Dynamic Programming with Applications

Class Notes

René Caldentey
Stern School of Business, New York University

Spring 2011
Preface

These lecture notes are based on the material that my colleague Gustavo Vulcano uses in the Dynamic Programming Ph.D. course that he regularly teaches at the New York University Leonard N. Stern School of Business.

Part of this material is based on the widely used Dynamic Programming and Optimal Control textbook by Dimitri Bertsekas, including a set of lecture notes publicly available in the textbooks website: http://www.athenasc.com/dpbook.html

However, I have added some additional material on Optimal Control for deterministic systems (Chapter 1) and for point processes (Chapter 6). I have also tried to add more applications related to Operations Management.

The booklet is organized in six chapters. We will cover each chapter in a 3-hour lecture except for Chapter 2 where we will spend two 3-hour lectures. The details of each session is presented in the syllabus. I hope that you will find the material useful!
## Contents

1 Deterministic Optimal Control  
1.1 Introduction to Calculus of Variations  
\hspace{10pt} 1.1.1 Abstract Vector Space  
\hspace{10pt} 1.1.2 Classical Calculus of Variations  
\hspace{10pt} 1.1.3 Exercises  
1.2 Continuous-Time Optimal Control  
1.3 Pontryagin Minimum Principle  
\hspace{10pt} 1.3.1 Weak & Strong Extremals  
\hspace{10pt} 1.3.2 Necessary Conditions  
1.4 Deterministic Dynamic Programming  
\hspace{10pt} 1.4.1 Value Function  
\hspace{10pt} 1.4.2 DP’s Partial Differential Equations  
\hspace{10pt} 1.4.3 Feedback Control  
\hspace{10pt} 1.4.4 The Linear-Quadratic Problem  
1.5 Extensions  
\hspace{10pt} 1.5.1 The Method of Characteristics for First-Order PDEs  
\hspace{10pt} 1.5.2 Optimal Control and Myopic Solution  
1.6 Exercises  
1.7 Exercises

2 Discrete Dynamic Programming  
2.1 Discrete-Time Formulation  
\hspace{10pt} 2.1.1 Markov Decision Processes  
2.2 Deterministic DP and the Shortest Path Problem  
\hspace{10pt} 2.2.1 Deterministic finite-state problem  
\hspace{10pt} 2.2.2 Backward and forward DP algorithms
3.1.4 Exercises ................................................................. 113
3.2 Single-Leg Revenue Management ....................................... 114
  3.2.1 System with observable disturbances .......................... 115
  3.2.2 Structure of the value function ................................. 116
  3.2.3 Structure of the optimal policy ................................ 120
  3.2.4 Computational complexity ....................................... 121
  3.2.5 Airlines: Practical implementation ............................ 121
  3.2.6 Exercises ........................................................... 121
3.3 Optimal Stopping and Scheduling Problems ......................... 123
  3.3.1 Optimal stopping problems ..................................... 123
  3.3.2 General stopping problems and the one-step look ahead policy ... 129
  3.3.3 Scheduling problem .............................................. 132
  3.3.4 Exercises ........................................................... 132
4 DP with Imperfect State Information ................................... 135
  4.1 Reduction to the perfect information case ......................... 135
  4.2 Linear-Quadratic Systems and Sufficient Statistics ............. 144
    4.2.1 Linear-Quadratic systems ..................................... 144
    4.2.2 Implementation aspects – Steady-state controller ........ 149
    4.2.3 Sufficient statistics ........................................... 151
    4.2.4 The conditional state distribution recursion ............... 153
  4.3 Sufficient Statistics ................................................ 154
    4.3.1 Conditional state distribution: Review of basics .......... 155
    4.3.2 Finite-state systems .......................................... 157
  4.4 Exercises ........................................................... 163
5 Infinite Horizon Problems .............................................. 167
  5.1 Types of infinite horizon problems ............................... 167
    5.1.1 Preview of infinite horizon results ......................... 168
    5.1.2 Total cost problem formulation .............................. 168
  5.2 Stochastic shortest path problems ................................ 169
    5.2.1 Computational approaches .................................... 175
  5.3 Discounted problems ................................................. 178
  5.4 Average cost-per-stage problems .................................. 182
5.4.1 General setting .................................................... 182
5.4.2 Associated stochastic shortest path (SSP) problem ............... 183
5.4.3 Heuristic argument ................................................ 184
5.4.4 Bellman’s equation .................................................. 186
5.4.5 Computational approaches ....................................... 189

5.5 Semi-Markov Decision Problems ................................... 193
5.5.1 General setting ..................................................... 193
5.5.2 Problem formulation .............................................. 193
5.5.3 Discounted cost problems ........................................ 195
5.5.4 Average cost problems ........................................... 198

5.6 Application: Multi-Armed Bandits .................................. 198

5.7 Exercises ............................................................... 198

6 Point Process Control .................................................. 201
6.1 Basic Definitions ..................................................... 201
6.2 Counting Processes .................................................. 202
6.3 Optimal Intensity Control ........................................... 205
6.3.1 Dynamic Programming for Intensity Control ................. 205
6.4 Applications to Revenue Management ................................ 206
6.4.1 Model Description and HJB Equation ......................... 206
6.4.2 Bounds and Heuristics ........................................... 207

7 Papers and Additional Readings ....................................... 209
Chapter 1

Deterministic Optimal Control

In this chapter, we discuss the basic Dynamic Programming framework in the context of deterministic, continuous-time, continuous-state-space control.

1.1 Introduction to Calculus of Variations

Given a function \( f : \mathcal{X} \to \mathbb{R} \), we are interested in characterizing a solution to

\[
\min_{x \in \mathcal{X}} f(x), \quad [\star]
\]

where \( \mathcal{X} \) is a finite-dimensional space, e.g., in classical calculus \( \mathcal{X} \subseteq \mathbb{R}^n \).

If \( n = 1 \) and \( \mathcal{X} = [a, b] \), then under some smoothness conditions we can characterize solutions to \([\star]\) through a set of necessary conditions.

**Necessary conditions for a minimum at \( x^* \):**

- **Interior point:** \( f'(x^*) = 0, \ f''(x^*) \geq 0, \) and \( a < x^* < b \).

- **Left Boundary:** \( f'(x^*) \geq 0 \) and \( x^* = a \).

- **Right Boundary:** \( f'(x^*) \leq 0 \) and \( x^* = b \).

**Existence:** If \( f \) is continuous on \([a, b]\) then it has a minimum on \([a, b]\).

**Uniqueness:** If \( f \) is strictly convex on \([a, b]\) then it has a unique minimum on \([a, b]\).

1.1.1 Abstract Vector Space

Consider a general optimization problem:

\[
\min_{x \in \mathcal{D}} J(x), \quad [**]
\]

where \( \mathcal{D} \) is a subset of a vector space \( \mathcal{V} \).

