1 Dynamic Programming: The Optimality Equation

We introduce the idea of dynamic programming and the principle of optimality. We give notation for state-structured models, and introduce ideas of feedback, open-loop, and closed-loop controls, a Markov decision process, and the idea that it can be useful to model things in terms of time to go.

1.1 Control as optimization over time

Optimization is a key tool in modelling. Sometimes it is important to solve a problem optimally. Other times either a near-optimal solution is good enough, or the real problem does not have a single criterion by which a solution can be judged. However, even then optimization is useful as a way to test thinking. If the 'optimal' solution is ridiculous it may suggest ways in which both modelling and thinking can be refined.

Control theory is concerned with dynamic systems and their **optimization over time**. It accounts for the fact that a dynamic system may evolve stochastically and that key variables may be unknown or imperfectly observed (as we see, for instance, in the UK economy).

This contrasts with optimization models in the IB course (such as those for LP and network flow models); these static and nothing was random or hidden. It is these three new features: dynamic and stochastic evolution, and imperfect state observation, that give rise to new types of optimization problem and which require new ways of thinking.

We could spend an entire lecture discussing the importance of control theory and tracing its development through the windmill, steam governor, and so on. Such 'classic control theory' is largely concerned with the question of stability, and there is much of this theory which we ignore, e.g., Nyquist criterion and dynamic lags.

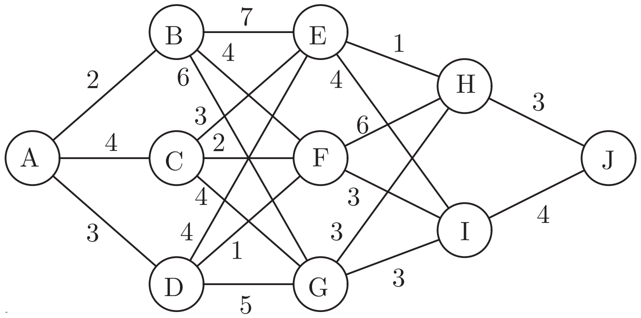
1.2 The principle of optimality

A key idea is that optimization over time can often be regarded as 'optimization in stages'. We trade off our desire to obtain the lowest possible cost at the present stage against the implication this would have for costs at future stages. The best action minimizes the sum of the cost incurred at the current stage and the least total cost that can be incurred from all subsequent stages, consequent on this decision. This is known as the Principle of Optimality.

Definition 1.1 (Principle of Optimality) *From any point on an optimal trajectory, the remaining trajectory is optimal for the corresponding problem initiated at that point.*

1.3 Example: the shortest path problem

Consider the 'stagecoach problem' in which a traveler wishes to minimize the length of a journey from town A to town J by first traveling to one of B, C or D and then onwards to one of E, F or G then onwards to one of H or I and the finally to J. Thus there are 4 'stages'. The arcs are marked with distances between towns.



Road system for stagecoach problem

Solution. Let $F(\mathrm{X})$ be the minimal distance required to reach J from X. Then clearly, $F(\mathrm{J}) = 0$, $F(\mathrm{H}) = 3$ and $F(\mathrm{I}) = 4$.

$$F(\mathrm{F}) = \min[\,6 + F(\mathrm{H}), 3 + F(\mathrm{I})\,] = 7\,,$$

and so on. Recursively, we obtain $F(\mathrm{A}) = 11$ and simultaneously an optimal route, i.e., A→D→F→I→J (although it is not unique). ■

The study of dynamic programming dates from Richard Bellman, who wrote the first book on the subject (1957) and gave it its name. A very large number of problems can be treated this way.

1.4 The optimality equation

The optimality equation in the general case. In **discrete-time** t takes integer values, say $t = 0, 1, \ldots$. Suppose u_t is a **control variable** whose value is to be chosen at time t. Let $U_{t-1} = (u_0, \ldots, u_{t-1})$ denote the partial sequence of controls (or decisions) taken over the first t stages. Suppose the cost up to the **time horizon** h is given by

$$\mathbf{C} = G(U_{h-1}) = G(u_0, u_1, \ldots, u_{h-1})\,.$$

Then the **principle of optimality** is expressed in the following theorem.

Theorem 1.2 (The principle of optimality) *Define the functions*

$$G(U_{t-1}, t) = \inf_{u_t, u_{t+1}, \ldots, u_{h-1}} G(U_{h-1})\,.$$

Then these obey the recursion

$$G(U_{t-1}, t) = \inf_{u_t} G(U_t, t+1) \quad t < h\,,$$

with terminal evaluation $G(U_{h-1}, h) = G(U_{h-1})$.

The proof is immediate from the definition of $G(U_{t-1}, t)$, i.e.,

$$G(U_{t-1}, t) = \inf_{u_t} \inf_{u_{t+1}, \ldots, u_{h-1}} G(u_0, \ldots, u_{t-1},\ u_t\ , u_{t+1}, \ldots, u_{h-1})\,.$$

The state structured case. The control variable u_t is chosen on the basis of knowing $U_{t-1} = (u_0, \ldots, u_{t-1})$, (which determines everything else). But a more economical representation of the past history is often sufficient. For example, we may not need to know the entire path that has been followed up to time t, but only the place to which it has taken us. The idea of a **state variable** $x \in \mathbb{R}^d$ is that its value at t, denoted x_t, is calculable from known quantities and obeys a **plant equation** (or law of motion)

$$x_{t+1} = a(x_t, u_t, t) \,.$$

Suppose we wish to minimize a cost function of the form

$$\mathbf{C} = \sum_{t=0}^{h-1} c(x_t, u_t, t) + \mathbf{C}_h(x_h) \,, \tag{1.1}$$

by choice of controls $\{u_0, \ldots, u_{h-1}\}$. Define the cost from time t onwards as,

$$\mathbf{C}_t = \sum_{\tau=t}^{h-1} c(x_\tau, u_\tau, \tau) + \mathbf{C}_h(x_h) \,, \tag{1.2}$$

and the minimal cost from time t onwards as an optimization over $\{u_t, \ldots, u_{h-1}\}$ conditional on $x_t = x$,

$$F(x, t) = \inf_{u_t, \ldots, u_{h-1}} \mathbf{C}_t \,.$$

Here $F(x, t)$ is the minimal future cost from time t onward, given that the state is x at time t. Then by an inductive proof, one can show as in Theorem 1.2 that

$$F(x, t) = \inf_u [c(x, u, t) + F(a(x, u, t), t+1)] \,, \quad t < h \,, \tag{1.3}$$

with terminal condition $F(x, h) = \mathbf{C}_h(x)$. Here x is a generic value of x_t. The minimizing u in (1.3) is the optimal control $u(x, t)$ and values of $x_0, \ldots, x_{t-1}$ are irrelevant. The **optimality equation** (1.3) is also called the **dynamic programming equation** (DP) or **Bellman equation**.

The DP equation defines an optimal control problem in what is called **feedback** or **closed loop** form, with $u_t = u(x_t, t)$. This is in contrast to the **open loop** formulation in which $\{u_0, \ldots, u_{h-1}\}$ are to be determined all at once at time 0. A **policy** (or strategy) is a rule for choosing the value of the control variable under all possible circumstances as a function of the perceived circumstances. To summarise:

(i) The optimal u_t is a function only of x_t and t, i.e, $u_t = u(x_t, t)$.

(ii) The DP equation expresses the optimal u_t in closed loop form. It is optimal whatever the past control policy may have been.

(iii) The DP equation is a backward recursion in time (from which we get the optimum at $h-1$, then $h-2$ and so on.) The later policy is decided first.

