_0\}} p^{\mathbf{f}}(i,j) R^{\mathbf{f}}(j,i_0). \tag{2.15}$$

Multiplying (2.14) by $g^{\mathbf{f}}$, subtracting it from (2.15) and using that $d^{\mathbf{f}}(i_0,i_0) = 0$ we have

$$d^{\mathbf{f}}(i,i_0) + g^{\mathbf{f}} = r^{\mathbf{f}}(i) + \sum_{j \in \mathcal{I}} p^{\mathbf{f}}(i,j) d^{\mathbf{f}}(j,i_0). \tag{2.16}$$

Theorem 2.4.1 *[5, Thm.3.1.1] If some bounded function $d(i)$ and some constant g satisfy*

$$d(i) + g = r^{\mathbf{f}}(i) + \sum_{j \in \mathcal{I}} p^{\mathbf{f}}(i,j) d(j), \tag{2.17}$$

then $g = g^{\mathbf{f}}$ and $d(i) = d^{\mathbf{f}}(i,i_0) + c$ for some constant c.

Proof The fact that $g = g^{\mathbf{f}}$ follows from Theorem 2.3.1. The remainder of the proof is given in Appendix A.3. □

Theorem 2.4.1 says that, apart from a constant shift in the function $d^{\mathbf{f}}(i,i_0)$, the set of linear equations in (2.16) admits a unique solution. Note the similarity between (2.11) which must be satisfied by an optimal stationary decision rule (if it exists) and (2.16) which is satisfied by any stationary decision rule that meets Assumption 2.1.1. These two functional equations give rise to two different algorithms to compute (or approximate) an optimal strategy: the algorithm of *Policy Iteration* and that of *Successive Approximations* . These algorithms are discussed in Sections 2.5 and 2.6, respectively. In Section 2.7 we shall treat a third method relying on linear programming. These algorithms are discussed without assuming finiteness of $\mathcal{I}$. We emphasize, however, that implementation of each of these algorithms requires either a finite state space or truncation of the state space, except in some special cases.

2.5 Policy Iteration

Suppose that we are given some stationary decision rule $\mathbf{f}$ and that we were able to compute the average reward and the relative rewards corresponding to it by solving (2.16). What we actually want is a stationary decision rule that satisfies (2.11), since such a rule will render the maximal average reward. The idea of Policy Iteration (PI) is to simply apply the maximization procedure on the right hand side of (2.11) to the relative values obtained from (2.16). Hopefully, the maximizing decisions give us a better policy. This indeed turns out to be the case as is stated in the next theorem. (This theorem can be sharpened as we shall see in Lemma 2.6.1.)

Theorem 2.5.1 *Let $\mathbf{f} = (f, f, f, \ldots)$ be a stationary decision rule satisfying Assumption 2.1.1 with relative rewards $d^{\mathbf{f}}(i, i_0)$. Let $\mathbf{f}' = (f', f', f', \ldots)$ be a stationary decision rule also satisfying Assumption 2.1.1 and*

$$f'(i) \in \operatorname{argmax}_{a \in \mathcal{A}_i} \left\{ r^a(i) + \sum_{j \in \mathcal{I}} p^a(i,j)\, d^{\mathbf{f}}(j, i_0) \right\}. \tag{2.18}$$

Then $g^{\mathbf{f}'} \geq g^{\mathbf{f}}$.

Proof See Appendix A.4. □

It is helpful to realize what (2.18) actually means. If we are forced to use f from time 1 onward, then f' chooses the best actions at time 0. The theorem then says that if we always use f', we are not worse off. Thus by improving on *one step* we in fact improve on the *total future*!

Theorem 2.5.2 *If $\mathbf{f} = \mathbf{f}'$ satisfies (2.18) then $\mathbf{f}$ is a decision rule that gives the maximal average reward.*

Proof By (2.18) and (2.16), the average reward and the relative reward function of $\mathbf{f}$ also satisfy (2.11). Hence, by Theorem (2.3.1) $\mathbf{f}$ is optimal. □

This theorem says that if we can take $f' = f$ in (2.18) then we have found an optimal strategy. We thus have the following algorithm.

Policy Iteration (PI) Algorithm

This algorithm consists of the following steps.

0. Set $n := 0$. Choose any initial stationary decision rule $\mathbf{f}_0$.
1. Compute the average reward and the relative reward function of $\mathbf{f}_n$ by solving (2.16).
2. Put $\mathbf{f} = \mathbf{f}_n$ and compute $\mathbf{f}_{n+1} = \mathbf{f}'$ from (2.18), taking $\mathbf{f}' = \mathbf{f}$ if possible.
3. If $\mathbf{f}_{n+1} = \mathbf{f}_n$ then this strategy is optimal, otherwise set $n := n+1$ and repeat steps 1, 2 and 3.

Remark 2.5.1 *Note that step 1 requires solving a set of linear equations. In actual implementations of this algorithm, this is only feasible when the number of states is not too large. In case of infinite state spaces, one usually needs to truncate the state space.*

Theorem 2.5.3 *If we can not take $\mathbf{f}' = \mathbf{f}$ in (2.18) then, one of the two following holds*

(i) $g^{\mathbf{f}'} > g^{\mathbf{f}}$;

(ii) $d^{\mathbf{f}'}(i, i_0) \geq d^{\mathbf{f}}(i, i_0)$ *for all* $i \in \mathcal{I}$ *and* $d^{\mathbf{f}'}(i, i_0) > d^{\mathbf{f}}(i, i_0)$ *for at least one state* $i \in \mathcal{I}$.

Proof See Appendix A.5. □

Corollary 2.5.1 *If the number of states in* $\mathcal{I}$ *is finite, then the PI algorithm converges in finitely many steps.*

Proof In each step either a new rule is computed or the algorithm has converged. If a new rule is computed then, because of Theorem (2.5.3), either the average reward increases or the relative reward function increases strictly for one state and does not decrease for other states. Hence, the algorithm can not return to the same rule after more than one step. Since the number of states and the number of actions are finite, also the number of stationary decision rules is finite. So after finitely many steps the algorithm must converge. □

In practice, the number of iterations needed for convergence of PI is usually very small, especially if the initializing strategy is cleverly chosen. As noted in Remark (2.5.1), however, step 1 in the iterations may be very expensive, or even infeasible. In the next section we discuss an approach that is less sensitive to the number of states but, in contrast with PI, may require a large number of iterations to get satisfactory results.

Example 2.5.1 Inventory control *(see Section 2.9 for solutions)*
A class of expensive goods kept in stock at a warehouse are sold directly to customers. The inventory level can be increased by placing a new order at the beginning of each period. Lead times are negligible, so that we assume that any order is available immediately. At most three items can be kept on stock. Items that are not sold at the end of the period, can be kept for the next period, but imply an inventory cost of $h = 4$ units per item. An order of n items costs $K + rn$ units; $K = 4$ and $r = 2$ are the fixed and the variable ordering costs. If the demand in a period exceeds the number of items in stock, a penalty cost of $b = 12$ units is incurred per item that can not be delivered. The demands in subsequent periods form a sequence of independent and identically distributed random variables. Each period, the demand equals 0, 1, 2 or 3 items; each of these possibilities occurs with probability $\frac{1}{4}$. Future costs of lost demand is already accounted for in the penalty cost p, so that the goal is to minimize the average cost per period.

(Hint to simplify the calculations: If the i-th and j-th rows of the matrix B are identical and $y = z + By$ then the column vectors y and z satisfy $y_i - z_i = y_j - z_j$.)

a) Formulate this problem as a Markov Decision Problem: Describe the state space and the possible actions; determine the direct costs and the transition probabilities.

b) Suppose the stock level is increased to its maximum level of 3 items at the beginning of each period (if there are already 3 items in stock, then no order is placed). What are the average cost and the relative values of this strategy?

c) Do *one* step of the *policy iteration* algorithm, starting with the strategy described in question b. It suffices to determine the new strategy. (The average costs and the relative values of the new strategy need not be determined.)

d) Show that it is optimal to order 3 items when there is no item in stock, 2 items when there is 1 item in stock, and that no order should be placed if there are 2 (or 3) items in stock.

Example 2.5.2 Machine repair
A production facility has 3 machines. If a machine starts up correctly in the morning, it renders a daily production of 1 euro. A machine that does not start up correctly needs to be repaired. A visit of a repair man costs 3 euro per day. The repair man repairs *all broken machines* in the same day (the repair cost is 3 euro, independent of the number of machines repaired). A machine that has been repaired always starts up correctly the next day. The probability distribution of the number of machines that start up correctly the next day depends on the number of presently working machines. This probability distribution is given in the table below, where m stands for the number of (presently) working machines and n stands for the number of those that start up correctly the next day.

m	$n=0$	$n=1$	$n=2$	$n=3$
1	$\frac{1}{2}$	$\frac{1}{2}$	0	0
2	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0
3	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

Example: If there are $m=2$ machines working today, then the probability that exactly one of those 2 ($n=1$) starts up again the next morning (and the other does not) equals $\frac{1}{3}$.

a) Formulate this problem as a Markov Decision Problem: Describe the state space and action space; determine the direct rewards/costs and the transition probabilities.

b) Suppose it is decided that no machine should ever be repaired. What are the average rewards and corresponding relative values?

c) Apply the policy iteration *algorithm* once, *starting with the strategy in part b. It is sufficient to determine the new strategy. (The average rewards and relative values corresponding to that strategy need* not *be determined.)*

d) Show that it is optimal to only let the repair man come when all *machines are broken.*

2.6 Successive Approximation

The Successive Approximation (SA) algorithm is inspired by (2.11) alone, or, in fact, by Bellman's optimality equation which underlies (2.11).

Successive Approximation Algorithm

0. Set $n := 0$. Choose an $\epsilon > 0$ and any bounded function $v_0(i)$, $i \in \mathcal{I}$. (A common choice is $v_0(i) \equiv 0$.)