We consider functions \( \zeta = \zeta(\varepsilon) : [a, b] \to \mathcal{D} \) such that the composite \( J \circ \zeta \) is differentiable. Suppose that \( x^* \in \mathcal{D} \) and \( J(x^*) \leq J(x) \) for all \( x \in \mathcal{D} \). In addition, let \( \zeta \) such that \( \zeta(\varepsilon^*) = x^* \) then (necessary conditions):
- **Interior point**: \( \frac{d}{d \varepsilon} J(\varepsilon) \bigg|_{\varepsilon = \varepsilon^*} = 0 \), \( \frac{d^2}{d \varepsilon^2} J(\varepsilon) \bigg|_{\varepsilon = \varepsilon^*} \geq 0 \), and \( a < \varepsilon^* < b \).

- **Left Boundary**: \( \frac{d}{d \varepsilon} J(\varepsilon) \bigg|_{\varepsilon = \varepsilon^*} \geq 0 \) and \( \varepsilon^* = a \).

- **Right Boundary**: \( \frac{d}{d \varepsilon} J(\varepsilon) \bigg|_{\varepsilon = \varepsilon^*} \leq 0 \) and \( \varepsilon^* = b \).

How do we use these necessary conditions to identify “good candidates” for \( x^* \)?

### Extremals and Gâteau Variations

**Definition 1.1.1**

Let \((V, \| \cdot \|)\) be a normed linear space and let \( D \subseteq V \).

- We say that a point \( x^* \in D \) is an extremal point for a real-valued function \( J \) on \( D \) if
  \[
  J(x^*) \leq J(x) \quad \text{for all } x \in D \quad \vee \quad J(x^*) \geq J(x) \quad \text{for all } x \in D.
  \]

- A point \( x_0 \in D \) is called a local extremal point for \( J \) if for some \( \varepsilon > 0 \), \( x_0 \) is an extremal point on \( D_{\varepsilon}(x_0) := \{ x \in D : \| x - x_0 \| < \varepsilon \} \).

- A point \( x \in D \) is an internal (radial) point of \( D \) in the direction \( v \in V \) if
  \[
  \exists \varepsilon(v) > 0 \text{ such that } x + \varepsilon v \in D \text{ for all } |\varepsilon| < \varepsilon(v) \quad (0 \leq \varepsilon < \varepsilon(v)).
  \]

- The directional derivative of order \( n \) of \( J \) at \( x \) in the direction \( v \) is denoted by
  \[
  \delta^n J(x; v) = \frac{d^n}{d \varepsilon^n} J(x + \varepsilon v) \bigg|_{\varepsilon = 0}.
  \]

- \( J \) is Gâteau-differentiable at \( x \) if \( x \) is an internal point in the direction \( v \) and \( \delta J(x; v) \) exists for all \( v \in V \).

**Theorem 1.1.1 (Necessary Conditions)** Let \((V, \| \cdot \|)\) be a normed linear space. If \( J \) has a (local) extremal at a point \( x^* \) on \( D \) then \( \delta J(x^*, v) = 0 \) for all \( v \in V \) such that (i) \( x^* \) is an internal point in the direction \( v \) and (ii) \( \delta J(x^*, v) \) exists.

This result is useful if there is “enough” directions \( v \) so that the condition \( \delta J(x^*, v) = 0 \) can determine \( x^* \).

**Problem 1.1.1**

1. Find the extremal points for
  \[ J(y) = \int_a^b y^2(x) \, dx \]
on the domain \( D = \{ y \in C[a, b] : y(a) = \alpha \text{ and } y(b) = \beta \} \).

2. Find the extremal for
  \[ J(P) = \int_a^b P(t) D(P(t)) \, dt \]
on the domain \( D = \{ P \in C[a, b] : \dot{P}(t) \leq \xi \} \).
Extremal with Constraints

Suppose that in a normed linear space \((V, \| \cdot \|)\) we want to characterize extremal points for a real-valued function \(J\) on a domain \(D \subseteq V\). Suppose that the domain is given by the level set \(D := \{ x \in V : G(x) = \psi \}\), where \(G\) is a real-valued function on \(V\) and \(\psi \in \mathbb{R}\).

Let \(x^*\) be a (local) extremal point. We will assume that both \(J\) and \(G\) are defined in a neighborhood of \(x^*\). We pick an arbitrary pair of directions \(v, w\) and define the mapping

\[
F_{v,w}(r, s) := \begin{pmatrix} \rho(r, s) \\ \sigma(r, s) \end{pmatrix} = \begin{pmatrix} J(x^* + rv + sw) \\ G(x^* + rv + sw) \end{pmatrix}
\]

which is well defined in a neighborhood of the origin.

Suppose \(F\) maps a neighborhood of 0 in the \((r, s)\) plane into an neighborhood of \((\rho^*, \sigma^*) := (J(x^*), G(x^*))\) in the \((\rho, \sigma)\) plane. Then \(x^*\) cannot be an extremal point of \(J\).

This condition is assured if \(F\) has an inverse which is continuous at \((\rho^*, \sigma^*)\).

**Theorem 1.1.2** For \(\bar{x} \in \mathbb{R}^n\) and a neighborhood \(N(\bar{x})\), if a vector valued function \(F : N(\bar{x}) \to \mathbb{R}^n\) has continuous first partial derivatives in each component with nonvanishing Jacobian determinant at \(\bar{x}\), then \(F\) provides a continuously invertible mapping between a neighborhood of \(\bar{x}\) and a region containing a full neighborhood of \(F(\bar{x})\).

In our case, \(\bar{x} = 0\) and the Jacobian matrix of \(F\) is given by

\[
\nabla F(0, 0) = \begin{pmatrix} \delta J(x^*; v) & \delta J(x^*; w) \\ \delta G(x^*; v) & \delta G(x^*; w) \end{pmatrix}
\]

Then if \(|\nabla F(0, 0)| \neq 0\) then \(x^*\) cannot be an extremal point for \(J\) when constraint to the level set defined by \(G(x^*)\).
Definition 1.1.2 In a normed linear space $(\mathcal{V}, \| \cdot \|)$, the Gâteau variations $\delta J(x, v)$ of a real valued function $J$ are said to be weakly continuous at $x^* \in \mathcal{V}$ if for each $v \in \mathcal{V}$ $\delta J(x; v) \to \delta J(x^*; v)$ as $x \to x^*$.

Theorem 1.1.3 (Lagrange) In a normed linear space $(\mathcal{V}, \| \cdot \|)$, if a real valued functions $J$ and $G$ are defined in a neighborhood of $x^*$, a (local) extremal point for $J$ constrained by the level set $G(x^*)$, and have there weakly continuous Gâteau variations, then either

a) $\delta G(x^*; w) = 0$, for all $w \in \mathcal{V}$, or

b) there exists a constant $\lambda \in \mathbb{R}$ such that $\delta J(x^*, v) = \lambda \delta G(x^*; v)$, for all $v \in \mathcal{V}$.

Problem 1.1.2 Find the extremal for

$$J(P) = \int_0^T P(t) \, D(P(t)) \, dt$$

on the domain $\mathcal{D} = \{ P \in C[0, T] : \int_0^T D(P(t)) \, dt = I \}$.