‘*Life must be lived forward and understood backwards.*’ (Kierkegaard)

1.5 Markov decision processes

Consider now stochastic evolution. Let $X_t = (x_0, \ldots, x_t)$ and $U_t = (u_0, \ldots, u_t)$ denote the x and u histories at time t. As above, state structure is characterised by the fact that the evolution of the process is described by a state variable x, having value x_t at time t, with the following properties.

(a) *Markov dynamics:* (i.e., the stochastic version of the plant equation.)

$$P(x_{t+1} \mid X_t, U_t) = P(x_{t+1} \mid x_t, u_t) \,.$$

(b) *Decomposable cost*, (i.e., cost given by (1.1)).

These assumptions define state structure. For the moment we also require.

(c) *Perfect state observation:* The current value of the state is observable. That is, x_t is known at the time at which u_t must be chosen. So, letting W_t denote the observed history at time t, we assume $W_t = (X_t, U_{t-1})$. Note that $\mathbf{C}$ is determined by W_h, so we might write $\mathbf{C} = \mathbf{C}(W_h)$.

These assumptions define what is known as a discrete-time **Markov decision process** (MDP). Many of our examples will be of this type. As above, the cost from time t onwards is given by (1.2). Denote the minimal expected cost from time t onwards by

$$F(W_t) = \inf_\pi E_\pi[\mathbf{C}_t \mid W_t] \,,$$

where π denotes a policy, i.e., a rule for choosing the controls $u_0, \ldots, u_{h-1}$. We can assert the following theorem.

Theorem 1.3 *$F(W_t)$ is a function of x_t and t alone, say $F(x_t, t)$. It obeys the optimality equation*

$$F(x_t, t) = \inf_{u_t} \{c(x_t, u_t, t) + E[F(x_{t+1}, t+1) \mid x_t, u_t]\} \,, \quad t < h \,, \tag{1.4}$$

with terminal condition

$$F(x_h, h) = \mathbf{C}_h(x_h) \,.$$

Moreover, a minimizing value of u_t in (1.4) *(which is also only a function x_t and t) is optimal.*

Proof. The value of $F(W_h)$ is $\mathbf{C}_h(x_h)$, so the asserted reduction of F is valid at time h. Assume it is valid at time $t+1$. The DP equation is then

$$F(W_t) = \inf_{u_t} \{c(x_t, u_t, t) + E[F(x_{t+1}, t+1) \mid X_t, U_t]\} \,. \tag{1.5}$$

But, by assumption (a), the right-hand side of (1.5) reduces to the right-hand member of (1.4). All the assertions then follow. ■

2 Some Examples of Dynamic Programming

We illustrate the method of dynamic programming and some useful 'tricks'.

2.1 Example: managing spending and savings

An investor receives annual income from a building society of x_t pounds in year t. He consumes u_t and adds $x_t - u_t$ to his capital, $0 \leq u_t \leq x_t$. The capital is invested at interest rate $\theta \times 100\%$, and so his income in year $t+1$ increases to

$$x_{t+1} = a(x_t, u_t) = x_t + \theta(x_t - u_t).$$

He desires to maximize his total consumption over h years, $\mathbf{C} = \sum_{t=0}^{h-1} u_t$.

Solution. In the notation we have been using, $c(x_t, u_t, t) = u_t$, $\mathbf{C}_h(x_h) = 0$. This is a **time-homogeneous** model, in which neither costs nor dynamics depend on t. It is easiest to work in terms of '**time to go**', $s = h - t$. Let $F_s(x)$ denote the maximal reward obtainable, starting in state x and when there is time s to go. The dynamic programming equation is

$$F_s(x) = \max_{0 \leq u \leq x} \left[u + F_{s-1}(x + \theta(x-u)) \right],$$

where $F_0(x) = 0$, (since no more can be obtained once time h is reached.) Here, x and u are generic values for x_s and u_s.

We can substitute backwards and soon guess the form of the solution. First,

$$F_1(x) = \max_{0 \leq u \leq x} \left[u + F_0(u + \theta(x-u)) \right] = \max_{0 \leq u \leq x} [u + 0] = x\,.$$

Next,

$$F_2(x) = \max_{0 \leq u \leq x} \left[u + F_1(x + \theta(x-u)) \right] = \max_{0 \leq u \leq x} \left[u + x + \theta(x-u) \right].$$

Since $u + x + \theta(x-u)$ linear in u, its maximum occurs at $u = 0$ or $u = x$, and so

$$F_2(x) = \max[(1+\theta)x, 2x] = \max[1+\theta, 2]x = \rho_2 x\,.$$

This motivates the guess $F_{s-1}(x) = \rho_{s-1}x$. Trying this, we find

$$F_s(x) = \max_{0 \leq u \leq x} \left[u + \rho_{s-1}(x + \theta(x-u)) \right] = \max[(1+\theta)\rho_{s-1}, 1 + \rho_{s-1}]x = \rho_s x\,.$$

Thus our guess is verified and $F_s(x) = \rho_s x$, where ρ_s obeys the recursion implicit in the above, and i.e., $\rho_s = \rho_{s-1} + \max[\theta\rho_{s-1}, 1]$. This gives

$$\rho_s = \begin{cases} s & s \leq s^* \\ (1+\theta)^{s-s^*} s^* & s \geq s^* \end{cases},$$

where s^* is the least integer such that $s^* \geq 1/\theta$, i.e., $s^* = \lceil 1/\theta \rceil$. The optimal strategy is to invest the whole of the income in years $0, \ldots, h - s^* - 1$, (to build up capital) and then consume the whole of the income in years $h - s^*, \ldots, h-1$.

There are several things worth remembering from this example. (i) It is often useful to frame things in terms of time to go, s. (ii) Although the form of the dynamic programming equation can sometimes look messy, try working backwards from $F_0(x)$ (which is known). Often a pattern will emerge from which we can piece together a solution. (iii) When the dynamics are linear, the optimal control lies at an extreme point of the set of feasible controls. This form of policy, which either consumes nothing or consumes everything, is known as **bang-bang control**.

2.2 Example: exercising a stock option

The owner of a call option has the option to buy a share at fixed 'striking price' p. The option must be exercised by day h. If he exercises the option on day t and then immediately sells the share at the current price x_t, he can make a profit of $x_t - p$. Suppose the price sequence obeys the equation $x_{t+1} = x_t + \epsilon_t$, where the ϵ_t are i.i.d. random variables for which $E|\epsilon| < \infty$. The aim is to exercise the option optimally.

Let $F_s(x)$ be the value function (maximal expected profit) when the share price is x and there are s days to go. Show that (i) $F_s(x)$ is non-decreasing in s, (ii) $F_s(x) - x$ is non-increasing in x and (iii) $F_s(x)$ is continuous in x. Deduce that the optimal policy can be characterised as follows.

There exists a non-decreasing sequence $\{a_s\}$ such that an optimal policy is to exercise the option the first time that $x \geq a_s$, where x is the current price and s is the number of days to go before expiry of the option.

Solution. The state variable at time t is, strictly speaking, x_t plus a variable which indicates whether the option has been exercised or not. However, it is only the latter case which is of interest, so x is the effective state variable. Since dynamic programming makes its calculations backwards, from the termination point, it is often advantageous to write things in terms of the time to go, $s = h - t$. So if we let $F_s(x)$ be the value function (maximal expected profit) with s days to go then

$$F_0(x) = \max\{x - p, 0\},$$

and so the dynamic programming equation is

$$F_s(x) = \max\{x - p, E[F_{s-1}(x + \epsilon)]\}, \quad s = 1, 2, \ldots$$

Note that the expectation operator comes *outside*, not inside, $F_{s-1}(\cdot)$.