1. Compute

$$v_{n+1}(i) := \max_{a \in \mathcal{A}_i} \left\{ r^a(i) + \sum_{j \in \mathcal{I}} p^a(i,j) v_n(j) \right\} \tag{2.19}$$

and let

$$f_{n+1}(i) \in \operatorname{argmax}_{a \in \mathcal{A}_i} \left\{ r^a(i) + \sum_{j \in \mathcal{I}} p^a(i,j) v_n(j) \right\}. \tag{2.20}$$

2. Let $M_n := \max_{i \in \mathcal{I}} \{v_n(i) - v_{n-1}(i)\}$ and $m_n := \min_{i \in \mathcal{I}} \{v_n(i) - v_{n-1}(i)\}$. Stop the algorithm if $M_n - m_n < \epsilon$. Otherwise set $n := n + 1$ and repeat steps 1, and 2.

Choosing $v_0(i) \equiv 0$ has the advantage that we can interpret $v_n(i)$ as the maximal reward over n periods when starting in state i (and using a — possibly non-stationary — decision rule determined by Bellman's optimality equation (1.5). If $v_0(i) \neq 0$ for some i, we may still use this interpretation if we introduce (artificial) "final" rewards $q(i) := v_0(i)$, $i \in \mathcal{I}$ as in Chapter (1). Thus, SA does exactly what we argued heuristically in Section 2.2, but now we shall prove that, under some conditions, this procedure indeed converges to an optimal stationary decision rule. Additionally, SA provides bounds on the average reward of $\mathbf{f}_n = (f_n, f_n, f_n, \ldots)$ obtained in the n-th iterate, as is shown in the next theorem.

Theorem 2.6.1 *[5, Thm. 3.4.1] Let $v_n(i)$ be obtained from (2.19) and $f_n(i)$ from (2.20). Let M_n and m_n be as in the SA algorithm. If $\mathbf{f}_n = (f_n, f_n, f_n, \ldots)$ satisfies Assumption 2.1.1 then*

$$m_n \leq g^{\mathbf{f}_n} \leq g^* \leq M_n,$$

and m_n is non-decreasing and M_n is non-increasing.

Proof See Appendix **??**. □

The proof of Theorem 2.6.1 uses the next technical lemma, which is a stronger statement then that of Theorem 2.5.1.

Lemma 2.6.1 *[5, Thm. 3.2.1] Let* $\mathbf{f} = (f, f, f, \ldots)$ *be a stationary decision rule satisfying Assumption 2.1.1 and suppose that, for some constant* g *and some bounded function* $v(i)$, $i \in \mathcal{I}$,

$$r^{\mathbf{f}}(i) + \sum_{j \in \mathcal{I}} p^{\mathbf{f}}(i,j)\, v(j) \geq v(i) + g, \qquad i \in \mathcal{I},$$

then $g^{\mathbf{f}} \geq g$. *Similarly, if*

$$r^{\mathbf{f}}(i) + \sum_{j \in \mathcal{I}} p^{\mathbf{f}}(i,j)\, v(j) \leq v(i) + g, \qquad i \in \mathcal{I},$$

then $g^{\mathbf{f}} \leq g$.

Proof See Appendix **??**. □

Example 2.6.1 A two-state MDP *(see Section 2.9 for solutions)*
Consider a Markov Decision Problem with two states (0 and 1) and two actions (1 and 2) in each state. The direct-cost function is given by $r^1(0) = 1$, $r^2(0) = 0$ and $r^1(1) = r^2(1) = 2$ and the transition probabilities are $p^1(0,0) = \frac{1}{2}$, $p^1(1,0) = \frac{2}{3}$, $p^2(0,0) = \frac{1}{4}$ and $p^2(1,0) = \frac{1}{3}$. For a finite planning horizon, the final costs are $q(0) = 2$ and $q(1) = 1$.

a) Determine the *minimum* cost over a period with two *decision epochs*. What is the corresponding optimal strategy?

Now suppose we want to minimize the *average* cost for an infinite planning horizon. (Naturally, there are no final rewards.)

b) Find the strategy corresponding to the second iteration of the *successive approximation* algorithm. Also determine the corresponding upper and lower bounds for the average cost.

c) Determine the average cost and the corresponding relative values of the strategy found in part b.

d) Carry out one step of the *policy iteration* algorithm, starting with the strategy of part b.

Example 2.6.2 Drill platform
The maximum daily output of a drill platform in the North sea is 10 million euros per day. For security reasons the process is paused at night. It is possible that the interruption leads to pollution of the installation, giving a daily production of only 5 million euro. If that's the case it is possible to clean the installation, at the cost of loosing the production of one day. The cleaning costs are negligible.

The probability that the installation is polluted after a day of maximum production is $\frac{1}{3}$ (and with probability $\frac{2}{3}$ the installation can work at full capacity). A polluted installation which is not cleaned, remains polluted. Cleaning the installation has the desired effect with probability $\frac{1}{2}$ (the installation then works at full capacity the next day), but with probability $\frac{1}{2}$ the installation remains polluted (the next day it can be decided again to clean the installation). The aim is to maximize the average output.

a) Formulate this problem as a Markov decision problem: Describe the state and action spaces and give the transition probabilities and direct rewards.

b) Carry out one step of the successive approximation algorithm. Give the corresponding candidate strategy, as well as lower and upper bounds for the optimal rewards.

c) Explain whether the algorithm will converge in this example. Motivate your answer by discussing possible problems with this algorithm.

d) Compute the optimal strategie and the corresponding maximum reward (numerically, using a computer programm).

2.7 Linear Programming Approach

We now discuss a third approach to compute optimal strategies using a Linear Programming formulation (LP). Let us concentrate on a fixed stationary randomized strategy $\mathbf{s} = (s, s, s, \ldots)$ where $s^a(i)$ is the probability of choosing action a in state i. Clearly, any stationary decision rule $\mathbf{f} = (f, f, f, \ldots)$ falls within this class of strategies, specifically by taking $s^{f(i)}(i) = 1$ and $s^a(i) = 0$ for all $a \neq f(i)$. Like stationary decision rules, a stationary randomized strategy gives rise to a Markov chain X_n with stationary transition probabilities

$$p^{\mathbf{s}}(i,j) = \sum_{a \in \mathcal{A}_i} s^a(i) \sum_{j \in \mathcal{I}} p^a(i,j).$$

Similar to Section 2.1, we define $T^{\mathbf{s}}(i, i_0)$ as the expected time it takes X_n to reach some fixed reference state $i_0 \in \mathcal{I}$ starting from state i:

$$T^{\mathbf{s}}(i, i_0) = \mathbb{E}^{\mathbf{s}} \left[\inf \{n \geq 1 : X_n = i_0\} \,\middle|\, X_0 = i \right].$$

The following parallels Assumption 2.1.1.

Assumption 2.7.1 *The Markov chain with transition probabilities $p^{\mathbf{s}}(i,j)$ is aperiodic and the reference state i_0 and $T_0^{\mathbf{s}} < \infty$ can be chosen such that $T^{\mathbf{s}}(i,i_0) < T_0^{\mathbf{s}}$ for all $i \in \mathcal{I}$.*

Naturally, if this assumption is satisfied, there exists a unique stationary distribution $\pi^{\mathbf{s}}(j)$, $j \in \mathcal{I}$, which is the unique function satisfying $\sum_{j\in\mathcal{I}} \pi^{\mathbf{s}}(j) = 1$ and the balance equations

$$\pi^{\mathbf{s}}(j) = \sum_{i\in\mathcal{I}} \pi^{\mathbf{s}}(i)\, p^{\mathbf{s}}(i,j). \tag{2.21}$$

When using a fixed stationary randomized strategy satisfying Assumption 2.7.1 we can also define the *fraction of time points that X_n is in state i and action a is chosen*:

$$\pi^{\mathbf{s}}(i,a) := \pi^{s}(i)\, s^{a}(i), \qquad i \in \mathcal{I},\ a \in \mathcal{A}_i. \tag{2.22}$$

Naturally, $\pi^{\mathbf{s}}(i) = \sum_{a\in\mathcal{A}_i} \pi^{\mathbf{s}}(i,a)$. Similar to (2.21) we have the following set of equations

$$\sum_{a\in\mathcal{A}_j} \pi^{\mathbf{s}}(j,a) = \sum_{i\in\mathcal{I}} \sum_{a\in\mathcal{A}_i} \pi^{\mathbf{s}}(i,a) p^{a}(i,j).$$

The average reward satisfies

$$g^{\mathbf{s}}(i) \equiv g^{\mathbf{s}} := \sum_{j\in\mathcal{I}} \pi^{\mathbf{s}}(j) r^{\mathbf{s}}(j) = \sum_{j\in\mathcal{I}} \sum_{a\in\mathcal{A}_j} \pi^{\mathbf{s}}(j,a) r^{a}(j),$$

for all $i \in \mathcal{I}$. An average reward maximizing stationary randomized strategy can be obtained from the following LP.

Linear Program Formulation

$$\max_{x(i,k):i\in\mathcal{I},k\in\mathcal{A}_i} \sum_{j\in\mathcal{I}} \sum_{a\in\mathcal{A}_j} x(j,a) r^{a}(j)$$

subject to

$$\begin{aligned}
&\sum_{a\in\mathcal{A}_j} x(j,a) = \sum_{i\in\mathcal{I}} \sum_{a\in\mathcal{A}_i} x(i,a) p^{a}(i,j), \qquad j \in \mathcal{I}\\
&\sum_{i\in\mathcal{I}} \sum_{a\in\mathcal{A}_i} x(i,a) = 1\\
&x(i,a) \geq 0.
\end{aligned}$$

The object value returned by this program is the maximally attainable average reward. The corresponding optimal solution $x^*(i,a)$ can be decomposed into an optimal stationary randomized strategy $\mathbf{s} = (s,s,s,\ldots)$ and the corresponding stationary distribution:

$$\begin{aligned}
s^{a}(i) &:= \frac{x(i,a)}{\sum_{k\in\mathcal{A}_i} x(i,k)},\\
\pi^{\mathbf{s}}(i) &= \sum_{a\in\mathcal{A}_i} x(i,a).
\end{aligned} \tag{2.23}$$

In fact, there exists an optimal solution $x^*(i,a)$ which has the property that for each i there is exactly one $a \in \mathcal{A}_i$ such that $x^*(i,a) = 1$ and $x^*(i,k) = 0$ for all other $k \in \mathcal{A}_i$. This of course is a consequence of the results in Section 2.5, but it can also independently be proved by standard linear programming theory: the optimum is attained for at least one basic solution. A basic solution can be shown to have $x^*(i,a) = 1$ for some $a \in \mathcal{A}_i$ and $x^*(i,k) = 0$ for all other $k \in \mathcal{A}_i$.