1.1.2 Classical Calculus of Variations

Historical Background

The theory of Calculus of Variations has been the “classic” approach to solve dynamic optimization problems, dating back to the late 17th century. It started with the Brachistochrone problem proposed by Johann Bernoulli in 1696: Find the planar curve which would provide the faster time of transit to a particle sliding down it under the action of gravity (see Figure 1.1.2). Five solutions were proposed by Jakob Bernoulli (Johann’s brother), Newton, Euler, Leibniz, and L’Hôpital. Another classical example of the method of calculus of variations is the Geodesic problems: Find the shortest path in a given domain connecting two points of it (e.g., the shortest path in a sphere).

Figure 1.1.2: The Brachistochrone problem: Find the curve which would provide the faster time of transit to a particle sliding down it from Point A to Point B under the action of gravity.
More generally, calculus of variations problems involve finding (possibly multidimensional) curves $x(t)$ with certain optimality properties. In general, the calculus of variations approach requires the differentiability of the functions that enter the problem in order to get “interior solutions”.

**The Simplest Problem in Calculus of Variations**

$$J(x) = \int_a^b L(t, x(t), \dot{x}(t)) \, dt,$$

where $\dot{x}(t) = \frac{d}{dt} x(t)$. The *variational integrand* is assumed to be smooth enough (e.g., at least $C^2$).

**Example 1.1.1**

- Geodesic: $L = \sqrt{1 + \dot{x}^2}$
- Brachistochrone: $L = \sqrt{\frac{1 + \dot{x}^2}{x - a}}$
- Minimal Surface of Revolution: $L = x \sqrt{1 + \dot{x}^2}$.

**Admissible Solutions:** A function $x(t)$ is called *piecewise* $C^n$ on $[a, b]$, if $x(t)$ is $C^{n-1}$ on $[a, b]$ and $x^{(n)}(t)$ is piecewise continuous on $[a, b]$, i.e., continuous except on a finite number of points. We denote by $\mathcal{H}[a,b]$ the vector space of all real-valued piecewise $C^1$ function on $[a, b]$ and by $\mathcal{H}_e[a,b]$ the subspace of $\mathcal{H}[a,b]$ such that $x(a) = x_a$ and $x(b) = x_b$ for all $x \in \mathcal{H}_e[a,b]$.

Problem: $\min_{x \in \mathcal{H}_e[a,b]} J(x)$.

**Admissible Variations or Test Functions:** Let $\mathcal{Y}[a, b] \subseteq \mathcal{H}[a, b]$ be the subspace of piecewise $C^1$ functions $y$ such that

$y(a) = y(b) = 0$.

We note that for $x \in \mathcal{H}_e[a,b]$, $y \in \mathcal{Y}[a, b]$, and $\varepsilon \in \mathbb{R}$, the function $x + \varepsilon y \in \mathcal{H}_e[a,b]$.

**Theorem 1.1.4** Let $J$ have a minimum on $\mathcal{H}_e[a,b]$ at $x^*$. Then

$$L_{\ddot{x}} - \int_a^t L_x \, d\tau = \text{constant} \quad \text{for all } t \in [a,b]. \quad (1.1.1)$$

A function $x^*(t)$ satisfying (1.1.1) is called *extremal*.

**Corollary 1.1.1 (Euler’s Equation)** Every extremal $x^*$ satisfies the differential equation

$$L_x = \frac{d}{dt} L_{\dot{x}}.$$

**Problem 1.1.3 (Production-Inventory Control)**

Consider a firm that operates according to a make-to-stock policy during a planning horizon $[0, T]$. The company faces an exogenous and deterministic demand with intensity $\lambda(t)$. Production is costly; if the firm chooses a production rate $\mu$ at time $t$ then the instantaneous production cost rate is $c(t, \mu)$. In
addition, there are holding and backordering costs. We denote by \( h(t, I) \) the holding/backordering cost rate if the inventory position at time \( t \) is \( I \). We suppose that the company starts with an initial inventory \( I_0 \) and tries to minimize total operating costs during the planning horizon of length \( T > 0 \) subject to the requirement that the final inventory position at time \( T \) is \( I_T \).

a) Formulate the optimization problem as a calculus of variations problem.

b) What is Euler's equation?

**Sufficient Conditions: Weierstrass Method**

Suppose that \( x^\ast \) is an extremal for

\[
J(x) = \int_a^b f(t, x(t), \dot{x}(t)) \, dt := \int_a^b f[x(t)] \, dt
\]
on \( D = \{ x \in C^1[a, b] : x(a) = x^\ast(a); x(b) = x^\ast(b) \} \). Let \( \tilde{x}(t) \in D \) be an arbitrary feasible solution. For each \( \tau \in (a, b) \) we define the function \( \Psi(t; \tau) \) on \((a, \tau)\) such that \( \Psi(t; \tau) \) is an extremal function for \( f \) on \((a, \tau)\) whose graph joins \((a, x^\ast(a))\) to \((\tau, \tilde{x}(\tau))\) and such that \( \Psi(t; b) = x^\ast(t) \).

We define

\[
\sigma(\tau) := -\int_a^\tau f[\Psi(t; \tau)] \, dt - \int_\tau^b f[\tilde{x}(t)] \, dt,
\]
which has the following properties:

\[
\sigma(a) = -\int_a^b f[\tilde{x}(t)] \, dt = -J(\tilde{x}) \quad \text{and} \quad \sigma(b) = -\int_a^b f[\Psi(t, b)] \, dt = -J(x^\ast).
\]

Therefore, we have that

\[
J(\tilde{x}) - J(x^\ast) = \sigma(b) - \sigma(a) = \int_a^b \dot{\sigma}(\tau) \, d\tau,
\]
so that a sufficient condition for the optimality of \( x^\ast \) is \( \dot{\sigma}(\tau) \geq 0 \). That is,

**Weierstrass' formula**

\[
\dot{\sigma}(\tau) := E(\tau, \tilde{x}(\tau), \dot{x}(\tau), \dot{\Psi}(\tau; \tau), \dot{\tilde{x}}(\tau)) \\
= f[\tilde{x}(\tau)] - f(\tau, \tilde{x}(\tau), \dot{x}(\tau), \dot{\Psi}(\tau; \tau)) - f(t, \tilde{x}(\tau), \dot{x}(\tau), \dot{\Psi}(\tau; \tau)) \cdot (\dot{\tilde{x}}(\tau) - \dot{\Psi}(\tau; \tau)) \geq 0
\]
1.1.3 Exercises

Exercise 1.1.1 (Convexity and Euler’s Equation) Let $\mathcal{V}$ be a linear vector space and $\mathcal{D}$ a subset of $\mathcal{V}$. A real-valued function $f$ defined on $\mathcal{D}$ is said to be [strictly] convex on $\mathcal{D}$ if

$$f(y + v) - f(y) \geq \delta f(y; v) \quad \text{for all } y, y + v \in \mathcal{D},$$

[with equality if and only if $v = 0$]. Where $\delta f(y; v)$ is the first Gâteau variation of $f$ at $y$ on the direction $v$.

a) Prove the following: If $f$ is [strictly] convex on $\mathcal{D}$ then each $x^* \in \mathcal{D}$ for which $\delta f(x^*; y) = 0$ for all $x^* + y \in \mathcal{D}$ minimizes $f$ on $\mathcal{D}$ [uniquely].