One can use induction to show (i), (ii) and (iii). For example, (i) is obvious, since increasing s means we have more time over which to exercise the option. However, for a formal proof

$$F_1(x) = \max\{x - p, E[F_0(x + \epsilon)]\} \geq \max\{x - p, 0\} = F_0(x).$$

Now suppose, inductively, that $F_{s-1} \geq F_{s-2}$. Then

$$F_s(x) = \max\{x - p, E[F_{s-1}(x + \epsilon)]\} \geq \max\{x - p, E[F_{s-2}(x + \epsilon)]\} = F_{s-1}(x),$$

whence F_s is non-decreasing in s. Similarly, an inductive proof of (ii) follows from

$$\underbrace{F_s(x) - x} = \max\{-p, E[\underbrace{F_{s-1}(x+\epsilon) - (x+\epsilon)}] + E(\epsilon)\},$$

since the left hand underbraced term inherits the non-increasing character of the right hand underbraced term. Thus the optimal policy can be characterized as stated. For from (ii), (iii) and the fact that $F_s(x) \geq x - p$ it follows that there exists an a_s such that $F_s(x)$ is greater that $x - p$ if $x < a_s$ and equals $x - p$ if $x \geq a_s$. It follows from (i) that a_s is non-decreasing in s. The constant a_s is the smallest x for which $F_s(x) = x - p$.

2.3 Example: accepting the best offer

We are to interview h candidates for a job. At the end of each interview we must either hire or reject the candidate we have just seen, and may not change this decision later. Candidates are seen in random order and can be ranked against those seen previously. The aim is to maximize the probability of choosing the candidate of greatest rank.

Solution. Let W_t be the history of observations up to time t, i.e., after we have interviewed the tth candidate. All that matters are the value of t and whether the tth candidate is better than all her predecessors: let $x_t = 1$ if this is true and $x_t = 0$ if it is not. In the case $x_t = 1$, the probability she is the best of all h candidates is

$$P(\text{best of } h \mid \text{best of first } t) = \frac{P(\text{best of } h)}{P(\text{best of first } t)} = \frac{1/h}{1/t} = \frac{t}{h}.$$

Now the fact that the tth candidate is the best of the t candidates seen so far places no restriction on the relative ranks of the first $t-1$ candidates; thus $x_t = 1$ and W_{t-1} are statistically independent and we have

$$P(x_t = 1 \mid W_{t-1}) = \frac{P(W_{t-1} \mid x_t = 1)}{P(W_{t-1})} P(x_t = 1) = P(x_t = 1) = \frac{1}{t}.$$

Let $F(t-1)$ be the probability that under an optimal policy we select the best candidate, given that we have passed over the first $t-1$ candidates.Dynamic programming gives

$$F(t-1) = \frac{t-1}{t} F(t) + \frac{1}{t} \max\left(\frac{t}{h}, F(t)\right) = \max\left(\frac{t-1}{t} F(t) + \frac{1}{h}\,,\ F(0,t)\right)$$

The first term deals with what happens when the tth candidate is not the best so far; we should certainly pass over her. The second term deals with what happens when it is. In that case we have a choice: accept that candidate (which will turn out to be best with probability t/h, or pass over that candidate).

These imply $F(t-1) \geq F(t)$ for all $t \leq h$. Therefore, since t/h and $F(t)$ are respectively increasing and non-increasing in t, it must be that for small t we have $F(t) > t/h$ and for large t we have $F(t) \leq t/h$. Let t_0 be the smallest t such that $F(t) \leq t/h$. Then

$$F(t-1) = \begin{cases} F(t_0)\,, & t < t_0\,, \\ \dfrac{t-1}{t} F(t) + \dfrac{1}{h}\,, & t \geq t_0\,. \end{cases}$$

Solving the second of these backwards from the point $t = h$, $F(h) = 0$, we obtain

$$\frac{F(t-1)}{t-1} = \frac{1}{h(t-1)} + \frac{F(t)}{t} = \cdots = \frac{1}{h(t-1)} + \frac{1}{ht} + \cdots + \frac{1}{h(h-1)}\,,$$

whence

$$F(t-1) = \frac{t-1}{h} \sum_{\tau=t-1}^{h-1} \frac{1}{\tau}, \quad t \geq t_0\,.$$

Since we require $F(t_0) \leq t_0/h$, it must be that t_0 is the smallest integer satisfying

$$\sum_{\tau=t_0}^{h-1} \frac{1}{\tau} \leq 1\,.$$

For large h the sum on the left above is about $\log(h/t_0)$, so $\log(h/t_0) \approx 1$ and we find $t_0 \approx h/e$. The optimal policy is to interview $\approx h/e$ candidates, but without selecting any of these, and then select the first one thereafter that is the best of all those seen so far. The probability of success is $F(t_0) \sim t_0/h \sim 1/e = 0.3679$. It is surprising that the probability of success is so large for arbitrarily large h.

There are a couple lessons in this example. (i) It is often useful to try to establish the fact that terms over which a maximum is being taken are monotone in opposite directions, as we did with t/h and $F(t)$. (ii) A typical approach is to first determine the form of the solution, then find the optimal cost (reward) function by backward recursion from the terminal point, where its value is known.

3 Dynamic Programming over the Infinite Horizon

We define the cases of discounted, negative and positive dynamic programming and establish the validity of the optimality equation for an infinite horizon problem.

3.1 Discounted costs

For a discount factor, $\beta \in (0,1]$, the **discounted-cost criterion** is defined as

$$\mathbf{C} = \sum_{t=0}^{h-1} \beta^t c(x_t, u_t, t) + \beta^h \mathbf{C}_h(x_h)\,. \tag{3.1}$$

This simplifies things mathematically, particularly when we want to consider an infinite horizon. If costs are uniformly bounded, say $|c(x,u)| < B$, and discounting is strict ($\beta < 1$) then the infinite horizon cost is bounded by $B/(1-\beta)$. In economic language, if there is an interest rate of $r\%$ per unit time, then a unit amount of money at time t is worth $\rho = 1 + r/100$ at time $t+1$. Equivalently, a unit amount at time $t+1$ has present value $\beta = 1/\rho$. The function, $F(x,t)$, which expresses the minimal present value at time t of expected-cost from time t up to h is

$$F(x,t) = \inf_{u_t,\ldots,u_{h-1}} E\left[\sum_{\tau=t}^{h-1} \beta^{\tau-t} c(x_\tau, u_\tau, \tau) + \beta^{h-t}\mathbf{C}_h(x_h) \,\middle|\, x_t = x \right]. \tag{3.2}$$

The DP equation is now

$$F(x,t) = \inf_u \left[c(x,u,t) + \beta E F(a(x,u,t), t+1) \right], \quad t < h\,, \tag{3.3}$$

where $F(x,h) = \mathbf{C}_h(x)$.

3.2 Example: job scheduling

A collection of n jobs is to be processed in arbitrary order by a single machine. Job i has processing time p_i and when it completes a reward r_i is obtained. Find the order of processing that maximizes the sum of the discounted rewards.

Solution. Here we take 'time k' as the point at which the $n-k$ th job has just been completed and the state at time k as the collection of uncompleted jobs, say S_k. The dynamic programming equation is

$$F_k(S_k) = \max_{i \in S_k} [r_i \beta^{p_i} + \beta^{p_i} F_{k-1}(S_k - \{i\})]\,.$$

Obviously $F_0(\emptyset) = 0$. Applying the method of dynamic programming we first find $F_1(\{i\}) = r_i \beta^{p_i}$. Then, working backwards, we find

$$F_2(\{i,j\}) \quad = \quad \max[r_i\beta^{p_i} + \beta^{p_i+p_j} r_j\,,\, r_j\beta^{p_j} + \beta^{p_j+p_i} r_i]\,.$$

There will be 2^n equations to evaluate, but with perseverance we can determine $F_n(\{1,2,\ldots,n\})$. However, there is a simpler way.