The strength of LP compared to PI and SA is its ability to incorporate restrictions on the allowed strategies. For instance, if it is not allowed that the process X_n resides in state i more than a fraction ϵ of the time, then we can insert the restriction $\sum_{a \in \mathcal{A}_i} x(i,a) \leq \epsilon$ into the LP formulation. The program will then render an optimal stationary randomized strategy (if it exists) under this additional restriction. Neither the PI algorithm, nor the SA algorithm are able to cope with such a restriction! Note also that in this case, it may be so that no stationary decision rule attains the same maximum average reward as the optimal randomized strategy.

The computational effort of LP in finding an optimal solution is comparable to that of PI. Like PI, in each step LP needs to solve systems of linear equations of the same size as the state space.

Example 2.7.1 Two-state MDP (revisited)
Consider again Example 2.6.1. The LP formulation in this case is as follows. Let $x(i,a)$ be the long-run fraction of periods that the state is i *and* action a is chosen. Then, we can obtain the maximum average reward from

$$\max_{x(0,1),x(0,2),x(1,1),x(1,2)} x(0,1) + 2x(1,1) + 2x(1,2)$$

subject to

$$\begin{aligned}
x(0,1) + x(0,2) &= \frac{1}{2}x(0,1) + \frac{1}{4}x(0,2) + \frac{2}{3}x(1,1) + \frac{1}{3}x(1,2) \\
x(1,1) + x(1,2) &= \frac{1}{2}x(0,1) + \frac{3}{4}x(0,2) + \frac{1}{3}x(1,1) + \frac{2}{3}x(1,2) \\
x(0,1) + x(0,2) + x(1,1) + x(1,2) &= 1 \\
x(0,1),\, x(0,2),\, x(1,1),\, x(1,2) &\geq 0.
\end{aligned}$$

It may be verified that solving this LP indeed gives the same solution as with in Example 2.6.1. Suppose now that it is not allowed to take action 2 in more than 10% of the decision epochs. This restriction can be incorporated by adding the constraint

$$x(0,2) + x(1,2) \leq \frac{1}{10}$$

to the LP.

2.8 Generalizations

We have concentrated on the case where stationary decision rules give rise to a Markov chain with one positive recurrent class. When this is not satisfied, the framework does not immediately brake down. However, modifications are needed. For instance, the PI algorithm then requires an additional step that first selects the actions that give rise to the highest average reward. Then the ordinary step 2 is carried out among these actions. Also, the SA and LP approaches need modification. In particular, the lower and upper bounds in the SA algorithm may no longer converge to the same value, thus invalidating one of the most powerful properties of this algorithm. In order to apply the LP approach, additional equations need to be added to the linear program. A rigorous treatment of these extensions is beyond the scope of this course, but some of these issues receive attention either in the lectures or the assignments.

Example 2.8.1 Successive approximation and policy iteration
Consider a stochastic process X_n, $n = 0, 1, 2, \ldots$ with state space $\{1, 2, 3\}$. In states 2 and 3 only one action is possible. The following transition probabilities and direct rewards correspond to this "action": $p^1(2,3) = p^1(3,2) = 1$, $r^1(2) = 10$ and $r^1(3) = 14$. In state 1 there are two possible actions with $p^1(1,1) = 1$, $r^1(1) = 11$ and $r^2(1) = 6$. In addition it is known that $p^2(1,1) < 1$. (If necessary you may make additional assumptions, e.g., $p^2(1,1) = 0$) or even $p^2(1,2) = 1$.)

a) Formulate the algorithms of successive approximation (SA) and policy iteration (PI) for this specific problem.

b) Show that the decision rules generated by the SA algorithm do not converge to an optimal rule.

c) Apply the PI algorithm to this problem.

2.9 Solutions to selected exercises

Example 2.5.1 Inventory control

There are two ways to interpret inventory cost: charge the expected cost for the current period, or charge the cost for the inventory of the previous period. Note that for the average optimality criterion this does not make a difference since all periods are equally important (with discounted costs it would matter). We start with the first.

Alternative 1

a) State at the beginning of period n: X_n = number of items in stock $\in \{0,1,2,3\}$.
Action in state i: number of items ordered $\in \mathcal{A}_i = \{0,1,\ldots,3-i\}$.
In general $r^a(i) = h\mathbf{E}[(i+a-D)^+] + K1_{a>0} + ra + b\mathbf{E}[(D-i-a)^+] = 4\mathbf{E}[(i+a-D)^+]+4\cdot 1_{a>0}+2a+12\mathbf{E}[(D-i-a)^+]$. Note that $\mathbf{E}[(D-i-a)^+] =$ 3/2, 3/4, 1/4, 0 if $i+a =$ 0, 1, 2, 3; and $\mathbf{E}[(i+a-D)^+] =$ 0, 1/4, 3/4, 3/2 if $i+a =$ 0, 1, 2, 3;
So $r^0(i) =$ 18, 10, 6, 6, for $i = 0,1,2,3$; $r^1(i) =$ 16, 12, 12, for $i = 0,1,2$; $r^2(i) =$ 14, 14, for $i = 0,1$; $r^3(i) = 16$.

$$P^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3/4 & 1/4 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}; P^1 = \begin{pmatrix} 3/4 & 1/4 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ \star & \star & \star & \star \end{pmatrix};$$

$$P^2 = \begin{pmatrix} 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ \star & \star & \star & \star \\ \star & \star & \star & \star \end{pmatrix}; P^3 = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ \star & \star & \star & \star \\ \star & \star & \star & \star \\ \star & \star & \star & \star \end{pmatrix}$$

b) $f = (3,2,1,0)$, $r^f = (16,14,12,6)$, $P^f = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$.
Because of the structure we immediately see that each of the four states occurs with equal probability, so $g^f = 1/4(16+14+12+6) = 12$. The relative rewards also follow easily (using the hint): $d^f = (0,-2,-4,-10)$.

c) $f'(0) = \arg\min\{18+0, 16-1/2, 14-3/2, 16-4\} = 3$;
$f'(1) = \arg\min\{10-1/2, 12-3/2, 14-4\} = 0$;
$f'(2) = \arg\min\{6-3/2, 12-4\} = 0$;
$f'(3) = 0$.

d) $f = (3,2,0,0)$, $r^f = (16,14,6,6)$, $P^f = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$.
Using the hint we already know that if $d^f(0) = 0$ then $d^f(1) = -2$ and $d^f(3) = -10$. Using the standard equations: $g^f = 11+1/8$, $d^f = (0,2,-7-1/2,2)$. Apply the policy improvement step once:
$f'(0) = \arg\min\{18+0, 16-1/2, 14-1/4*19/2, 16-1/4*39/2\} = 3$,
$f'(1) = \arg\min\{10-1/2, 12-1/4*19/2, 14-1/4*39/2\} = 2$,
$f'(2) = \arg\min\{6-1/4*19/2, 12-1/4*39/2\} = 0$,
$f'(3) = 0$.
The policy does not change; so it is optimal.

Alternative 2
Now we charge the cost for the inventory of the previous period.

a) State at the beginning of period n: X_n = number of items in stock $\in \{0,1,2,3\}$.
Action in state i: number of items ordered $\in \mathcal{A}_i = \{0,1,\ldots,3-i\}$.
In general $r^a(i) = hi + K1_{a>0} + ra + b\mathbf{E}[(D-i-a)^+] = 4i + 4\cdot 1_{a>0} + 2a + 12\mathbf{E}[(D-i-a)^+]$, and $\mathbf{E}[(D-i-a)^+]$ = 3/2, 3/4, 1/4, 0 if $i+a =$ 0, 1, 2, 3.
So $r^0(i) =$ 18, 13, 11, 12, for $i = 0,1,2,3$; $r^1(i) =$ 15, 13, 14, for $i = 0,1,2$; $r^2(i) =$ 11, 12, for $i = 0,1$; $r^3(i) = 10$.

$$P^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3/4 & 1/4 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}; P^1 = \begin{pmatrix} 3/4 & 1/4 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ \star & \star & \star & \star \end{pmatrix};$$

$$P^2 = \begin{pmatrix} 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ \star & \star & \star & \star \\ \star & \star & \star & \star \end{pmatrix}; P^3 = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ \star & \star & \star & \star \\ \star & \star & \star & \star \\ \star & \star & \star & \star \end{pmatrix}$$

b) $f = (3,2,1,0)$, $r^f = (10,12,14,12)$, $P^f = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$.
Because of the structure we immediately see that each of the four states occurs with equal probability, so $g^f = 1/4(10+12+14+12) = 12$. The relative rewards also follow easily (using the hint): $d^f = (0,2,4,2)$.

c) $f'(0) = \arg\min\{18+0, 15+1/2, 11+3/2, 10+2\} = 3$;
$f'(1) = \arg\min\{13+1/2, 13+3/2, 12+2\} = 0$;
$f'(2) = \arg\min\{11+3/2, 14+2\} = 0$;
$f'(3) = 0$.

d) $f = (3,2,0,0)$, $r^f = (10,12,11,12)$, $P^f = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$.
Using the hint we already know that if $d^f(0) = 0$ then $d^f(1) = 2$ and $d^f(3) = 2$. Using the standard equations: $g^f = 11 + 1/8$, $d^f = (0,2,1/2,2)$. Applying the policy improvement step once, the policy does not change; so it is optimal. □

Example 2.6.1 A two-state MDP

a) $V_0(i) = q(i) \Rightarrow V_0(0) = 2$, $V_0(1) = 1$.