Let $f = f(x, y, z)$ be a real value function on $[a, b] \times \mathbb{R}^2$. Assume that $f$ and the partial derivatives $f_y$ and $f_z$ are defined and continuous on $S$. For all $y \in C^1[a, b]$ we define the integral function

$$F(y) = \int_a^b f(x, y(x), y'(x)) \, dx : = \int_a^b f[y(x)] \, dx,$$

where $f[y(x)] = f(x, y(x), y'(x))$.

b) Prove that the first Gâteau variation of $F$ is given by

$$\delta F(y; v) = \int_a^b \left( f'_y[y(x)] v(x) + f'_z[y(x)] v'(x) \right) \, dx.$$

c) Let $D$ be a domain in $\mathbb{R}^2$. For two arbitrary real numbers $\alpha$ and $\beta$ define

$$D^{\alpha, \beta}[a, b] = \{ y \in C^1[a, b] : y(a) = \alpha, y(b) = \beta, \text{ and } (y(x), y'(x)) \in D \forall x \in [a, b] \}.$$

Prove that if $f(x, y, z)$ is convex on $[a, b] \times D$ then

1. $F(y)$ defined above is convex on $\mathcal{D}$ and
2. each $y \in \mathcal{D}$ for which
   $$\frac{d}{dx} f_z[y(x)] = f_y[y(x)] \quad \text{[\ast]}$$
   
   on $(a, b)$ satisfies $\delta F(y, v) = 0$ for all $y + v \in \mathcal{D}$.

Conclude that such a $y \in \mathcal{D}$ that satisfies [\ast] minimizes $F$ on $\mathcal{D}$. That is, extremal solutions are minimizers.

Exercise 1.1.2 (du Bois-Reymond’s Lemma) The proof of Euler’s equation uses du Bois-Reymond’s Lemma:

If $h \in C[a, b]$ and $\int_a^b h(x)v'(x) \, dx = 0$ for all $v \in D_0 = \{ v \in C^1[a, b] : v(a) = v(b) = 0 \}$

then $h =$constant on $[a, b]$. Using this lemma prove the more general results.
a) If \( g, h \in C[a, b] \) and \( \int_a^b [g(x)v(x) + h(x)v'(x)] \, dx = 0 \)

for all \( v \in \mathcal{D}_0 = \{ v \in C^1[a, b] : v(a) = v(b) = 0 \} \)

then \( h \in C^1[a, b] \) and \( h' = g \).

b) If \( h \in C[a, b] \) and for some \( m = 1, 2, \ldots \) we have \( \int_a^b h(x)v^{(m)}(x) \, dx = 0 \)

for all \( v \in \mathcal{D}^{(m)}_0 = \{ v \in C^m[a, b] : v^{(k)}(a) = v^{(k)}(b) = 0, k = 0, 1, 2, \ldots, m - 1 \} \)

then on \([a, b]\), \( h \) is a polynomial of degree \( \leq m - 1 \).

Exercise 1.1.3 Suppose you have inherited a large sum \( S \) and plan to spend it so as to maximize your discounted cumulative utility for the next \( T \) units of time. Let \( u(t) \) be the amount that you expend on period \( t \) and let \( \sqrt{u(t)} \) the the instantaneous utility rate that you receive at time \( t \). Let \( \beta \) be the discount factor that you use to discount future utility, i.e., the discounted value of expending \( u \) at time \( t \) is equal to \( \exp(-\beta t) \sqrt{u} \). Let \( \alpha \) be the risk-free interest rate available on the market, i.e., one dollar today is equivalent to \( \exp(\alpha t) \) dollars \( t \) units of time in the future.

a) Formulate the control problem that maximizes the discounted cumulative utility given all necessary constraints.

b) Find the optimal expenditure rate \( \{u(t)\} \) for all \( t \in [0, T] \).

Exercise 1.1.4 (Production-Inventory Problem) Consider a make-to-stock manufacturing facility producing a single type of product. Initial inventory at time \( t = 0 \) is \( I_0 \). Demand rate for the next selling season \([0, T]\) is known and equal to \( \lambda(t) \) \( t \in [0, T] \). We denote by \( \mu(t) \) the production rate and by \( I(t) \) the inventory position. Suppose that due to poor inventory management there is a fixed proportion \( \alpha \) of inventory that is lost per unit time. Thus, at time \( t \) the inventory \( I(t) \) increases at a rate \( \mu(t) \) and decreases at a rate \( \lambda(t) + \alpha I(t) \).

Suppose the company has set target values for the inventory and production rate during \([0, T]\). Let \( \bar{I} \) and \( \bar{P} \) be these target values, respectively. Deviation from these values are costly, and the company uses the following cost function \( C(I, P) \) to evaluate a production-inventory strategy \( (P, I) \):

\[
C(I, P) = \int_0^T \left[ \beta^2 (\bar{I} - I(t))^2 + (\bar{P} - P(t))^2 \right] \, dt.
\]

The objective of the company is to find and optimal production-inventory strategy that minimizes the cost function subject to the additional condition that \( I(T) = \bar{I} \).

a) Rewrite the cost function \( C(I, P) \) as a function of the inventory position and its first derivative only.

b) Find the optimal production-inventory strategy.
1.2 Continuous-Time Optimal Control

The Optimal Control problem that we study in this section, and in particular the optimality conditions that we derive (HJB equation and Pontryagin Minimum principle) will provide us with an alternative and powerful method to solve the variational problems discussed in the previous section. This new method is not only useful as a solution technique but also as a insightful methodology to understand how dynamic programming works.

Compared to the method of Calculus of Variation, Optimal Control theory is a more modern and flexible approach that requires less stringent differentiability conditions and can handle corner solutions. In fact, calculus of variations problems can be reformulated as optimal control problems, as we show lated in this section.

The first, and most fundamental, step in the derivation of these new solution techniques is the notion of a System Equation:

- **System Equation** (also called equation of motion or system dynamics):
  \[
  \dot{x}(t) = f(t, x(t), u(t)), \quad 0 \leq t \leq T, \quad x(0) : \text{given},
  \]
  i.e.
  \[
  \frac{dx_i(t)}{dt} = f_i(t, x(t), u(t)), \quad i = 1, \ldots, n.
  \]

  where:
  - \(x(t) \in \mathbb{R}^n\) is the state vector at time \(t\),
  - \(\dot{x}(t)\) is the gradient of \(x(t)\) with respect to \(t\),
  - \(u(t) \in U \subseteq \mathbb{R}^m\) is the control vector at time \(t\),
  - \(T\) is the terminal time.