An interchange argument. Suppose that jobs are scheduled in the order $i_1,\ldots,i_k,i,j,i_{k+3},\ldots,i_n$. Compare the reward of this schedule to one in which the order of jobs i and j are reversed: $i_1,\ldots,i_k,j,i,i_{k+3},\ldots,i_n$. The rewards under the two schedules are respectively

$$R_1 + \beta^{T+p_i} r_i + \beta^{T+p_i+p_j} r_j + R_2 \quad \text{and} \quad R_1 + \beta^{T+p_j} r_j + \beta^{T+p_j+p_i} r_i + R_2\,,$$

where $T = p_{i_1} + \cdots + p_{i_k}$, and R_1 and R_2 are respectively the sum of the rewards due to the jobs coming before and after jobs i, j; these are the same under both schedules. The reward of the first schedule is greater if $r_i\beta^{p_i}/(1-\beta^{p_i}) > r_j\beta^{p_j}/(1-\beta^{p_j})$. Hence a schedule can be optimal only if the jobs are taken in decreasing order of the indices $r_i\beta^{p_i}/(1-\beta^{p_i})$. This type of reasoning is known as an **interchange argument**.

There are a couple points to note. (i) An interchange argument can be useful for solving a decision problem about a system that evolves in stages. Although such problems can be solved by dynamic programming, an interchange argument – when it works – is usually easier. (ii) The decision points need not be equally spaced in time. Here they are the points at which a number of jobs have been completed.

3.3 The infinite-horizon case

In the finite-horizon case the cost function is obtained simply from (3.3) by the backward recursion from the terminal point. However, when the horizon is infinite there is no terminal point and so the validity of the optimality equation is no longer obvious.

Let us consider the time-homogeneous Markov case, in which costs and dynamics do not depend on t, i.e., $c(x,u,t) = c(x,u)$. Suppose also that there is no terminal cost, i.e., $\mathbf{C}_h(x) = 0$. Define the *s-horizon cost under policy* π as

$$F_s(\pi, x) = E_\pi\left[\sum_{t=0}^{s-1} \beta^t c(x_t, u_t) \,\middle|\, x_0 = x \right],$$

where E_π denotes expectation over the path of the process under policy π. If we take the infimum with respect to π we have the *infimal s-horizon cost*

$$F_s(x) = \inf_\pi F_s(\pi, x).$$

Clearly, this always exists and satisfies the optimality equation

$$F_s(x) = \inf_u \left\{ c(x,u) + \beta E[F_{s-1}(x_1) \mid x_0 = x, u_0 = u] \right\}, \tag{3.4}$$

with terminal condition $F_0(x) = 0$.

The *infinite-horizon cost under policy* π is also quite naturally defined as

$$F(\pi, x) = \lim_{s\to\infty} F_s(\pi, x). \tag{3.5}$$

This limit need not exist, but it will do so under any of the following scenarios.

D (**discounted programming**): $0 < \beta < 1$, and $|c(x,u)| < B$ for all x, u.

N (**negative programming**): $0 < \beta \le 1$ and $c(x,u) \ge 0$ for all x, u.

P (**positive programming**): $0 < \beta \le 1$ and $c(x,u) \le 0$ for all x, u.

Notice that the names 'negative' and 'positive' appear to be the wrong way around with respect to the sign of $c(x,u)$. However, the names make sense if we think of equivalent problems of maximizing rewards. Maximizing positive rewards (P) is the same thing as minimizing negative costs. Maximizing negative rewards (N) is the same thing as minimizing positive costs. In cases N and P we usually take $\beta = 1$.

The existence of the limit (possibly infinite) in (3.5) is assured in cases N and P by monotone convergence, and in case D because the total cost occurring after the sth step is bounded by $\beta^s B/(1-\beta)$.

3.4 The optimality equation in the infinite-horizon case

The *infimal infinite-horizon cost* is defined as

$$F(x) = \inf_{\pi} F(\pi, x) = \inf_{\pi} \lim_{s\to\infty} F_s(\pi, x) \,. \tag{3.6}$$

The following theorem justifies our writing an optimality equation.

Theorem 3.1 *Suppose D, N, or P holds. Then $F(x)$ satisfies the optimality equation*

$$F(x) = \inf_{u} \{c(x,u) + \beta E[F(x_1) \mid x_0 = x, u_0 = u)]\} \,. \tag{3.7}$$

Proof. We first prove that '$\ge$' holds in (3.7). Suppose π is a policy, which chooses $u_0 = u$ when $x_0 = x$. Then

$$F_s(\pi, x) = c(x,u) + \beta E[F_{s-1}(\pi, x_1) \mid x_0 = x, u_0 = u] \,. \tag{3.8}$$

Either D, N or P is sufficient to allow us to takes limits on both sides of (3.8) and interchange the order of limit and expectation. In cases N and P this is because of monotone convergence. Infinity is allowed as a possible limiting value. We obtain

$$\begin{aligned} F(\pi, x) &= c(x,u) + \beta E[F(\pi, x_1) \mid x_0 = x, u_0 = u] \\ &\ge c(x,u) + \beta E[F(x_1) \mid x_0 = x, u_0 = u] \\ &\ge \inf_{u} \{c(x,u) + \beta E[F(x_1) \mid x_0 = x, u_0 = u]\} \,. \end{aligned}$$

Minimizing the left hand side over π gives '$\ge$'.

To prove '$\le$', fix x and consider a policy π that having chosen u_0 and reached state x_1 then follows a policy π^1 which is suboptimal by less than ϵ from that point, i.e., $F(\pi^1, x_1) \le F(x_1) + \epsilon$. Note that such a policy must exist, by definition of F, although π^1 will depend on x_1. We have

$$\begin{aligned} F(x) &\le F(\pi, x) \\ &= c(x, u_0) + \beta E[F(\pi^1, x_1) \mid x_0 = x, u_0] \\ &\le c(x, u_0) + \beta E[F(x_1) + \epsilon \mid x_0 = x, u_0] \\ &\le c(x, u_0) + \beta E[F(x_1) \mid x_0 = x, u_0] + \beta\epsilon \,. \end{aligned}$$

Minimizing the right hand side over u_0 and recalling ϵ is arbitrary gives '$\le$'. ■

3.5 Example: selling an asset

A spectulator owns a rare collection of tulip bulbs and each day has one opportunity to sell it, which he may either accept or reject. The potential sale prices are independently and identically distributed with probability density function $g(x)$, $x \ge 0$. Each day there is a probability $1-\beta$ that the market for tulip bulbs will collapse, making his bulb collection completely worthless. Find the policy that maximizes his expected return and express it as the unique root of an equation. Show that if $\beta > 1/2$, $g(x) = 2/x^3$, $x \ge 1$, then he should sell the first time the sale price is at least $\sqrt{\beta/(1-\beta)}$.

Solution. There are only two states, depending on whether he has sold the collection or not. Let these be 0 and 1 respectively. The optimality equation is

$$\begin{aligned} F(1) &= \int_{y=0}^{\infty} \max[y, \beta F(1)]\, g(y)\, dy \\ &= \beta F(1) + \int_{y=0}^{\infty} \max[y - \beta F(1), 0]\, g(y)\, dy \\ &= \beta F(1) + \int_{y=\beta F(1)}^{\infty} [y - \beta F(1)]\, g(y)\, dy \end{aligned}$$

Hence

$$(1-\beta)F(1) = \int_{y=\beta F(1)}^{\infty} [y - \beta F(1)]\, g(y)\, dy \,. \tag{3.9}$$

That this equation has a unique root, $F(1) = F^*$, follows from the fact that left and right hand sides are increasing and decreasing in $F(1)$ respectively. Thus he should sell when he can get at least βF^*. His maximal reward is F^*.