$V_1(0) = \min\left\{1 + \frac{1}{2}\cdot 2 + \frac{1}{2}\cdot 1, 0 + \frac{1}{4}\cdot 2 + \frac{3}{4}\cdot 1\right\} = \frac{5}{4}$;
$V_1(1) = \min\left\{2 + \frac{2}{3}\cdot 2 + \frac{1}{3}\cdot 1, 2 + \frac{1}{3}\cdot 2 + \frac{2}{3}\cdot 1\right\} = \frac{10}{3}$;

$V_2(0) = \min\left\{1 + \frac{1}{2}\cdot \frac{5}{4} + \frac{1}{2}\cdot \frac{10}{3}, 0 + \frac{1}{4}\cdot \frac{5}{4} + \frac{3}{4}\cdot \frac{1}{3}\right\} = 2\frac{13}{16}$;
$V_2(1) = \min\left\{2 + \frac{2}{3}\cdot \frac{5}{4} + \frac{1}{3}\cdot \frac{1}{3}, 2 + \frac{1}{3}\cdot \frac{5}{4} + \frac{2}{3}\cdot \frac{10}{3}\right\} = 3\frac{17}{18}$;

Optimal strategy for two periods: $f_1(0) = 2$, $f_1(1) = 2$, $f_2(0) = 2$, $f_2(1) = 1$.

b) There are two possibilities:

1. The initial value function of the SA algorithm is may be chosen arbitrarily; by choosing $V_0 = q$ we can conclude from part a) that the corresponding strategy is f_2, with $f_2(0) = 2$ and $f_2(1) = 1$.
Since $V_2(0) - V_1(0) = 1\frac{9}{16}$ and $V_2(1) - V_1(1) = \frac{11}{18}$, we have an upper bound $M_2 = 1\frac{9}{16}$ and a lower bound $m_2 = \frac{11}{18}$ for the average cost.

2. If one chooses the standard initialization $V_0(0) = V_0(1) = 0$, one needs to carry out again the same steps as in a). $V_1(0) = 0$, $V_1(1) = 2$, $V_2(0) = \frac{3}{2}$, $V_2(1) = 2\frac{2}{3}$; the strategy then becomes $f_2(0) = 2$ and $f_2(1) = 1$ (i.e., the same strategy as above).
Now $V_2(0) - V_1(0) = \frac{3}{2}$ and $V_2(1) - V_1(1) = \frac{2}{3}$, so that the upper bound is $M_2 = \frac{3}{2}$ and the lower bound is $m_2 = \frac{2}{3}$. (I.e., in this case the standard initialization gives sharper upper and lower bounds.)

c) Strategy $f = f_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, $P^f = \begin{pmatrix} \frac{1}{4} & \frac{3}{4} \\ \frac{2}{3} & \frac{1}{3} \end{pmatrix}$, $r^f = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$.

The relative values and average rewards satisfy

$$\begin{aligned} d(0) + g &= 0 + \frac{1}{4}d(0) + \frac{3}{4}d(1), \\ d(1) + g &= 2 + \frac{2}{3}d(0) + \frac{1}{3}d(1). \end{aligned}$$

$d(0) := 0 \Rightarrow g = \frac{3}{4}d(1) \Rightarrow \left(1 + \frac{3}{4} - \frac{1}{3}\right) d(1) = 2 \Rightarrow d(1) = \frac{24}{17} \Rightarrow g = \frac{18}{17}$.

d) $f'(0) \in \text{argmin}\left\{1 + \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot \frac{24}{17}, 0 + \frac{3}{4}\cdot \frac{24}{17}\right\} = \text{argmin}\left\{1\frac{12}{17}, \frac{18}{17}\right\} = \{2\}$, so that $f'(0) = 2$;
$f'(1)$: $\min\left\{2 + \frac{2}{3}\cdot 0 + \frac{1}{3}\cdot \frac{24}{17}, 2 + \frac{1}{3}\cdot 0 + \frac{2}{3}\cdot \frac{24}{17}\right\} = \{1\} \Rightarrow f'(1) = 1$.
(Since $f = f'$ we may conclude that f is optimal.) □

Chapter 3

Discounted rewards

A second approach to handle an infinite planning horizon is to maximize the *total discounted expected reward*, which for a fixed strategy $\mathbf{s}$ and initial state i is defined as

$$V_\alpha^{\mathbf{s}}(i) := \sum_{n=0}^{\infty} \alpha^n \mathbb{E}^{\mathbf{s}} \left[r^{A_n}(X_n) \,\middle|\, X_0 = i \right]. \tag{3.1}$$

The parameter $\alpha \in (0,1)$ is a fixed pre-determined discount factor. We are interested in determining

$$V_\alpha^*(i) := \sup_{\mathbf{s}} V_\alpha^{\mathbf{s}}(i), \tag{3.2}$$

and a strategy that attains this maximal discounted reward, if it exists.

3.1 Fixed Stationary Decision Rule

Before focussing on the maximization problem (3.2), we first concentrate on the discounted rewards when using a fixed decision rule $\mathbf{f}$. As we shall see in Theorem 3.2.1 there exists a decision rule $\mathbf{f}^*$ that attains the optimal reward, i.e., $V_\alpha^{\mathbf{f}^*}(i) = V_\alpha^*(i)$.

For a fixed decision rule $\mathbf{f}$ and a bounded function $v(i)$, $i \in \mathcal{I}$, we define the mapping $T^{\mathbf{f}}v$ of $v(\cdot)$ as

$$(T_\alpha^{\mathbf{f}} v)(i) := r^{\mathbf{f}}(i) + \alpha \sum_{j \in \mathcal{I}} p^a(i,j) v(j), \quad i \in \mathcal{I}, \tag{3.3}$$

and the n-th, $n = 2, 3, \ldots$, iterate of this mapping is denoted by

$$(T_{\alpha,n}^{\mathbf{f}} v)(i) := r^{\mathbf{f}}(i) + \alpha \sum_{j \in \mathcal{I}} p^a(i,j) (T_{\alpha,n-1}^{\mathbf{f}} v)(j), \quad i \in \mathcal{I}. \tag{3.4}$$

The mapping $T_\alpha^{\mathbf{f}}$ and its iterates have the following important properties:

Lemma 3.1.1 *If $u(\cdot)$ and $v(\cdot)$ are two bounded functions on $\mathcal{I}$ then*

(i) (Monotonicity) If $u(i) \leq v(i)$ *for all* $i \in \mathcal{I}$ *then* $(T_\alpha^{\mathbf{f}} u)(i) \leq (T_\alpha^{\mathbf{f}} v)(i)$ *for all* $i \in \mathcal{I}$*;*

(ii) (Convergence) $\lim_{n\to\infty}(T_{\alpha,n}^{\mathbf{f}} v)(i) = V_\alpha^{\mathbf{f}}(i)$ *for all* $i \in \mathcal{I}$*;*

(iii) (Unique fixed point) $v(i) = V_\alpha^{\mathbf{f}}(i)$ *is the unique solution to the functional equation* $(T_\alpha^{\mathbf{f}} v)(i) = v(i)$ *for all* $i \in \mathcal{I}$*.*

Proof Discussed at the lectures. □

Lemma 3.1.1 has the following corollary which parallels Theorem 2.5.3 in the average reward criterion. The corollary states that, if a stationary decision rule $\mathbf{f}'$ improves on a stationary decision rule $\mathbf{f}$ *in one step* then $\mathbf{f}'$ attains at least the same discounted rewards as $\mathbf{f}$ for all states and has higher discounted rewards for at least one state. In other words: $\mathbf{f}'$ really improves on $\mathbf{f}$.

Corollary 3.1.1 *Suppose that the stationary decision rules* $\mathbf{f}$ *and* $\mathbf{f}'$ *are such that* $(T_\alpha^{\mathbf{f}'} V_\alpha^{\mathbf{f}})(i) \geq V_\alpha^{\mathbf{f}}(i)$ *for all* $i \in \mathcal{I}$ *and* $(T_\alpha^{\mathbf{f}'} V_\alpha^{\mathbf{f}})(i_0) > V_\alpha^{\mathbf{f}}(i_0)$ *for some* $i_0 \in \mathcal{I}$*. Then* $V_\alpha^{\mathbf{f}'}(i) \geq V_\alpha^{\mathbf{f}}(i)$ *for all* $i \in \mathcal{I}$ *and* $V_\alpha^{\mathbf{f}'}(i_0) > V_\alpha^{\mathbf{f}}(i_0)$*.*

3.2 Functional Equation

We now define the mapping T_α^* for any function $v(\cdot)$, bounded on the state space:

$$(T_\alpha^* v)(i) := \max_{a\in\mathcal{A}_i} \left\{ r^a(i) + \alpha \sum_{j\in\mathcal{I}} p^a(i,j) v(j) \right\}, \qquad i \in \mathcal{I}. \tag{3.5}$$

Its n-th iterate is defined by

$$(T_{\alpha,n}^* v)(i) := \max_{a\in\mathcal{A}_i} \left\{ r^a(i) + \alpha \sum_{j\in\mathcal{I}} p^a(i,j) (T_{\alpha,n-1}^* v)(j) \right\}, \qquad i \in \mathcal{I}. \tag{3.6}$$

The following lemma establishes convergence of these iterates.

Lemma 3.2.1 *For any bounded function* $v(i)$*,* $i \in \mathcal{I}$*,*

$$\lim_{n\to\infty} (T_{\alpha,n}^* v)(i) = V_\alpha^*(i),$$

for all $i \in \mathcal{I}$*.*

Proof Discussed at the lectures. (Interpret the n-th iterate as the maximum discounted rewards over n periods with final reward $v(j)$ in state j.) □

Theorem 3.2.1 *$v(i) = V_\alpha^*(i)$, defined by (3.2), is the unique solution to the functional equation $T_\alpha^* v = v$, i.e.,*

$$V_\alpha^*(i) = \max_{a \in \mathcal{A}_i} \left\{ r^a(i) + \alpha \sum_{j \in \mathcal{I}} p^a(i,j) V_\alpha^*(j) \right\}, \qquad i \in \mathcal{I}. \tag{3.7}$$

Any stationary decision rule $\mathbf{f} = (f, f, f, \ldots)$, satisfying

$$f(i) \in \operatorname{argmax}_{a \in \mathcal{A}_i} \left\{ r^a(i) + \alpha \sum_{j \in \mathcal{I}} p^a(i,j) V_\alpha^*(j) \right\}, \qquad i \in \mathcal{I},$$

attains the maximal discounted rewards: $V_\alpha^{\mathbf{f}}(i) = V_\alpha^*(i)$.