- **Assumptions:**
  - An admissible control trajectory is a piecewise continuous function \(u(t) \in U, \forall t \in [0, T]\), that does not involve an infinite value of \(u(t)\) (i.e., all jumps are of finite size).
    \(U\) could be a bounded control set. For instance, \(U\) could be a compact set such as \(U = [0, 1]\), so that corner solutions (boundary solutions) could be admitted. When this feature is combined with jump discontinuities on the control path, an interesting phenomenon called a bang-bang solution may result, where the control alternates between corners.
  - An admissible state trajectory \(x(t)\) is continuous, but it could have a finite number of corners; i.e., it must be piecewise differentiable. A sharp point on the state trajectory occurs at a time when the control trajectory makes a jump.
    Like admissible control paths, admissible state paths must have a finite \(x(t)\) value for every \(t \in [0, T]\). See Figure 1.2.1 for an illustration of a control path and the associated state path.
  - The control trajectory \(\{u(t) \mid t \in [0, T]\}\) uniquely determines \(\{x^u(t) \mid t \in [0, T]\}\). We will drop the superscript \(u\) from now on, but this dependence should be clear. In a more rigorous treatment, the issue of existence and uniqueness of the solution should be addressed more carefully.
• **Objective:** Find an admissible policy (control trajectory) \( \{ u(t) \mid t \in [0, T] \} \) and corresponding state trajectory that optimizes a given functional \( J \) of the state \( x = (x_t : 0 \leq t \leq T) \). The following are some common formulations for the functional \( J \) and the associated optimal control problem.

**LAGRANGE PROBLEM:**
\[
\min_{u \in \mathcal{U}} J(x) = \int_0^T g(t, x(t), u(t)) \, dt
\]
subject to
\[
\dot{x}(t) = f(t, x(t), u(t)), \quad x(0) = x_0 \quad \text{(system dynamics)}
\]
\[
\phi(x(T)) = 0 \quad \text{(boundary conditions)}.
\]

**MAYER PROBLEM:**
\[
\min_{u \in \mathcal{U}} h(x(T))
\]
subject to
\[
\dot{x}(t) = f(t, x(t), u(t)), \quad x(t_0) = x_0 \quad \text{(system dynamics)}
\]
\[
\phi(x(T)) = 0 \quad k = 2, \ldots, k \quad \text{(boundary conditions)}.
\]

**BOLZA PROBLEM:**
\[
\min_{u \in \mathcal{U}} h(x(T)) + \int_0^T g(t, x(t), u(t)) \, dt.
\]
subject to
\[
\dot{x}(t) = f(t, x(t), u(t)), \quad x(t_0) = x_0 \quad \text{(system dynamics)}
\]
\[
\phi(x(T)) = 0 \quad k = 2, \ldots, k \quad \text{(boundary conditions)}.
\]

The functions \( f, h, g \) and \( \phi \) are normally assumed to be continuously differentiable with respect to \( x \); and \( f, g \) are continuous with respect to \( t \) and \( u \).
Problem 1.2.1 Show that all three versions of the optimal control problem are equivalent.

Example 1.2.1 (Motion Control) A unit mass moves on a line under the influence of a force $u$. Here, $u =$ force $=$ acceleration. (Recall from physics that force $=$ mass $\times$ acceleration, with mass $= 1$ in this case).

- **State:** $x(t) = (x_1(t), x_2(t))$, where $x_1$ represents position and $x_2$ represents velocity.
- **Problem:** From a given initial $(x_1(0), x_2(0))$, bring the mass near a given final position-velocity pair $(\bar{x}_1, \bar{x}_2)$ at time $T$; in the sense that it minimizes
  $$|x_1(T) - \bar{x}_1|^2 + |x_2(T) - \bar{x}_2|^2,$$
  such that $|u(t)| \leq 1, \forall t \in [0, T]$.
- **System Equation:**
  $$\dot{x}_1(t) = x_2(t)$$
  $$\dot{x}_2(t) = u(t)$$
- **Costs:**
  $$h(x(T)) = (x_1(T) - \bar{x}_1)^2 + (x_2(T) - \bar{x}_2)^2$$
  $$g(x(t), u(t)) = 0, \forall t \in [0, T].$$

Example 1.2.2 (Resource Allocation) A producer with production rate $x(t)$ at time $t$ may allocate a portion $u(t) \in [0, 1]$ of her production rate to reinvestment (i.e., to increase the production rate) and $[1 - u(t)]$ to produce a storable good. Assume a terminal cost $h(x(T)) = 0$.

- **System Equation:**
  $$\dot{x}_1(t) = \gamma u(t)x(t),$$
  where $\gamma > 0$ is the reinvestment benefit, $u(t) \in [0, 1]$.
- **Problem:** The producer wants to maximize the total amount of product stored
  $$\max_{u(t) \in [0, 1]} \int_0^T (1 - u(t))x(t)dt$$
  Assume $x(0)$ is given.

Example 1.2.3 (An application of Calculus of Variations) Find a curve from a given point to a given vertical line that has minimum length. (Intuitively, this should be a straight line) Figure 1.2.2 illustrates the formulation as an infinite sum of infinitely small hypotenuses $\alpha$.

- **The problem in terms of calculus of variations is:**
  $$\min \int_0^T \sqrt{1 + (\dot{x}(t))^2}dt$$
  s.t. $x(0) = \alpha.$
CHAPTER 1. DETERMINISTIC OPTIMAL CONTROL

Figure 1.2.2: Problem of finding a curve of minimum length from a given point to a given line, and its formulation as an optimal control problem.

- The corresponding optimal control problem is:

\[
\min_{u(t)} \int_0^T \sqrt{1 + (u(t))^2} \, dt \\
\text{s.t. } \dot{x}(t) = u(t) \\
x(0) = \alpha
\]

1.3 Pontryagin Minimum Principle

1.3.1 Weak & Strong Extremals

Let \( \mathcal{H}[a,b] \) be a subset of piecewise right-continuous function with left-limit (càdlàg). We define on \( \mathcal{H}[a,b] \) two norms

\[
\|x\| = \sup_{t \in [a,b]} \{|x(t)|\} \quad \text{and} \quad \|x\|_1 = \|x\| + \|\dot{x}\|.
\]

A set \( \{x \in \mathcal{H}[a,b] : \|x - x^*\|_1 < \epsilon\} \) is called a weak neighborhood of \( x^* \). A solution \( x^* \) is called a weak solution if \( J(x^*) \leq J(x) \) for all \( x \) in a weak neighborhood containing \( x^* \).

A set \( \{x \in \mathcal{H}[a,b] : \|x - x^*\| < \epsilon\} \) is called a strong neighborhood of \( x^* \). A solution \( x^* \) is called a strong solution if \( J(x^*) \leq J(x) \) for all \( x \) in a strong neighborhood containing \( x^* \).