Consider the case $g(y) = 2/y^3$, $y \ge 1$. The left hand side of (3.9) is less that the right hand side at $F(1) = 1$ provided $\beta > 1/2$. In this case the root is greater than 1 and we compute it as

$$(1-\beta)F(1) = 2/\beta F(1) - \beta F(1)/[\beta F(1)]^2 \,,$$

and thus $F^* = 1/\sqrt{\beta(1-\beta)}$ and $\beta F^* = \sqrt{\beta/(1-\beta)}$.

If $\beta \le 1/2$ he should sell at any price.

Notice that discounting arises in this problem because at each stage there is a probability $1-\beta$ that a 'catastrophe' will occur that brings things to a sudden end. This characterization of a manner in which discounting can arise is often quite useful.

4 Positive Programming

We address the special theory of maximizing positive rewards, (noting that there may be no optimal policy but that if a policy has a value function that satisfies the optimality equation then it is optimal), and the method of value iteration.

4.1 Example: possible lack of an optimal policy.

Positive programming concerns minimizing non-positive costs, $c(x, u) \leq 0$. The name originates from the equivalent problem of maximizing non-negative rewards, $r(x, u) \geq 0$, and for this section we present results in that setting. The following example shows that there may be no optimal policy.

Suppose the possible states are the non-negative integers and in state x we have a choice of either moving to state $x + 1$ and receiving no reward, or moving to state 0, obtaining reward $1 - 1/i$, and then remaining in state 0 thereafter and obtaining no further reward. The optimality equations is

$$F(x) = \max\{1 - 1/x, F(x + 1)\} \quad x > 0 .$$

Clearly $F(x) = 1$, $x > 0$, but the policy that chooses the maximizing action in the optimality equation always moves on to state $x+1$ and hence has zero reward. Clearly, there is no policy that actually achieves a reward of 1.

4.2 Characterization of the optimal policy

The following theorem provides a necessary and sufficient condition for a policy to be optimal: namely, its value function must satisfy the optimality equation. This theorem also holds for the case of strict discounting and bounded costs.

Theorem 4.1 *Suppose D or P holds and π is a policy whose value function $F(\pi, x)$ satisfies the optimality equation*

$$F(\pi, x) = \sup_u\{r(x, u) + \beta E[F(\pi, x_1) \mid x_0 = x, u_0 = u]\} .$$

Then π is optimal.

Proof. Let π' be any policy and suppose it takes $u_t(x) = f_t(x)$. Since $F(\pi, x)$ satisfies the optimality equation,

$$F(\pi, x) \geq r(x, f_0(x)) + \beta E_{\pi'}[F(\pi, x_1) \mid x_0 = x, u_0 = f_0(x)] .$$

By repeated substitution of this into itself, we find

$$F(\pi, x) \geq E_{\pi'}\left[\sum_{t=0}^{s-1} \beta^t r(x_t, u_t) \,\middle|\, x_0 = x\right] + \beta^s E_{\pi'}[F(\pi, x_s) \mid x_0 = x] . \qquad (4.1)$$

In case P we can drop the final term on the right hand side of (4.1) (because it is non-negative) and then let $s \to \infty$; in case D we can let $s \to \infty$ directly, observing that this term tends to zero. Either way, we have $F(\pi, x) \geq F(\pi', x)$. ■

4.3 Example: optimal gambling

A gambler has i pounds and wants to increase this to N. At each stage she can bet any fraction of her capital, say $j \leq i$. Either she wins, with probability p, and now has $i+j$ pounds, or she loses, with probability $q = 1 - p$, and has $i - j$ pounds. Let the state space be $\{0, 1, \dots, N\}$. The game stops upon reaching state 0 or N. The only non-zero reward is 1, upon reaching state N. Suppose $p \geq 1/2$. Prove that the timid strategy, of always betting only 1 pound, maximizes the probability of the gambler attaining N pounds.

Solution. The optimality equation is

$$F(i) = \max_{j, j \leq i}\{pF(i + j) + qF(i - j)\} .$$

To show that the timid strategy is optimal we need to find its value function, say $G(i)$, and show that it is a solution to the optimality equation. We have $G(i) = pG(i + 1) + qG(i - 1)$, with $G(0) = 0$, $G(N) = 1$. This recurrence gives

$$G(i) = \begin{cases} \dfrac{1 - (q/p)^i}{1 - (q/p)^N} & p > 1/2, \\[2ex] \dfrac{i}{N} & p = 1/2. \end{cases}$$

If $p = 1/2$, then $G(i) = i/N$ clearly satisfies the optimality equation. If $p > 1/2$ we simply have to verify that

$$G(i) = \frac{1 - (q/p)^i}{1 - (q/p)^N} = \max_{j:j \leq i}\left\{ p\left[\frac{1 - (q/p)^{i+j}}{1 - (q/p)^N}\right] + q\left[\frac{1 - (q/p)^{i-j}}{1 - (q/p)^N}\right] \right\} .$$

It is a simple exercise to show that $j = 1$ maximizes the right hand side. ■

4.4 Value iteration

The infimal cost function F can be approximated by **successive approximation** or **value iteration**. This is important and practical method of computing F. Let us define

$$F_\infty(x) = \lim_{s\to\infty} F_s(x) = \lim_{s\to\infty} \inf_\pi F_s(\pi, x) . \qquad (4.2)$$

This exists (by monotone convergence under N or P, or by the fact that under D the cost incurred after time s is vanishingly small.)

Notice that (4.2) reverses the order of $\lim_{s\to\infty}$ and $\inf_\pi$ in (3.6). The following theorem states that we can interchange the order of these operations and that therefore

$F_s(x) \to F(x)$. However, in case N we need an additional assumption:

F (**finite actions**): There are only finitely many possible values of u in each state.

Theorem 4.2 *Suppose that D or P holds, or N and F hold. Then $F_\infty(x) = F(x)$.*

Proof. First we prove '$\leq$'. Given any $\bar{\pi}$,

$$F_\infty(x) = \lim_{s\to\infty} F_s(x) = \lim_{s\to\infty} \inf_\pi F_s(\pi, x) \leq \lim_{s\to\infty} F_s(\bar{\pi}, x) = F(\bar{\pi}, x).$$

Taking the infimum over $\bar{\pi}$ gives $F_\infty(x) \leq F(x)$.

Now we prove '$\geq$'. In the positive case, $c(x,u) \leq 0$, so $F_s(x) \geq F(x)$. Now let $s \to \infty$. In the discounted case, with $|c(x,u)| < B$, imagine subtracting $B > 0$ from every cost. This reduces the infinite-horizon cost under any policy by exactly $B/(1-\beta)$ and $F(x)$ and $F_\infty(x)$ also decrease by this amount. All costs are now negative, so the result we have just proved applies. [Alternatively, note that

$$F_s(x) - \beta^s B/(1-\beta) \leq F(x) \leq F_s(x) + \beta^s B/(1-\beta)$$

(can you see why?) and hence $\lim_{s\to\infty} F_s(x) = F(x)$.]