Proof Discussed at the lectures. (Uniqueness follows from Lemma 3.2.1 and for optimality it is sufficient to verify that $V_\alpha^{\mathbf{f}}(i)$ satisfies (3.7.) □

3.3 Policy Iteration

Corollary 3.1.1 is the basis for the policy iteration (PI) algorithm in case of discounted rewards. If we determine $\mathbf{f}' = (f', f', \ldots)$ from $\mathbf{f} = (f, f, \ldots)$ using

$$f'(i) \in \operatorname{argmax}_{a \in \mathcal{A}_i} \left\{ r^a(i) + \alpha \sum_{j \in \mathcal{I}} p^a(i,j)\, V_\alpha^{\mathbf{f}}(j) \right\}, \tag{3.8}$$

then, either the conditions of Corollary 3.1.1 are satisfied, or $V_\alpha^{\mathbf{f}}$ satisfies (3.7). This means that either $\mathbf{f}'$ improves on $\mathbf{f}$ or, by Theorem 3.2.1, $\mathbf{f}$ is optimal.

The Policy Iteration Algorithm

0. Set $n := 0$. Choose any initial stationary decision rule $\mathbf{f}_0$.
1. Compute the discounted value function $V_\alpha^{\mathbf{f}_n}$ by solving $V_\alpha^{\mathbf{f}_n} = T_\alpha^{\mathbf{f}_n} V_\alpha^{\mathbf{f}_n}$ as prescribed in Lemma 3.1.1.
2. Put $\mathbf{f} = \mathbf{f}_n$ and compute $\mathbf{f}_{n+1} = \mathbf{f}'$ from (3.8), taking $\mathbf{f}' = \mathbf{f}$ if possible.
3. If $\mathbf{f}_{n+1} = \mathbf{f}_n$ then this strategy is optimal, otherwise set $n := n+1$ and repeat steps 1, 2 and 3.

Remark 2.5.1 also applies to the discounted reward: step 1 requires solving a set of linear equations, which is infeasible if the number of states is large. The number of iterations needed to converge is, however, usually very small. This method is therefore well suited when the state space is not too large. Similar to Corollary 2.5.1 we have the following result:

Corollary 3.3.1 *If the number of states in $\mathcal{I}$ is finite, then the PI algorithm converges in finitely many steps.*

3.4 Successive Approximation

We now discuss the successive approximation (SA) approach for discounted rewards. The computational complexity per iteration using this approach is less sensitive to the number of states than PI, but it may require a large number of iterations to get satisfactory results. Lemma 3.2.1 provides the necessary ingredients to formulate the SA algorithm. From an approximation $v_n(\cdot)$ of $V_\alpha^*(\cdot)$ we can find a new approximation $v_{n+1} := (T_\alpha^* v_n)$:

Successive Approximation Algorithm

0. Set $n := 0$. Choose an $\epsilon > 0$ and any bounded function $v_0(i)$, $i \in \mathcal{I}$. (A common choice is $v_0(i) \equiv 0$.)

1. Compute
$$v_{n+1}(i) := \max_{a \in \mathcal{A}_i} \left\{ r^a(i) + \alpha \sum_{j \in \mathcal{I}} p^a(i,j) v_n(j) \right\} \tag{3.9}$$
and let
$$f_{n+1}(i) \in \text{argmax}_{a \in \mathcal{A}_i} \left\{ r^a(i) + \alpha \sum_{j \in \mathcal{I}} p^a(i,j) v_n(j) \right\}. \tag{3.10}$$

2. Let $M_n := \max_{i \in \mathcal{I}} \{v_n(i) - v_{n-1}(i)\}$ and $m_n := \min_{i \in \mathcal{I}} \{v_n(i) - v_{n-1}(i)\}$. Stop the algorithm if $M_n - m_n < \epsilon$. Otherwise set $n := n + 1$ and repeat steps 1, and 2.

Lemma 3.2.1 ensures that $v_n(i) \to V_\alpha^*(i)$, as $n \to \infty$, for all $i \in \mathcal{I}$. As with average rewards, SA provides bounds on total discounted rewards of $\mathbf{f}_n = (f_n, f_n, f_n, \ldots)$ obtained in the n-th iterate, as is shown in the next theorem.

Theorem 3.4.1 *Let $v_n(i)$ be obtained from (3.9) and $f_n(i)$ from (3.10). Let M_n and m_n be as in the SA algorithm. Then*

$$v_n(i) + \frac{\alpha}{1-\alpha} m_n \leq V_\alpha^{\mathbf{f}_n}(i) \leq V_\alpha^*(i) \leq v_n(i) + \frac{\alpha}{1-\alpha} M_n.$$

Proof Discussed at the lectures. □

Example 3.4.1 Selling a house *(See Section 3.7 for solutions)*
Suppose somebody wants to sell his house. Each day, one potential buyer makes an offer, to which the owner must react immediately. He can either accept or reject the offer (no bargaining is allowed). Each rejection of an offer implies a daily maintenance cost of C euros to the owner. A rejected offer is lost. Each day, the offer equals i euros with probability $p(i)$, $i = 0, 1, 2, \ldots$, independent of all past offers. ($p(0)$ may be interpreted as he probability that no offer is made.)

It may be assumed that the expected value of an offer is finite: $\sum_{j=0}^{\infty} jp_j < \infty$. Future offers are discounted using a fixed daily discount rate of α. The goal is to maximize the expected total discounted rewards.

a) Formulate this problem as a Markov Decision Problem: Describe the state space and the possible actions; also determine the direct rewards/costs and the transition probabilities. (Hint: introduce an absorbing state "∞" which is reached after accepting an offer.)

b) Formulate the optimality equation for the maximum expected discounted rewards and use it to show that the optimal strategy is a "threshold strategy". (A threshold strategy is characterized by a threshold value $i' \geq 0$ such that all offers $i \leq i'$ are rejected and the first offer larger than i' is accepted.)

c) Suppose a threshold strategy f_0 with threshold value $i_0 \geq 0$ is used. Show that, for all initial states $i \leq i_0$, the discounted reward function of f_0 satisfies

$$V_\alpha^{f_0}(i) = V_\alpha^{f_0}(0)$$

(the value itself need not be determined) and apply the policy improvement strategy once. Show that the new strategy is again a threshold strategy (and denote its threshold value by i_1).

d) Repeating the policy iteration step, we thus obtain a sequence of threshold strategies with threshold values i_0, i_1, $i_2, \ldots$. Use the (strict) monotonicity property of the policy iteration algorithm (i.e., the property that subsequent value functions are monotonically increasing) and the fact that $V_\alpha^{f_n}(i_n) \geq i_n$, for $n > 0$, to show that $i_1 \leq i_2 \leq i_3 \leq \ldots$. (Warning: it may not be true that $i_0 \leq i_1$.)

Example 3.4.2 Race horse

The owner of a race horse wants to maximize the (discounted) returns of his horse. The (daily) discount factor is $\frac{2}{3}$. It is possible to participate in a race every day, but after participating, the horse may not be fit the next day. If the horse is fit, the expected returns for that day are 2 million euros. If the horse is still tired, the expected returns are only 1 million euros. Participation in a match is for free. If the horse is fit and participates in a match, it is again fit the next day with probability $\frac{2}{3}$ and with probability $\frac{1}{3}$ it is still tired the next day. If the horse is fit and does not participate in a race, it will still be fit the next day. Similarly, the horse will not be fit the next day, if it participates in race while it is not fit. If a tired horse rests for a day, it will be fit the next day with probability $\frac{1}{2}$ and it is still tired the next day with probability $\frac{1}{2}$.

a) Formulate this problem as a Markov Decision Problem: Describe the state and action spaces and give the transition probabilities and direct rewards.

b) Apply two steps of the successive approximation algorithm. In each step give the corresponding candidate strategy, as well as lower and upper bounds for the optimal discounted rewards.

c) Show that it is optimal to let the horse participate every day and determine the optimal discounted rewards.

If, instead of the discounted rewards, we wish to maximize the long-term average rewards, it turns out it is not optimal any more to let the horse participate in a race every day.

d) Show that it is (average) optimal to only let the horse participate if it is fit.