Example 1.3.1

\[
\min_x J(x) = \int_{-1}^1 (x(t) - \text{sign}(t))^2 \, dt + \sum_{t \in [-1,1]} (x(t) - x(t^-))^2,
\]

where \( x(t^-) = \lim_{\tau \uparrow t} x(\tau) \).
1.3.2 Necessary Conditions

Given a control $u \in U$ with corresponding trajectory $x(t)$, we consider the following family of variations:

For a fixed direction $v \in U$, $\tau \in [0, T]$, and $\eta > 0$ small, we defined the "strong" variation $\xi$ of $u(t)$ in the direction $v$ by the function

$$\xi : 0 \leq \epsilon \leq \eta \rightarrow U$$

$$\epsilon \rightarrow \xi(\epsilon) = u^\epsilon,$$

where

$$u^\epsilon(t) = \begin{cases} v & \text{if } t \in (\tau - \epsilon, \tau] \\ u(t) & \text{if } t \in [0, T] \cap (\tau - \epsilon, \tau]. \end{cases}$$

![Strong Variation vs Weak Variation](image)

Figure 1.3.1: An example of strong and weak variations

**Lemma 1.3.1** For a real variable $\epsilon$, let $x^\epsilon(t)$ be the solution of $\dot{x}^\epsilon(t) = f(t, x^\epsilon(t), u(t))$ on $[0,T]$ with initial condition $x^\epsilon(0) = x(0) + \epsilon y + o(\epsilon)$. Then,

$$x^\epsilon(t) = x(t) + \epsilon \delta(t) + o(t, \epsilon),$$

where $\delta(t)$ is the solution of

$$\dot{\delta}(t) = f_x(t, x(t), u(t)) \delta(t), \quad t \in [0, T] \text{ and } \delta(0) = y.$$

**Lemma 1.3.2** If $x^\epsilon$ are solutions to $\dot{x}^\epsilon(t) = f(t, x^\epsilon(t), u^\epsilon(t))$ with the same initial condition $x^\epsilon(0) = x_0$ then

$$x^\epsilon(t) = x(t) + \epsilon \delta(t) + o(t, \epsilon),$$

where $\delta(t)$ solves

$$\delta(t) = \begin{cases} 0 & \text{if } 0 \leq t < \tau \\ f(\tau, x(\tau), v) - f(\tau, x(\tau), u(\tau)) + \int_{\tau}^{t} f_x(s, x(s), u(s)) \delta(s) \, ds & \text{if } \tau \leq t \leq T. \end{cases}$$
Theorem 1.3.1 (Pontryagin Principle For Free Terminal Conditions)

- **Mayer’s formulation:** Let \( P(t) \) be the solution of

\[
\dot{P}(t) = -P(t) f_x(t, x(t), u(t)), \quad P(t_1) = -\phi_x(x(T)).
\]

A necessary condition for optimality of a control \( u \) is that

\[
P(t) [f(t, x(t), v) - f(t, x(t), u(t))] \leq 0
\]

for each \( v \in U \) and \( t \in (0, T) \).

- **Lagrange’s formulation:** We define the Hamiltonian \( H \) as

\[
H(t, x, u) := P(t) f(t, x, u) - L(t, x, u).
\]

Where \( P(t) \) solves

\[
\dot{P}(t) = -\frac{\partial}{\partial x} H(t, x, u)
\]

with boundary condition \( P(T) = 0 \). A necessary condition for a control \( u \) to be optimal is

\[
H(t, x(t), v) - H(t, x(t), u(t)) \leq 0 \quad \text{for all} \ v \in U, \ t \in [0, T].
\]

Theorem 1.3.2 (Pontryagin Principle with Terminal Conditions)

- **Mayer’s formulation:** Let \( P(t) \) be the solution of

\[
\dot{P}'(t) = -P'(t) f_x(t, x(t), u(t)), \quad P(t_1) = -\lambda \phi_x(T, x(T)).
\]

A necessary condition for optimality of a control \( u \in U \) is that there exists \( \lambda \), a nonzero \( k \)-dimensional vector with \( \lambda_1 \leq 0 \), such that

\[
P(t)' [f(t, x(t), v) - f(t, x(t), u(t))] \leq 0
\]

\[
P(T)' f(T, x(T), u(T)) = -\lambda \phi_x(T, x(T)).
\]

Problem 1.3.1 Solve

\[
\min_u \int_0^T (u(t) - 1)x(t) \, dt,
\]

subject to \( \dot{x}(t) = \gamma u(t) x(t) \quad x_0 > 0, \)

\( 0 \leq u(t) \leq 1, \quad \text{for all} \ t \in [0, T]. \)
1.4 Deterministic Dynamic Programming

1.4.1 Value Function

Consider the following optimal control problem in Mayer’s form:

\[
V(t_0, x_0) = \inf_{u \in \mathcal{U}} J(t_1, x(t_1)) \tag{1.4.1}
\]
subject to \[\dot{x}(t) = f(t, x(t), u(t)), \quad x(t_0) = x_0 \quad \text{(state dynamics)} \tag{1.4.2}\]
\[ (t_1, x(t_1)) \in M \quad \text{(boundary conditions).} \tag{1.4.3} \]

The terminal set \(M\) is a closed subset of \(\mathbb{R}^{n+1}\). The admissible control set \(\mathcal{U}\) is assumed to be the set of piecewise continuous function on \([t_0, t_1]\). The performance function \(J\) is assumed to be \(C^1\).

The function \(V(\cdot, \cdot)\) is called the value function and we shall use the convention \(V(t_0, x_0) = \infty\) if the control problem above admits no feasible solution. We will denote by \(\mathcal{U}(x_0, t_0)\), the set of feasible controls with initial condition \((x_0, t_0)\), that is, the set of control \(u\) such that the corresponding trajectory \(x\) satisfies \(x(t_1) \in M\).

**Remark 1.4.1** For notational convenience, in this section the time horizon is denoted by the interval \([t_0, t_1]\) instead of \([0, T]\).

**Proposition 1.4.1** Let \(u(t) \in \mathcal{U}(x_0, t_0)\) be a feasible control and \(x(t)\) the corresponding trajectory. Then, for any \(t_0 \leq \tau_1 \leq \tau_2 \leq t_1\), \(V(\tau_1, x(\tau_1)) \leq V(\tau_2, x(\tau_2))\). That is, the value function is a nondecreasing function along any feasible trajectory.

Proof:

**Corollary 1.4.1** The value function evaluated along any optimal trajectory is constant.

Proof: Let \(u^*\) be an optimal control with corresponding trajectory \(x^*\). Then \(V(t_0, x_0) = J(t_1, x^*(t_1))\). In addition, for any \(t \in [t_0, t_1]\) \(u^*\) is a feasible control starting at \((t, x^*(t))\) and so \(V(t, x^*(t)) \leq J(t_1, x^*(t_1))\). Finally by Proposition 1.4.1 \(V(t_0, x_0) \leq V(t, x^*(t))\) so we conclude \(V(t, x^*(t)) = V(t_0, x_0)\) for all \(t \in [t_0, t_1]\). 