In the negative case,

$$\begin{aligned} F_\infty(x) &= \lim_{s\to\infty} \min_u \{c(x,u) + E[F_{s-1}(x_1) \mid x_0 = x, u_0 = u]\} \\ &= \min_u \{c(x,u) + \lim_{s\to\infty} E[F_{s-1}(x_1) \mid x_0 = x, u_0 = u]\} \\ &= \min_u \{c(x,u) + E[F_\infty(x_1) \mid x_0 = x, u_0 = u]\}, \qquad (4.3) \end{aligned}$$

where the first equality follows because the minimum is over a finite number of terms and the second equality follows by Lebesgue monotone convergence (since $F_s(x)$ increases in s). Let π be the policy that chooses the minimizing action on the right hand side of (4.3). This implies, by substitution of (4.3) into itself, and using the fact that N implies $F_\infty \geq 0$,

$$\begin{aligned} F_\infty(x) &= E_\pi\left[\left.\sum_{t=0}^{s-1} c(x_t, u_t) + F_\infty(x_s) \,\right|\, x_0 = x\right] \\ &\geq E_\pi\left[\left.\sum_{t=0}^{s-1} c(x_t, u_t) \,\right|\, x_0 = x\right]. \end{aligned}$$

Letting $s \to \infty$ gives $F_\infty(x) \geq F(\pi, x) \geq F(x)$. ■

4.5 Example: pharmaceutical trials

A doctor has two drugs available to treat a disease. One is well-established drug and is known to work for a given patient with probability p, independently of its success for other patients. The new drug is untested and has an unknown probability of success θ, which the doctor believes to be uniformly distributed over $[0,1]$. He treats one patient per day and must choose which drug to use. Suppose he has observed s successes and f failures with the new drug. Let $F(s,f)$ be the maximal expected-discounted number of future patients who are successfully treated if he chooses between the drugs optimally from this point onwards. For example, if he uses only the established drug, the expected-discounted number of patients successfully treated is $p + \beta p + \beta^2 p + \cdots = p/(1-\beta)$. The posterior distribution of θ is

$$f(\theta \mid s, f) = \frac{(s+f+1)!}{s!f!}\theta^s(1-\theta)^f, \quad 0 \leq \theta \leq 1,$$

and the posterior mean is $\bar{\theta}(s,f) = (s+1)/(s+f+2)$. The optimality equation is

$$F(s,f) = \max\left[\frac{p}{1-\beta}, \frac{s+1}{s+f+2}(1 + \beta F(s+1, f)) + \frac{f+1}{s+f+2}\beta F(s, f+1)\right].$$

It is not possible to give a nice expression for F, but we can find an approximate numerical solution. If $s+f$ is very large, say 300, then $\bar{\theta}(s,f) = (s+1)/(s+f+2)$ is a good approximation to θ. Thus we can take $F(s,f) \approx (1-\beta)^{-1}\max[p, \bar{\theta}(s,f)]$, $s+f = 300$ and work backwards. For $\beta = 0.95$, one obtains the following table.

f s	0	1	2	3	4	5
0	.7614	.8381	.8736	.8948	.9092	.9197
1	.5601	.6810	.7443	.7845	.8128	.8340
2	.4334	.5621	.6392	.6903	.7281	.7568
3	.3477	.4753	.5556	.6133	.6563	.6899
4	.2877	.4094	.4898	.5493	.5957	.6326

These numbers are the greatest values of p for which it is worth continuing with at least one more trial of the new drug. For example, with $s = 3$, $f = 3$ it is worth continuing with the new drug when $p = 0.6 < 0.6133$. At this point the probability that the new drug will successfully treat the next patient is 0.5 and so the doctor should actually prescribe the drug that is least likely to cure! This example shows the difference between a **myopic policy**, which aims to maximize immediate reward, and an optimal policy, which forgets immediate reward in order to gain information and possibly greater rewards later on. Notice that it is worth using the new drug at least once if $p < 0.7614$, even though at its first use the new drug will only be successful with probability 0.5.

5 Negative Programming

We address the special theory of minimizing positive costs, (noting that the action that extremizes the right hand side of the optimality equation gives an optimal policy), and stopping problems and their solution.

5.1 Stationary policies

A **Markov policy** is a policy that specifies the control at time t to be simply a function of the state and time. In the proof of Theorem 4.1 we used $u_t = f_t(x_t)$ to specify the control at time t. This is a convenient notation for a Markov policy, and we write $\pi = (f_0, f_1, \dots)$. If in addition the policy does not depend on time, it is said to be a **stationary Markov policy**, and we write $\pi = (f, f, \dots) = f^\infty$.

5.2 Characterization of the optimal policy

Negative programming concerns minimizing non-negative costs, $c(x,u) \geq 0$. The name originates from the equivalent problem of maximizing non-positive rewards, $r(x,u) \leq 0$.

The following theorem gives a necessary and sufficient condition for a stationary policy to be optimal: namely, it must choose the optimal u on the right hand side of the optimality equation. Note that in the statement of this theorem we are requiring that the infimum over u is attained as a minimum over u.

Theorem 5.1 *Suppose D or N holds. Suppose $\pi = f^\infty$ is the stationary Markov policy such that*

$$\begin{aligned} c(x, f(x)) &+ \beta E[F(x_1) \mid x_0 = x, u_0 = f(x)] \\ &= \min_u \left[c(x,u) + \beta E[F(x_1) \mid x_0 = x, u_0 = u \right] . \end{aligned}$$

Then $F(\pi, x) = F(x)$, and π is optimal.

Proof. Suppose this policy is $\pi = f^\infty$. Then by substituting the optimality equation into itself and using the fact that π specifies the minimizing control at each stage,

$$F(x) = E_\pi \left[\sum_{t=0}^{s-1} \beta^t c(x_t, u_t) \middle| \, x_0 = x \right] + \beta^s E_\pi \left[F(x_s) \middle| \, x_0 = x \right] . \tag{5.1}$$

In case N we can drop the final term on the right hand side of (5.1) (because it is non-negative) and then let $s \to \infty$; in case D we can let $s \to \infty$ directly, observing that this term tends to zero. Either way, we have $F(x) \geq F(\pi, x)$. ■

A corollary is that if assumption F holds then an optimal policy exists. Neither Theorem 5.1 or this corollary are true for positive programming (c.f., the example in Section 4.1).

5.3 Optimal stopping over a finite horizon

One way that the total-expected cost can be finite is if it is possible to enter a state from which no further costs are incurred. Suppose u has just two possible values: $u = 0$ (stop), and $u = 1$ (continue). Suppose there is a termination state, say 0, that is entered upon choosing the stopping action. Once this state is entered the system stays in that state and no further cost is incurred thereafter.

Suppose that stopping is mandatory, in that we must continue for no more that s steps. The finite-horizon dynamic programming equation is therefore

$$F_s(x) = \min\{k(x), c(x) + E[F_{s-1}(x_1) \mid x_0 = x, u_0 = 1]\} , \tag{5.2}$$

with $F_0(x) = k(x)$, $c(0) = 0$.

Consider the set of states in which it is at least as good to stop now as to continue one more step and then stop:

$$S = \{x : k(x) \leq c(x) + E[k(x_1) \mid x_0 = x, u_0 = 1)]\} .$$

Clearly, it cannot be optimal to stop if $x \notin S$, since in that case it would be strictly better to continue one more step and then stop. The following theorem characterises all finite-horizon optimal policies.

Theorem 5.2 *Suppose S is closed (so that once the state enters S it remains in S.) Then an optimal policy for all finite horizons is: stop if and only if $x \in S$.*

Proof. The proof is by induction. If the horizon is $s = 1$, then obviously it is optimal to stop only if $x \in S$. Suppose the theorem is true for a horizon of $s - 1$. As above, if $x \notin S$ then it is better to continue for more one step and stop rather than stop in state x. If $x \in S$, then the fact that S is closed implies $x_1 \in S$ and so $F_{s-1}(x_1) = k(x_1)$. But then (5.2) gives $F_s(x) = k(x)$. So we should stop if $s \in S$. ■

The optimal policy is known as a **one-step look-ahead rule** (OSLA).

5.4 Example: optimal parking

A driver is looking for a parking space on the way to his destination. Each parking space is free with probability p independently of whether other parking spaces are free or not. The driver cannot observe whether a parking space is free until he reaches it. If he parks s spaces from the destination, he incurs cost s, $s = 0, 1, \dots$. If he passes the destination without having parked the cost is D. Show that an optimal policy is to park in the first free space that is no further than s^* from the destination, where s^* is the greatest integer s such that $(Dp + 1)q^s \geq 1$.