Animal protection regulation does not allow the horse to participate in more than 50% of the races.

e) Describe how an optimal strategy can be found under this restriction. (We are interested in the method*; the optimal strategy itself need not be determined.)*

3.5 Linear Programming Approach

We finally discuss the linear programming (LP) approach for discounted rewards in case $|\mathcal{I}| < \infty$, i.e., the state space is finite. Suppose X_0 follows some initial distribution $\mathbb{P}\{X_0 = i\} = p_0(i) > 0$. The requirement $p_0(i) > 0$ for all i is to ensure that we eventually visit all states. (Until now we simply considered the process starting from each state separately; in the LP formulation we need to ensure that the rewards starting in each state matter.) Suppose we are given a fixed (possibly non-stationary) strategy $\mathbf{s}$. Let

$$p_n^{\mathbf{s},a}(i) := \mathbb{P}^{\mathbf{s},p_0}\{X_n = i, A_n = a\},$$

where the superscript p_0 denotes that this probability depends on the choice of $p_0(\cdot)$. Clearly,

$$p_n^{\mathbf{s}}(i) := \mathbb{P}^{\mathbf{s},p_0}\{X_n = i\} = \sum_{a \in \mathcal{A}_i} p_n^{\mathbf{s},a}(i).$$

We may also write the Chapman-Kolmogorov type equations

$$p_{n+1}^{\mathbf{s}}(j) = \sum_{i \in \mathcal{I}} \sum_{a \in \mathcal{A}_i} p_n^{\mathbf{s},a}(i) p^a(i,j).$$

This all brings us to the formulation of the following LP:

$$\begin{aligned} &\max_{\mathbf{s}} \sum_{n=0}^{\infty} \alpha^n \sum_{i \in \mathcal{I}} \sum_{a \in \mathcal{A}_i} p_n^{\mathbf{s},a}(i) r^a(i) \\ &\text{subject to} \\ &\sum_{a \in \mathcal{A}_i} p_0^{\mathbf{s},a}(i) = p_0(i), \qquad i \in \mathcal{I}, \end{aligned} \tag{3.11}$$

$$\sum_{a\in\mathcal{A}_j} p_{n+1}^{\mathbf{s},a}(j) = \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{A}_i} p_n^{\mathbf{s},k}(i)p^k(i,j), \qquad j\in\mathcal{I},\, n=0,1,2,\ldots,$$
$$p_n^{\mathbf{s},a}(i) \geq 0, \qquad i\in\mathcal{I},\, a\in\mathcal{A}_i,\, n=0,1,2,\ldots.$$

Instead of maximizing over all $\mathbf{s}$ we may as well maximize over the variables $p_n^{\mathbf{s},a}(i)$. This is clear, since we can identify a strategy with any realization of the $p_n^{\mathbf{s},a}(i)$:

$$s_n^a(i) := \frac{p_n^{\mathbf{s},a}(i)}{\sum_{k\in\mathcal{A}_i} p_n^{\mathbf{s},k}(i)},$$

if $\sum_{k\in\mathcal{A}_i} p_n^{\mathbf{s},k}(i) > 0$, otherwise the choice of $s_n^a(i)$ does not matter (the process can not reach state i at time n).

A severe problem with (3.11) is that the number of decision variables is infinite, even if the state space is finite. This can be circumvented by defining

$$x^{\mathbf{s},a}(i) := \sum_{n=0}^{\infty} \alpha^n p_n^{\mathbf{s},a}(i),$$

which can be interpreted as the "discounted" number of visits to state i which are followed by action a. From (3.11) we obtain

$$\max \sum_{i\in\mathcal{I}}\sum_{a\in\mathcal{A}_i} x^{\mathbf{s},a}(i)r^a(i)$$
subject to
$$\sum_{a\in\mathcal{A}_j} x^{\mathbf{s},a}(j) = p_0(j) + \alpha\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{A}_i} x^{\mathbf{s},k}(i)p^k(i,j), \qquad j\in\mathcal{I}, \quad (3.12)$$
$$x^{\mathbf{s},a}(i) \geq 0, \qquad i\in\mathcal{I},\, a\in\mathcal{A}_i.$$

Since the optimum is attained for at least 1 basic solution it is sufficient that we concentrate on these. Note that the number of non-zero variables in a basic solution can not be more than $|\mathcal{I}|$, which is the number of restrictions in (3.12). Since we required that $p_0(i) > 0$ for all i, it must be that for each i we have $x^{\mathbf{s},a}(i) > 0$ for exactly one value $a\in\mathcal{A}_i$ and $x^{\mathbf{s},k}(i) = 0$ for $k\neq a$. It may be verified that choosing $\mathbf{s} = \mathbf{f} = (f,f,f,\ldots)$, the stationary decision rule with $f(i) = a$ if $x^{\mathbf{s},a}(i) > 0$, the maximum discounted rewards for each initial state can be attained.

Again, the LP formulation (3.12) can be modified to incorporate restrictions on the allowed strategies. The program will then render an optimal stationary randomized strategy (if it exists) under this additional restriction. We emphasize again that neither PI nor SA can cope with such restrictions. Moreover, it may be so that no stationary decision rule attains the same maximum discounted reward as the optimal randomized strategy.

Example 3.5.1 Two-state MDP with discounted costs *(See Section 3.7 for solutions)*
Consider again the problem described in Example 2.6.1, but now with discounted costs (with discount factor $\alpha = \frac{1}{2}$ per period.)

a) Carry out two steps of the successive approximation algorithm. In each step, determine the corresponding strategy.

b) Show that it is optimal to take action 2 in state 0 and action 1 in state 1. Also determine the minimal discounted cost.

c) State the corresponding Linear Programming formulation for this MDP with discounted cost. (The linear program need not be solved.)

d) How can we distill the optimal solution from the solution of the linear program?

3.6 Relating Average and Discounted Rewards

We finally state the following theorem that relates average rewards and corresponding relative values with discounted rewards for $\alpha \to 1$.

Theorem 3.6.1 *Suppose that for the stationary decision rule* $\mathbf{f}$ *the average reward* $g^{\mathbf{f}}$ *and a corresponding relative reward function* $d^{\mathbf{f}}(i)$, $i \in \mathcal{I}$ *are well defined (see Chapter 2). We then have:*

$$g^{\mathbf{f}} = \lim_{\alpha \to 1} (1-\alpha) V_\alpha^{\mathbf{f}}(i), \qquad i \in \mathcal{I}, \tag{3.13}$$

$$d^{\mathbf{f}}(i) - d^{\mathbf{f}}(j) = \lim_{\alpha \to 1} \left(V_\alpha^{\mathbf{f}}(i) - V_\alpha^{\mathbf{f}}(j) \right). \tag{3.14}$$

Proof Discussed at the lectures. □

Example 3.6.1 Problem 2: Bacteria farm *(See Section 3.7 for solutions)*
Consider a scientific bacteria farm. Each day a sample worth € 2000 can be drawn from a healthy bacteria population. Due to the absence of sun light, an epidemic reaction may take place during the night, infecting the entire population (instantly). Such a reaction occurs in a healthy population with probability $\frac{1}{3}$. With probability $\frac{2}{3}$, the population is still healthy the next day. An infected population remains infected for ever. Still, from an infected population, a sample worth € 1.000 can be drawn every day. At the beginning of the day, the bacteria reservoir is inspected. If the population is infected, it can be replaced with a new (healthy) population. The replacement cost is € 1.000 (that must be payed immediately when the decision to replace the population is made). If a population is replaced, no sample can be taken from the old one, that must be thrown away immediately so that the reservoir can be cleaned. The new population is available the next day (and is always healthy). The goal is to maximize the total discounted rewards. The discount factor is $\alpha \in (0, 1)$ per day.

This problem can be modeled as a Markov Decision Problem as follows. There are two possible states at the beginning of the day: state 0 (the population is

infected) and state 1 (the population is healthy). There are also two possible actions: action 0 (the population is not replaced) and action 1 (the population is replaced). The direct rewards are: $r^0(0) = 1$, $r^0(1) = 2$, $r^1(0) = r^1(1) = -1$. The transition probabilities are given by $p^0(0,0) = p^1(0,1) = p^1(1,1) = 1$, $p^0(1,0) = 1 - p^0(1,1) = \frac{1}{3}$.

a) *Carry out two steps of the* successive approximation *algorithm (choose the null function for initialization) Show that, starting with a healthy population, the maximum discounted rewards are between* $2+\frac{5}{3}\alpha+\frac{\alpha^2}{1-\alpha}$ *and* $2+\frac{5}{3}\alpha+\frac{5\alpha^2}{3(1-\alpha)}$. *(Use the bounds corresponding to the second iteration.)*

Consider the stationary decision rule f that replaces the bacteria population whenever it is infected, and never replaces the population when it is healthy.

b) *Verify that, starting with a healthy population, the discounted rewards for this strategy equal* $\frac{2-\frac{1}{3}\alpha}{(1-\alpha)(1+\frac{1}{3}\alpha)}$. *(This may be done by substituting this value.) Also determine the discounted rewards when starting with an infected population.*

After a more thorough investigation of this problem, it turns out that strategy f renders the maximum discounted rewards for all $\alpha \in (\frac{6}{7}, 1)$.

c) *Determine the average rewards rendered by* f *as well as the corresponding relative values, by using the expressions for the relative value function found in part b. (If part b has not been answered, you may determine the average rewards and the relative rewards using a different approach.)*

3.7 Solutions to selected exercises

Example 3.4.1 Selling a house

a) State on the n-th day: $X_n =$ current offer $\in \mathcal{I} = \{0, 1, 2, \ldots\} \cup \{\infty\}$.
$\mathcal{A}_i = \{0, 1\} =$ {reject,accept} for $i < \infty$, $\mathcal{A}_\infty = \{\bullet\}$.
$r^0(i) = -C$, $\forall i < \infty$; $r^1(i) = i$, $\forall i < \infty$; $r^\bullet(\infty) = 0$.
$p^0(i,j) = p(j)$, $\forall i, j < \infty$; $p^1(i,\infty) = 1$, $\forall i < \infty$; $p^\bullet(\infty,\infty) = 1$.

b) Optimality equation:

$$V_\alpha^*(i) = \max\{-C + \alpha \sum_{j=0}^{\infty} p(j) V_\alpha^*(j),\ i + \alpha V_\alpha^*(\infty)\}.$$

Since the first entry is independent of i we have that action 0 is optimal if $i < -C + \alpha \sum_{j=0}^{\infty} p(j) V_\alpha^*(j) - \alpha V_\alpha^*(\infty)$ and action 1 is optimal if $i > -C + \alpha \sum_{j=0}^{\infty} p(j) V_\alpha^*(j) - \alpha V_\alpha^*(\infty)$.

c) Set $f_0(i) = 0$ if $i \leq i_0$ and $f_0(i) = 1$ if $i > i_0$. Then, for $i \leq i_0$,

$$V_\alpha^{f_0}(i) = -C + \alpha \sum_{j=0}^{\infty} p(j) V_\alpha^{f_0}(j),$$

and the right hand side is independent of i so that $V_\alpha^{f_0}(i) = V_\alpha^{f_0}(0)$. Furthermore note that

$$V_\alpha^{f_0}(\infty) = 0 + \alpha V_\alpha^{f_0}(\infty),$$

So that $V_\alpha^{f_0}(\infty) = 0$ and, for $i > i_0$,

$$V_\alpha^{f_0}(i) = i + \alpha V_\alpha^{f_0}(\infty) = i.$$

Applying the policy iteration step once, we have

$$f_1(i) = \arg\max\{-C + \alpha \sum_{j=0}^{\infty} p(j) V_\alpha^{f_0},\, i\},$$

and, since the first entry is again independent of i, we have that f_1 is a threshold strategy with threshold $i_1 = \lfloor -C + \alpha \sum_{j=0}^{\infty} p(j) V_\alpha^{f_0} \rfloor$.

d) First note that, for $n \geq 1$, $V_\alpha^{f_n}(i_n) \geq i_n$ because $f_n(i)$ results from the maximization in the improvement step (this is not true for $n = 0$). If $i_{n+1} < i_n$ then the optimal action in state i_n changed in step $n+1$ of the policy iteration algorithm (from 0 to 1). This is only possible if $V_\alpha^{f_{n+1}}(i_n) > V_\alpha^{f_n}(i_n)$ (strict monotonicity). Note also that $V_\alpha^{f_{n+1}}(i_n) = i_n$. This implies that $i_n = V_\alpha^{f_{n+1}}(i_n) > V_\alpha^{f_n}(i_n) \geq i_n$, which is a contradiction. So it must be that $i_{n+1} \geq i_n$.