According to the previous results a necessary condition for optimality is that the value function is constant along the optimal trajectory. The following result provides a sufficient condition.
Theorem 1.4.1  Let \( W(s,y) \) be an extended real valued function defined on \( \mathbb{R}^{n+1} \) such that \( W(s,y) = J(s,y) \) for all \((s,y) \in M\). Given an initial condition \((t_0, x_0)\), suppose that for any feasible trajectory \( x(t) \), the function \( W(t,x(t)) \) is finite and nondecreasing on \([t_0,t_1]\). If \( u^* \) is a feasible control with corresponding trajectory \( x^* \) such that \( W(t,x^*(t)) \) is constant then \( u^* \) is optimal and \( V(t_0, x_0) = W(t_0, x_0) \).

Proof: For any feasible trajectory \( x \), \( W(t_0, x_0) \leq W(t_1, x(t_1)) = J(t_1, x(t_1)) \). On the other hand, for \( x^* \), \( W(t_0, x_0) = W(t_1, x^*(t_1)) = J(t_1, x^*(t_1)). \)

Corollary 1.4.2  Let \( u^* \) be an optimal control with corresponding feasible trajectory \( x^* \). Then the restriction of \( u^* \) to \([t,t_1]\) is an optimal for the control problem with initial condition \((t,x^*(t))\).

In many applications, the control problem is given in its Lagrange form

\[
V(t_0, x_0) = \inf_{u \in \mathcal{U}(x_0,t_0)} \int_{t_0}^{t_1} L(t, x(t), u(t)) \, dt \tag{1.4.4}
\]
subject to \( \dot{x}(t) = f(t, x(t), u(t)), \ x(t_0) = x_0 \). \tag{1.4.5}

In this case, the following result is the analogue to Proposition 1.4.1.

Theorem 1.4.2 (Bellman’s Principle of Optimality).  Consider an optimal control problem in Lagrange form. For any \( u \in \mathcal{U}(s,y) \) and its corresponding trajectory \( x \)

\[
V(s,y) \leq \int_{s}^{\tau} L(t, x(t), u(t)) \, dt + V(\tau, x(\tau)).
\]

Proof: Given \( u \in \mathcal{U}(s,y) \), let \( \tilde{u} \in \mathcal{U}(\tau, x(\tau)) \) be arbitrary. Define

\[
\tilde{u}(t) = \begin{cases} 
  u(t) & s \leq t \leq \tau \\
  \tilde{u}(t) & \tau \leq t \leq t_1.
\end{cases}
\]

Thus, \( \tilde{u} \in \mathcal{U}(s,y) \) so that

\[
V(s,y) \leq \int_{s}^{t_1} L(t, \tilde{x}(t), \tilde{u}(t)) \, dt = \int_{s}^{\tau} L(t, x(t), u(t)) \, dt + \int_{\tau}^{t_1} L(t, \tilde{x}(t), \tilde{u}(t)) \, dt. \tag{1.4.6}
\]

Since the inequality holds for any \( \tilde{u} \in \mathcal{U}(\tau, x(\tau)) \) we conclude

\[
V(s,y) \leq \int_{s}^{\tau} L(t, x(t), u(t)) \, dt + V(\tau, x(\tau)). \]

Although the conditions given by Theorem 1.4.1 are sufficient, they do not provide a concrete way to construct an optimal solution. In the next section, we will provide a direct method to compute the value function.
1.4.2 DP’s Partial Differential Equations

Define $Q_0$ the reachable set as

$$Q_0 = \{(s, y) \in \mathbb{R}^{n+1} : U(s, y) \neq \emptyset\}.$$ 

This set define the collection of initial conditions for which the optimal control problem is feasible.

**Theorem 1.4.3** Let $(s, y)$ be any interior point of $Q_0$ at which $V(s, y)$ is differentiable. Then $V(s, y)$ satisfies

$$V_s(s, y) + V_y(s, y) f(s, y, v) \geq 0 \quad \text{for all } v \in U.$$ 

If there is an optimal $u^* \in U(s, y)$, then the PDE

$$\min_{v \in U} \{V_s(s, y) + V_y(s, y) f(s, y, v)\} = 0$$

is satisfied and the minimum is achieved by the right limit $u^*(s)^+$ of the optimal control at $s$.

**Proof:** Pick any $v \in U$ and let $x_v(t)$ be the corresponding trajectory for $s \leq t \leq s + \epsilon, \epsilon > 0$ small. Given the initial condition $(s, y)$, we define the feasible control $u_\epsilon$ as follows

$$u_\epsilon(t) = \begin{cases} 
  v & s \leq t \leq s + \epsilon \\
  \tilde{u}(t) & s + \epsilon \leq t \leq t_1.
\end{cases}$$

Where $\tilde{u} \in U(s + \epsilon, x_v(s + \epsilon))$ is arbitrary. Note that for $\epsilon$ small $(s + \epsilon, x_v(s + \epsilon)) \in Q_0$ and so $u_\epsilon \in U(s,y)$. We denote by $x_\epsilon(t)$ the corresponding trajectory. By proposition (1.5.1), $V(t, x_\epsilon(t))$ is nondecreasing, hence,

$$D^+ V(t, x_\epsilon(t)) := \lim_{h \downarrow 0} \frac{V(t + h, x_\epsilon(t + h)) - V(t, x_\epsilon(t))}{h} \geq 0$$

for any $t$ at which the limit exists, in particular $t = s$. Thus, from the chain rule we get

$$D^+ V(s, x_\epsilon(s)) = V_s(s, y) + V_y(s, y) D^+ x_\epsilon(s) = V_s(s, y) + V_y(s, y) f(s, y, v).$$

The equalities use the identity $x_\epsilon(s) = y$ and the system dynamic equation $D^+ x_\epsilon(t) = f(t, x_\epsilon, u_\epsilon(t)^+)$. If $u^* \in U(s, y)$ is an optimal control with trajectory $x^*$ then corollary 1.4.1 implies $V(t, x^*(t)) = J(t_1, x^*(t_1))$ for all $t \in [s, t_1]$, so differentiating (from the right) this equality at $t = 2$ we conclude

$$V_s(s, y) + V_y(s, y) f(s, y, u^*(s)^+) = 0.$$  

**Corollary 1.4.3 (Hamilton-Jacobi-Bellman equation (HJB))** For a control problem given in Lagrange form (1.4.4)-(1.4.5), the value function at a point $(s, y) \in \text{int}(Q_0)$ satisfies

$$V_s(y, s) + V_y(s, y) f(s, y, v) + L(s, y, v) \geq 0 \quad \text{for all } v \in U.$$ 

If there exists an optimal control $u^*$ then the PDE

$$\min_{v \in U} \{V_s(y, s) + V_y(s, y) f(s, y, v) + L(s, y, v)\} = 0$$

is satisfied and the minimum is achieved by the right limit $u^*(s)^+$ of the optimal control at $s$. 

---

23
In many applications, instead of solving the HJB equation a candidate for the value function is identified, say by inspection. It is important to be able to decide whether or not the proposed solution is in fact optimal.