Solution. When the driver is s spaces from the destination it only matters whether the space is available ($x = 1$) or full ($x = 0$). The optimality equation gives

$$F_s(0) = qF_{s-1}(0) + pF_{s-1}(1),$$

$$F_s(1) = \min \begin{cases} s, & \text{(take available space)} \\ qF_{s-1}(0) + pF_{s-1}(1), & \text{(ignore available space)} \end{cases}$$

where $F_0(0) = D$, $F_0(1) = 0$.

Suppose the driver adopts a policy of taking the first free space that is s or closer. Let the cost under this policy be $k(s)$, where

$$k(s) = ps + qk(s-1)\,,$$

with $k(0) = qD$. The general solution is of the form $k(s) = -q/p + s + cq^s$. So after substituting and using the boundary condition at $s = 0$, we have

$$k(s) = -\frac{q}{p} + s + \left(D + \frac{1}{p}\right) q^{s+1}, \quad s = 0, 1, \dots\ .$$

It is better to stop now (at a distance s from the destination) than to go on and take the first available space if s is in the stopping set

$$S = \{s : s \le k(s-1)\} = \{s : (Dp+1)q^s \ge 1\}\,.$$

This set is closed (since s decreases) and so by Theorem 5.2 this stopping set describes the optimal policy. ■

If the driver parks in the first available space past his destination and walk backs, then $D = 1 + qD$, so $D = 1/p$ and s^* is the greatest integer such that $2q^s \ge 1$.

5.5 Optimal stopping over the infinite horizon

Let us now consider the stopping problem over the infinite-horizon. As above, let $F_s(x)$ be the infimal cost given that we are required to stop by time s. Let $F(x)$ be the infimal cost when all that is required is that we stop eventually. Since less cost can be incurred if we are allowed more time in which to stop, we have

$$F_s(x) \ge F_{s+1}(x) \ge F(x)\,.$$

Thus by monotone convergence $F_s(x)$ tends to a limit, say $F_\infty(x)$, and $F_\infty(x) \ge F(x)$.

Example: we can have $F_\infty > F$

Consider the problem of stopping a symmetric random walk on the integers, where $c(x) = 0$, $k(x) = \exp(-x)$. The policy of stopping immediately, π, has $F(\pi, x) = \exp(-x)$, and this satisfies the infinite-horizon optimality equation,

$$F(x) = \min\{\exp(-x), (1/2)F(x+1) + (1/2)F(x-1)\}\,.$$

However, π is not optimal. A symmetric random walk is recurrent, so we may wait until reaching as large an integer as we like before stopping; hence $F(x) = 0$. Inductively, one can see that $F_s(x) = \exp(-x)$. So $F_\infty(x) > F(x)$.

(Note: Theorem 4.2 says that $F_\infty = F$, but that is in a setting in which there is no terminal cost and for different definitions of F_s and F than we take here.)

Example: Theorem 4.1 is not true for negative programming

Consider the above example, but now suppose one is allowed never to stop. Since continuation costs are 0 the optimal policy for all finite horizons and the infinite horizon is never to stop. So $F(x) = 0$ and this satisfies the optimality equation above. However, $F(\pi, x) = \exp(-x)$ also satisfies the optimality equation and is the cost incurred by stopping immediately. Thus it is not true (as for positive programming) that a policy whose cost function satisfies the optimality equation is optimal.

The following lemma gives conditions under which the infimal finite-horizon cost does converge to the infimal infinite-horizon cost.

Lemma 5.3 *Suppose all costs are bounded as follows.*

$$(a)\ K = \sup_x k(x) < \infty \qquad (b)\ C = \inf_x c(x) > 0\,. \tag{5.3}$$

Then $F_s(x) \to F(x)$ as $s \to \infty$.

Proof. (*starred*) Suppose π is an optimal policy for the infinite horizon problem and stops at the random time τ. Then its cost is at least $(s+1)CP(\tau > s)$. However, since it would be possible to stop at time 0 the cost is also no more than K, so

$$(s+1)CP(\tau > s) \le F(x) \le K\,.$$

In the s-horizon problem we could follow π, but stop at time s if $\tau > s$. This implies

$$F(x) \le F_s(x) \le F(x) + KP(\tau > s) \le F(x) + \frac{K^2}{(s+1)C}\,.$$

By letting $s \to \infty$, we have $F_\infty(x) = F(x)$. ■

Note that the problem posed here is identical to one in which we pay K at the start and receive a terminal reward $r(x) = K - k(x)$.

Theorem 5.4 *Suppose S is closed and (5.3) holds. Then an optimal policy for the infinite horizon is: stop if and only if $x \in S$.*

Proof. By Theorem 5.2 we have for all finite s,

$$F_s(x) \begin{array}{ll} = k(x) & x \in S\,, \\ < k(x) & x \notin S\,. \end{array}$$

Lemma 5.3 gives $F(x) = F_\infty(x)$. ■

6 Average-cost Programming

We address the infinite-horizon average-cost case, the optimality equation for this case and the policy improvement algorithm.

6.1 Average-cost optimization

It can happen that the undiscounted expected total cost is infinite, but the accumulation of cost per unit time is finite. Suppose that for a stationary Markov policy π, the following limit exists:

$$\lambda(\pi, x) = \lim_{t\to\infty} \frac{1}{t} E_\pi \left[\sum_{s=0}^{t-1} c(x_s, u_s) \middle| x_0 = x \right] .$$

It is reasonable to expect that there is a well-defined notion of an optimal **average-cost** function, $\lambda(x) = \inf_\pi \lambda(\pi, x)$, and that under appropriate assumptions, $\lambda(x) = \lambda$ should not depend on x. Moreover, one would expect

$$F_s(x) = s\lambda + \phi(x) + \epsilon(s, x) ,$$

where $\epsilon(s, x) \to 0$ as $s \to \infty$. Here $\phi(x) + \epsilon(s, x)$ reflects a transient due to the initial state. Suppose that the state space and action space are finite. From the optimality equation for the finite horizon problem we have

$$F_s(x) = \min_u \{c(x, u) + E[F_{s-1}(x_1) \mid x_0 = x, u_0 = u]\} . \tag{6.1}$$

So by substituting $F_s(x) \sim s\lambda + \phi(x)$ into (6.1), we obtain

$$s\lambda + \phi(x) \sim \min_u \{c(x, u) + E[(s-1)\lambda + \phi(x_1) \mid x_0 = x, u_0 = u]\}$$

which suggests, what it is in fact, the average-cost optimality equation:

$$\lambda + \phi(x) = \min_u \{c(x, u) + E[\phi(x_1) \mid x_0 = x, u_0 = u]\} . \tag{6.2}$$

Theorem 6.1 *Let λ denote the minimal average-cost. Suppose there exists a constant λ' and bounded function ϕ such that for all x and u,*

$$\lambda' + \phi(x) \le c(x, u) + E[\phi(x_1) \mid x_0 = x, u_0 = u] . \tag{6.3}$$

Then $\lambda' \le \lambda$. This also holds when $\le$ is replaced by $\ge$ and the hypothesis is weakened to: for each x there exists a u such that (6.3) *holds when $\le$ is replaced by $\ge$.*

Proof. Suppose u is chosen by some policy π. By repeated substitution of (6.3) into itself we have

$$\phi(x) \le -t\lambda' + E_\pi \left[\sum_{s=0}^{t-1} c(x_s, u_s) \middle| x_0 = x \right] + E_\pi[\phi(x_t) \mid x_0 = x]$$

Divide this by t and let $t \to \infty$ to obtain

$$0 \le -\lambda' + \lim_{t\to\infty} \frac{1}{t} E_\pi \left[\sum_{s=0}^{t-1} c(x_s, u_s) \middle| x_0 = x \right] ,$$

where the final term on the right hand side is simply the average-cost under policy π. Minimizing the right hand side over π gives the result. The claim for $\le$ replaced by $\ge$ is proved similarly. ■

Theorem 6.2 *Suppose there exists a constant λ and bounded function ϕ satisfying* (6.2)*. Then λ is the minimal average-cost and the optimal stationary policy is the one that chooses the optimizing u on the right hand side of* (6.2)*.*

Proof. Equation (6.2) implies that (6.3) holds with equality when one takes π to be the stationary policy that chooses the optimizing u on the right hand side of (6.2). Thus π is optimal and λ is the minimal average-cost. ■

The average-cost optimal policy is found simply by looking for a bounded solution to (6.2). Notice that if ϕ is a solution of (6.2) then so is ϕ+(a constant), because the (a constant) will cancel from both sides of (6.2). Thus ϕ is undetermined up to an additive constant. In searching for a solution to (6.2) we can therefore pick any state, say $\bar{x}$, and arbitrarily take $\phi(\bar{x}) = 0$.