Example 3.5.1 Two-state MDP with discounted costs

a) Take $v_0(0) = v_0(1) = 0$ (other choices are possible, but this choice simplifies the first step).

$v_1(0) = \min\{1, 0\} = 0,\ f_1(0) = 2;$
$v_1(1) = \min\{2, 2\} = 2,\ f_1(1) \in \{1, 2\};$

$v_2(0) = \min\left\{1 + \frac{1}{2}\left(\frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 2\right), 0 + \frac{1}{2}\left(\frac{1}{4} \cdot 0 + \frac{3}{4} \cdot 2\right)\right\} = \frac{3}{4},\ f_2(0) = 2;$
$v_2(1) = \min\left\{2 + \frac{1}{2}\left(\frac{2}{3} \cdot 0 + \frac{1}{3} \cdot 2\right), 2 + \frac{1}{2}\left(\frac{1}{3} \cdot 0 + \frac{2}{3} \cdot 2\right)\right\} = 2\frac{1}{3},\ f_2(1) = 1.$

b) One step of policy iteration with $f = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, $r^f = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$, $P^f = \begin{pmatrix} \frac{1}{4} & \frac{3}{4} \\ \frac{2}{3} & \frac{1}{3} \end{pmatrix}$.

$$\begin{aligned} V_\alpha^f(0) &= 0 + \alpha\left(\frac{1}{4}V_\alpha^f(0) + \frac{3}{4}V_\alpha^f(1)\right) \\ V_\alpha^f(1) &= 2 + \alpha\left(\frac{2}{3}V_\alpha^f(0) + \frac{1}{3}V_\alpha^f(1)\right) \end{aligned}$$

Solving (with $\alpha = \frac{1}{2}$) gives: $V_\alpha^f(0) = \frac{36}{29}$ en $V_\alpha^f(1) = \frac{84}{29}$.
$f'(0) \in \text{argmin}\left\{1 + \frac{1}{2}\left(\frac{1}{2}\cdot\frac{36}{29} + \frac{1}{2}\cdot\frac{84}{29}\right), \frac{36}{29}\right\} = \{2\}$,
$f'(1) \in \text{argmin}\left\{\frac{84}{29}, 2 + \frac{1}{2}\left(\frac{1}{3}\cdot\frac{36}{29} + \frac{2}{3}\cdot\frac{84}{29}\right)\right\} = \{1\}$.
$f' = f$ and so this strategy is optimal.
$V^*_{\alpha=\frac{1}{2}} = \begin{pmatrix} \frac{36}{29} \\ \frac{84}{29} \end{pmatrix}$.

c)

$$\min_{x(i,a)} x(0,1) + 2x(1,1) + 2x(1,2)$$

subject to

$$x(0,1) + x(0,2) = p_0(0) + \frac{1}{2}\left(\frac{1}{2}x(0,1) + \frac{1}{4}x(0,2) + \frac{2}{3}x(1,1) + \frac{1}{3}x(1,2)\right),$$

$$x(1,1) + x(1,2) = p_0(1) + \frac{1}{2}\left(\frac{1}{2}x(0,1) + \frac{3}{4}x(0,2) + \frac{1}{3}x(1,1) + \frac{2}{3}x(1,2)\right),$$

$$x(i,a) \geq 0.$$

Here $x(i,a)$ is the "discounted number" of visits to state i folllowed by action a. The initialisation probabilities $p_0(i)$ may be chosen arbitrarily, as long as $p_0(i) > 0$ for $i = 0, 1$.

d) In general: $s^{*,a}(i) = \frac{x(i,a)}{\sum_{k\in\mathcal{A}_i} x(i,k)}$, but since the optimum is also attained in at least one basic solution of the linear program, we can also find an optima decision rule: $f^*(i) = a$ if $x(i,a) > 0$ (and if $x(i,a) > 0$ for more than one action a, an arbitrary choice may be made among these actions.).

Example 3.6.1 Bacteria farm

a) (all rewards and costs are divided by 1000)

$$v_0(0) = v_0(1) = 0$$

$$v_1(0) = \max\{1 + \alpha v_0(0), -1 + \alpha v_0(1)\} = 1$$
$$v_1(1) = \max\left\{2 + \tfrac{1}{3}\alpha v_0(0) + \tfrac{2}{3}\alpha v_0(1), -1 + \alpha v_0(1)\right\} = 2$$

$v_2(0) = \max\{1+\alpha, -1+2\alpha\} = 1+\alpha$ (want $\alpha \leq 1$)
$v_2(1) = \max\left\{2+\frac{1}{3}\alpha+\frac{4}{3}\alpha, -1+2\alpha\right\} = 2+\frac{5}{3}\alpha$

$m_2 = \min\{v_2(0)-v_1(0), v_2(1)-v_1(1)\} = \alpha,$
$M_2 = \max\{v_2(0)-v_1(0), v_2(1)-v_1(1)\} = \frac{5}{3}\alpha,$

$$\Rightarrow 2+\frac{5}{3}\alpha+\frac{\alpha}{1-\alpha}\alpha \leq V_\alpha^*(1) \leq 2+\frac{5}{3}\alpha+\frac{\alpha}{1-\alpha}\cdot\frac{5}{3}\alpha$$

b) (1) $V_\alpha^f(0) = -1+\alpha V_\alpha^f(1)$
(2) $V_\alpha^f(1) = 2+\frac{1}{3}\alpha V_\alpha^f(0)+\frac{2}{3}\alpha V_\alpha^f(1)$
Vul (1) in (2): $\Rightarrow \left(1-\frac{1}{3}\alpha^2-\frac{2}{3}\alpha\right)V_\alpha^f(1) = 2-\frac{1}{3}\alpha$
$\Rightarrow V_\alpha^f(0) = -1+\frac{\alpha(2-\frac{1}{3}\alpha)}{(1-\alpha)(1+\frac{1}{3}\alpha)}.$

c)

$$g^f = \lim_{\alpha\to 1}(1-\alpha)V_\alpha^f(0) = \frac{5}{4}$$

$$d^f(1)-d^f(0) = \lim_{\alpha\to 1}\left(V_\alpha^f(1)-V_\alpha^f(0)\right) = \frac{9}{4}$$

Appendix A

Proofs

A.1 Theorem 1.3.1

Proof The proof is by induction on n. Clearly, since there's no decision to take when the planning horizon is left, we have for all possible strategies $V_0^{\mathbf{s}}(i) = V_0^*(i)$, $i \in \mathcal{I}$. Let $\mathbf{s}_n = (s_n, s_{n-1}, \ldots, s_0)$ be an arbitrary strategy over n periods, and let $V_n^{\mathbf{s}}(h, i)$ be the expected reward when using $\mathbf{s}_n$, starting from state i with history h. Assume that (this is the induction step; it is certainly satisfied for $n = 0$)

$$V_n^*(i) \geq V_n^{\mathbf{s}}(h, i),$$

uniformly for all possible histories h. Then,

$$\begin{aligned}
V_{n+1}^{\mathbf{s}}(h,i) &= \sum_{a \in \mathcal{A}_i} s_{n+1}^a(h,i) \left(r^a(i) + \sum_{j \in \mathcal{I}} p^a(i,j) V_n^{\mathbf{s}}((h,i,a),j) \right) \\
&\overset{\text{induction!}}{\leq} \sum_{a \in \mathcal{A}_i} s_{n+1}^a(h,i) \left(r^a(i) + \sum_{j \in \mathcal{I}} p^a(i,j) V_n^*(j) \right) \\
&\leq \max_{a \in \mathcal{A}_i} \left\{ r^a(i) + \sum_{j \in \mathcal{I}} p^a(i,j) V_n^*(j) \right\} \\
&= V_{n+1}^*(i).
\end{aligned}$$

The proof is completed by noting that $\mathbf{f}_n^*$ attains the (maximum) expected rewards V_n^* for all n. □

A.2 Theorem 2.3.1

Proof In the proof, we first show that $\limsup_{n\to\infty} \frac{1}{n} V_n^s(i) \leq g$ and then that this upper bound can actually be attained. We start with an upper bound for the

expected direct improvement of the function $d(\cdot)$ in one step. Note that for any strategy s

$$\begin{aligned}\mathbf{E}^s[d(X_{t+1})|X_t=i] &= \sum_a s^a(i)\left(\sum_j p^a(i,j)d(j)+r^a(i)-r^a(i)\right)\\ &\leq \max_a\left(\sum_j p^a(i,j)d(j)+r^a(i)\right)-\sum_a s^a(i)r^a(i)\\ &= d(i)+g-(\mathbf{E}^s[r^{A_t}(X_t)|X_t=i],\end{aligned}$$

for all $i\in\mathcal{I}$. Thus, the expected direct one-step improvement can not be more than $g-(\mathbf{E}^s[r^{A_t}(X_t)|X_t=i]$. By conditioning on the state at time t we can also bound the one-step improvement in the $(t+1)$-st step:

$$\begin{aligned}\mathbf{E}^s[d(X_{t+1})|X_0=i] &= \sum_j p_t^s(i,j)\mathbf{E}^s[d(X_{t+1})|X_t=j]\\ &\leq \sum_j p_t^s(i,j)\left(d(j)+g-\mathbf{E}^s[r^{A_t}(X_t)|X_t=j]\right)\\ &= g+\mathbf{E}^s[d(X_t)|X_0=i]-\mathbf{E}^s[r^{A_t}(X_t)|X_0=i],\end{aligned}$$

for all $i\in\mathcal{I}$. Adding the previous equation over $t=0$ up to $n-1$ and canceling identical terms we get

$$\mathbf{E}^s[d(X_n)|X_0=i] \leq n\,g+\mathbf{E}^s[d(X_0)|X_0=i]-V_n^s(i),$$

so that, since $d(\cdot)$ is bounded,

$$\limsup_{n\to\infty}\frac{1}{n}V_n^s(i)\leq g.$$

The proof is completed by noting that all above inequalities may be replaced by equality signs when a stationary decision rule $\mathbf{f}=(f,f,f,\ldots)$ that satisfies

$$f(i)\in \text{argmax}_{a\in\mathcal{A}_i}\left\{r^a(i)+\sum_{j\in\mathcal{I}}p^a(i,j)\,d(j)\right\},$$

is used. □

A.3 Theorem 2.4.1

Proof The fact that $g=g^{\mathbf{f}}$ follows from Theorem 2.3.1: By considering the same Markov decision process with the restriction that the action space in state i only consists of $f(i)$, for all $i\in\mathcal{I}$; since there's only one strategy allowed, it is also the

optimal strategy for this modified process.
To complete the proof, substitute (2.16) m times into itself to obtain

$$d_0^{\mathbf{f}}(i) = V_m^{\mathbf{f}}(i) - mg + \sum_{j\in\mathcal{I}} p_m^{\mathbf{f}}(i,j) d_0^{\mathbf{f}}(j).$$

From (2.17) we similarly have

$$d(i) = V_m^{\mathbf{f}}(i) - mg + \sum_{j\in\mathcal{I}} p_m^{\mathbf{f}}(i,j) d(j).$$

Subtracting these two equations gives

$$d_0^{\mathbf{f}}(i) - d(i) = \sum_{j\in\mathcal{I}} p_m^{\mathbf{f}}(i,j) \left(d_0^{\mathbf{f}}(j) - d(j)\right). \tag{A.1}$$

If $p_m^{\mathbf{f}}(i,j) \to \pi^{\mathbf{f}}(j)$, as $m \to \infty$, we therefore immediately obtain

$$d_0^{\mathbf{f}}(i) - d(i) = \sum_{j\in\mathcal{I}} \pi^{\mathbf{f}}(j) \left(d_0^{\mathbf{f}}(j) - d(j)\right), \tag{A.2}$$

which is independent of i and, thus, the theorem is proved. (The interchange of the limit and the summation is allowed because $d^{\mathbf{f}}$ and d are bounded functions; use the Dominated Convergence Theorem.)

If the $p_m^{\mathbf{f}}(i,j)$ do not converge we can still use the Césaro limit $\frac{1}{n}\sum_{n=1}^{m} p^{\mathbf{f}}(i,j) \to \pi^{\mathbf{f}}(j)$, as $n \to \infty$ to arrive at (A.2), by adding (A.1) over $m = 1, 2, \ldots, n$, dividing by n and letting $n \to \infty$ (again using dominated convergence). □

A.4 Theorem 2.5.1

Proof We start from the standard equations for the relative values:

$$\begin{aligned} d_0^{\mathbf{f}}(i) + g^{\mathbf{f}} &= r^{\mathbf{f}}(i) + \sum_{j\in\mathcal{I}} p^{\mathbf{f}}(i,j) d_0^{\mathbf{f}}(j) \\ &\leq r^{\mathbf{f}'}(i) + \sum_{j\in\mathcal{I}} p^{\mathbf{f}'}(i,j) d_0^{\mathbf{f}}(j). \end{aligned}$$

(The inequality is a consequence of the policy improvement step.) Using $\mathbf{f}'$, the expected reward at time t (starting in i_0) is

$$\begin{aligned} \sum_{i\in\mathcal{I}} p_t^{\mathbf{f}'}(i,i_0) r^{\mathbf{f}'}(i) &\geq \sum_{i\in\mathcal{I}} p_t^{\mathbf{f}'}(i,i_0) \left[d_0^{\mathbf{f}}(i) + g^{\mathbf{f}} - \sum_{j\in\mathcal{I}} p^{\mathbf{f}'}(i,j) d_0^{\mathbf{f}}(j) \right] \\ &= g^{\mathbf{f}} + \sum_{i\in\mathcal{I}} p_t^{\mathbf{f}'}(i,i_0) d_0^{\mathbf{f}}(i) - \sum_{j\in\mathcal{I}} p_{t+1}^{\mathbf{f}'}(i_0,j) d_0^{\mathbf{f}}(j). \end{aligned} \tag{A.3}$$

If $\pi^{\mathbf{f}'}(i) = \lim_{t\to\infty} p_t^{\mathbf{f}'}(i, i_0)$ exists, then we can take $t \to \infty$ and use the boundedness of $r^{\mathbf{f}}$ and $d^{\mathbf{f}}$ to interchange limit and summation:

$$\sum_{i\in\mathcal{I}} \pi^{\mathbf{f}'}(i) r^{\mathbf{f}'}(i) \quad \geq \quad g^{\mathbf{f}} + \sum_{i\in\mathcal{I}} \pi^{\mathbf{f}'}(i) d_0^{\mathbf{f}}(i) - \sum_{j\in\mathcal{I}} \pi^{\mathbf{f}'}(j) d_0^{\mathbf{f}}(j),$$

or, equivalently,

$$g^{\mathbf{f}'} \geq g^{\mathbf{f}}.$$

If $\lim_{t\to\infty} p_t^{\mathbf{f}'}(i, i_0)$ does not exist, then we can add (A.3) over $t = 0, \ldots, T$, divide by T, and let $T \to \infty$ to arrive at the same conclusion. □

A.5 Theorem 2.5.3

Because of Theorem 2.5.1, it is sufficient to show that $g^{\mathbf{f}'} = g^{\mathbf{f}}$ implies (ii). We therefore assume that $g^{\mathbf{f}'} = g^{\mathbf{f}}$. The proof is given in three steps.

(a) The recurrent sets of $\mathbf{f}$ and $\mathbf{f}'$ coincide and $\pi^{\mathbf{f}}(i) = \pi^{\mathbf{f}'}(i)$ for all $i \in \mathcal{R}$, where $\mathcal{R}$ is the set of recurrent states;

(b) $d^{\mathbf{f}'}(i, i_0)$ is well defined (with the same reference state i_0) and equal to $d^{\mathbf{f}}(i, i_0)$ for $i \in \mathcal{R}$;

(c) $d^{\mathbf{f}}(i, i_0) < d^{\mathbf{f}'}(i, i_0)$ for all $i \notin \mathcal{R}$.

For the first step, note that

$$r^{\mathbf{f}'}(i) + \sum_{j\in\mathcal{I}} p^{\mathbf{f}'}(i, j) d^{\mathbf{f}}(j) \geq d^{\mathbf{f}}(i) + g^{\mathbf{f}},$$

with strict inequality iff $f'(i) \neq f(i)$. Multiplying this inequality by $\pi^{\mathbf{f}'}(i)$ and summing over all $i \in \mathcal{I}$ gives

$$g^{\mathbf{f}'} + \sum_{j\in\mathcal{I}} \pi^{\mathbf{f}'}(j) d^{\mathbf{f}}(j) \geq g^{\mathbf{f}} + \sum_{i\in\mathcal{I}} \pi^{\mathbf{f}'}(i) d^{\mathbf{f}}(i),$$

with strict inequality iff for some $i \in \mathcal{I}$ *both* $f'(i) \neq f(i)$ *and* $\pi^{\mathbf{f}'}(i) > 0$. Since we assumed that $g^{\mathbf{f}} = g^{\mathbf{f}'}$, it must be that $f'(i) = f(i)$ for all i with $\pi^{f'(i)} > 0$. Since both $\mathbf{f}$ and $\mathbf{f}'$ are assumed to have a single recurrent set, they must coincide, because on the recurrent set the two strategies prescribe the same actions. This also implies that the equilibrium distributions are the same.

Step 2. Since the two chains have the same recurrent states, we may choose $i_0 \in \mathcal{R}$ as the reference state for $\mathbf{f}'$ too. Furthermore, for all $i \in \mathcal{R}$, $d^{\mathbf{f}}(i, i_0)$ and $d^{\mathbf{f}}(i, i_0)$ both satisfy

$$\begin{aligned} d(i) + g &= r^f(i) + \sum_{j\in\mathcal{I}} p^{\mathbf{f}}(i, j) d(j) \\ &= r^f(i) + \sum_{j\in\mathcal{R}} p^{\mathbf{f}}(i, j) d(j), \end{aligned}$$

because $p^{\mathbf{f}}(i,j) = 0$ if $j \notin \mathcal{R}$ and, again, because on $\mathcal{R}$ the two strategies prescribe the same actions.

Bibliography

[1] C. Derman. *Finite State Markovian Decision Processes.* Academic Press, New York, 1970.

[2] R.A. Howard. *Dynamic Programming and Markov Processes.* Wiley, New York, 1960.

[3] M. Putterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, New York, 1994.

[4] S.M. Ross. *Applied Probability Models with Optimization Applications.* Holden Day, San Francisco, 1970; republished by Dover, Mineola (NY), 1992.

[5] H.C. Tijms. *Stochastic Models – An Algorithmic Approach.* Wiley, New York, 1994.