6.2 Example: admission control at a queue

Each day a consultant is presented with the opportunity to take on a new job. The jobs are independently distributed over n possible types and on a given day the offered type is i with probability a_i, $i = 1, \dots, n$. Jobs of type i pay R_i upon completion. Once he has accepted a job he may accept no other job until that job is complete. The probability that a job of type i takes k days is $(1 - p_i)^{k-1} p_i$, $k = 1, 2, \dots$. Which jobs should the consultant accept?

Solution. Let 0 and i denote the states in which he is free to accept a job, and in which he is engaged upon a job of type i, respectively. Then (6.2) is

$$\begin{aligned} \lambda + \phi(0) &= \sum_{i=1}^{n} a_i \max[\phi(0), \phi(i)], \\ \lambda + \phi(i) &= (1 - p_i)\phi(i) + p_i[R_i + \phi(0)], \quad i = 1, \dots, n . \end{aligned}$$

Taking $\phi(0) = 0$, these have solution $\phi(i) = R_i - \lambda/p_i$, and hence

$$\lambda = \sum_{i=1}^{n} a_i \max[0, R_i - \lambda/p_i] .$$

The left hand side is increasing in λ and the right hand side is decreasing λ. Hence there is a root, say λ^*, and this is the maximal average-reward. The optimal policy takes the form: *accept only jobs for which $p_i R_i \ge \lambda^*$.* ■

6.3 Value iteration bounds

Value iteration in the average-cost case is based upon the idea that $F_s(x) - F_{s-1}(x)$ approximates the minimal average-cost for large s.

Theorem 6.3 *Define*

$$m_s = \min_x\{F_s(x) - F_{s-1}(x)\}\,, \qquad M_s = \max_x\{F_s(x) - F_{s-1}(x)\}\,. \tag{6.4}$$

Then $m_s \le \lambda \le M_s$, where λ is the minimal average-cost.

Proof. (*starred*) Suppose that the first step of a s-horizon optimal policy follows Markov plan f. Then

$$F_s(x) = F_{s-1}(x) + [F_s(x) - F_{s-1}(x)] = c(x, f(x)) + E[F_{s-1}(x_1) \mid x_0 = x, u_0 = f(x)]\,.$$

Hence

$$F_{s-1}(x) + m_s \le c(x,u) + E[F_{s-1}(x_1) \mid x_0 = x, u_0 = u]\,,$$

for all x, u. Applying Theorem 6.1 with $\phi = F_{s-1}$ and $\lambda' = m_s$, implies $m_s \le \lambda$. The bound $\lambda \le M_s$ is established in a similar way. ■

This justifies the following **value iteration algorithm**. At termination the algorithm provides a stationary policy that is within $\epsilon \times 100\%$ of optimal.

(0) Set $F_0(x) = 0$, $s = 1$.
(1) Compute F_s from

$$F_s(x) = \min_u\{c(x,u) + E[F_{s-1}(x_1) \mid x_0 = x, u_0 = u]\}\,.$$

(2) Compute m_s and M_s from (6.4). Stop if $M_s - m_s \le \epsilon m_s$. Otherwise set $s := s+1$ and goto step (1).

6.4 Policy improvement

Policy improvement is an effective method of improving stationary policies.

Policy improvement in the average-cost case.

In the average-cost case a policy improvement algorithm can be based on the following observations. Suppose that for a policy $\pi = f^\infty$, we have that λ, ϕ is a solution to

$$\lambda + \phi(x) = c(x, f(x_0)) + E[\phi(x_1) \mid x_0 = x, u_0 = f(x_0)]\,,$$

and suppose for some policy $\pi_1 = f_1^\infty$,

$$\lambda + \phi(x) \ge c(x, f_1(x_0)) + E[\phi(x_1) \mid x_0 = x, u_0 = f_1(x_0)]\,, \tag{6.5}$$

with strict inequality for some x. Then following the lines of proof in Theorem 6.1

$$\lim_{t\to\infty} \frac{1}{t} E_\pi \left[\sum_{s=0}^{t-1} c(x_s, u_s) \middle| x_0 = x \right] = \lambda \ge \lim_{t\to\infty} \frac{1}{t} E_{\pi_1} \left[\sum_{s=0}^{t-1} c(x_s, u_s) \middle| x_0 = x \right]\,.$$

If there is no π_1 for which (6.5) holds then π satisfies (6.2) and is optimal. This justifies the following **policy improvement algorithm**

(0) Choose an arbitrary stationary policy π_0. Set $s = 1$.
(1) For a given stationary policy $\pi_{s-1} = f_{s-1}^\infty$ determine ϕ, λ to solve

$$\lambda + \phi(x) = c(x, f_{s-1}(x)) + E[\phi(x_1) \mid x_0 = x, u_0 = f_{s-1}(x)]\,.$$

This gives a set of linear equations, and so is intrinsically easier to solve than (6.2).
(2) Now determine the policy $\pi_s = f_s^\infty$ from

$$\begin{aligned} &c(x, f_s(x)) + E[\phi(x_1) \mid x_0 = x, u_0 = f_s(x)] \\ &\qquad = \min_u\{c(x,u) + E[\phi(x_1) \mid x_0 = x, u_0 = u]\}\,, \end{aligned}$$

taking $f_s(x) = f_{s-1}(x)$ whenever this is possible. By applications of Theorem 6.1, this yields a strict improvement whenever possible. If $\pi_s = \pi_{s-1}$ then the algorithm terminates and π_{s-1} is optimal. Otherwise, return to step (1) with $s := s+1$.

If both the action and state spaces are finite then there are only a finite number of possible stationary policies and so the policy improvement algorithm will find an optimal stationary policy in finitely many iterations. By contrast, the value iteration algorithm can only obtain more and more accurate approximations of λ^*.

Policy improvement in the discounted-cost case.

In the case of strict discounting, the following theorem plays the role of Theorem 6.1. The proof is similar, by repeated substitution of (6.6) into itself.

Theorem 6.4 *Suppose there exists a bounded function G such that for all x and u,*

$$G(x) \le c(x,u) + \beta E[G(x_1) \mid x_0 = x, u_0 = u]. \tag{6.6}$$

Then $G \le F$, where F is the minimal discounted-cost function. This also holds when $\le$ is replaced by $\ge$ and the hypothesis is weakened to: for each x there exists a u such that (6.6) holds when $\le$ is replaced by $\ge$.

The policy improvement algorithm is similar. E.g., step (1) becomes

(1) For a given stationary policy $\pi_{s-1} = f_{s-1}^\infty$ determine G to solve

$$G(x) = c(x, f_{s-1}(x)) + \beta E[G(x_1) \mid x_0 = x, u_0 = f_{s-1}(x)]\,.$$