**Theorem 1.4.4 (Verification Theorem)** Let $W(s, y)$ be a $C^1$ solution to the partial differential equation

$$ \min_{v \in U} \{V_s(s, y) + V_y(s, y) f(s, y, v)\} = 0 $$

with boundary condition $W(s, y) = J(s, y)$ for all $(s, y) \in M$. Let $(t_0, x_0) \in Q_0$, $u \in U(t_0, x_0)$ and $x$ the corresponding trajectory. Then, $W(t, x(t))$ is nondecreasing on $t$. If $u^*$ is a control in $U(t_0, x_0)$ defined on $[t_0, t_1^*]$ with corresponding trajectory $x^*$ such that for any $t \in [t_0, t_1^*]$

$$ W_s(t, x^*(t)) + W_y(t, x^*(t)) f(t, x^*(t), u^*(t)) = 0 $$

then $u^*$ is an optimal control in $calU(t_0, x_0)$ and $V(s, y) = W(s, y)$.

**Example 1.4.1**

$$ \min_{\|u\| \leq 1} J(t_0, x_0, u) = \frac{1}{2}(x(\tau))^2 $$

subject to $\dot{x}(t) = u(t), \; x(t_0) = x_0$

where $\|u\| = \max_{0 \leq t \leq \tau}\{|u(t)|\}$. The HJB equation is $\min_{\|u\| \leq 1} \{V_t(t, x) + V_x(t, x) u\} = 0$ with boundary condition $V(\tau, x) = \frac{1}{2}x^2$. We can solve this problem by inspection. Since the only cost is associated to the terminal state $x(\tau)$, and optimal control will try to make $x(\tau)$ as close to zero as possible, i.e.,

$$ u^*(t, x) = -\text{sgn}(x) = \begin{cases} 
1 & x < 0 \\
0 & x = 0 \\
-1 & x > 0.
\end{cases} \text{ (Bang-Bang policy)} $$

We should now verify that $u^*$ is in fact an optimal control. Let $J^*(t, x) = J(t, x, u^*)$. Then, it is not hard to show that

$$ J^*(t, x) = \frac{1}{2}\left(\max\{0, |x| - (\tau - t)\}\right)^2 $$

which satisfies the boundary condition $J^*(\tau, x) = \frac{1}{2}x^2$. In addition,

$$ J^*_t(t, x) = (|x| - (\tau - t))^+ \quad \text{and} \quad J^*_x(t, x) = \text{sgn}(x) (|x| - (\tau - t))^+. $$

Therefore, for any $u$ such that $|u| \leq 1$ it follows that

$$ J^*_t(t, x) + J^*_x(t, x) u = (1 + \text{sgn}(x) u) (|x| - (\tau - t))^+ \geq 0 $$

with the equality holding for $u = u^*(t, x)$. Thus, $J^*(t, x)$ is the value function and $u^*$ is optimal. □

**1.4.3 Feedback Control**

In the previous example, the notion of a feedback control policy was introduced. Specifically, a feedback control $u$ is a mapping from $\mathbb{R}^{n+1}$ to $U$ such that $u = u(t, x)$ and the system dynamics
\[
\dot{x} = f(t, x, u(t, x)) \text{ has a unique solution for each initial condition } (s, y) \in Q_0. \text{ Given a feedback control } u \text{ and an initial condition } (s, y), \text{ we can define the trajectory } x(t; s, y) \text{ as the solution to }
\]
\[
\dot{x} = f(t, x, u(t, x)) \quad x(s) = y.
\]
The corresponding control policy is \( u(t) = u(t, x(t; s, y)) \).

A feedback control \( u^* \) is an \textit{optimal feedback control} if for any \((s, y) \in Q_0\) the control \( u(t) = u^*(t, x(t; s, y)) \) solve the optimization problem (1.4.1)-(1.4.3) with initial condition \((s, y)\).

**Theorem 1.4.5** If there is an optimal feedback control \( u^* \) and \( t_1(s, y) \) and \( x(t_1; s, y) \) are the terminal time and terminal state for the trajectory
\[
\dot{x} = f(t, x, u(t, x)) \quad x(s) = y
\]
then the value function \( V(s, y) \) is differentiable at each point at which \( t_1(s, y) \) and \( x(t_1; s, y) \) are differentiable with respect to \((s, y)\).

**Proof:** From the optimality of \( u^* \) we have that
\[
V(s, y) = J(t_1(s, y), x(t_1(s, y); s, y)).
\]
The result follows from this identity and the fact that \( J \) is \( C^1 \).

1.4.4 \textbf{The Linear-Quadratic Problem}

Consider the following optimal control problem.
\[
\begin{align*}
\min_{x(T)} & \quad x(T)'Q_T x(T) + \int_0^T [x(t)' Q x(t) + u(t)' R u(t)] \, dt \\
\text{subject to} & \quad \dot{x}(t) = A x(t) + B u(t)
\end{align*}
\]
(1.4.7)
where the \( n \times n \) matrices \( Q_T \) and \( Q \) are symmetric positive semidefinite and the \( m \times m \) matrix \( R \) is symmetric positive definite. The HJB equation for this problem is given by

\[
\min_{u \in \mathbb{R}^m} \left\{ V_i(t, x) + V_z(t, x)' (A x + B u) + x^t Q x + u^t R u \right\} = 0
\]
with boundary condition \( V(T, x) = x^t Q_T x \).

We guess a quadratic solution for the HJB equation. That is, we suppose that \( V(t, x) = x^t K(t) x \) for a \( n \times n \) symmetric matrix \( K(t) \). If this is the case then
\[
V_i(t, x) = 2K(t) x \quad \text{and} \quad V_z(t, x) = x^t \dot{K}(t) x.
\]

Plugging back these derivatives on the HJB equation we get
\[
\min_{u \in \mathbb{R}^m} \left\{ x^t \dot{K}(t) x + 2x^t K(t) A x + 2x^t K(t) B u + x^t Q x + u^t R u \right\} = 0. \tag{1.4.9}
\]
Thus, the optimal control satisfies
\[
2B'K(t) x + 2Ru = 0 \quad \implies \quad u^* = -R^{-1}B'K(t) x.
\]
Substituting the value of $u^*$ in equation (1.4.9) we obtain the condition
\[ x' \left( \dot{K}(t) + K(t)A + A'K(t) - K(t)BR^{-1}B'K(t) + Q \right) x = 0 \quad \text{for all} \ (t, x). \]

Therefore, for this to hold matrix $K(t)$ must satisfy the continuous-time Ricatti equation in matrix form
\[ \dot{K}(t) = -K(t)A - A'K(t) = K(t)BR^{-1}B'K(t) - Q, \quad \text{with boundary condition} \ K(T) = Q_T. \] (1.4.10)

Reversing the argument it can be shown that if $K(t)$ solves (1.4.10) then $W(t, x) = x'K(t)x$ is a solution of the HJB equation and so at the verification theorem we conclude that it is equal to the value function. In addition, the optimal feedback control is $u^*(t, x) = -R^{-1}B'K(t)x$.

### 1.5 Extensions

#### 1.5.1 The Method of Characteristics for First-Order PDEs

**First-Order Homogeneous Case**

Consider the following first-order homogeneous PDE
\[ u_t(t, x) + a(t, x)u_x(t, x) = 0, \quad x \in \mathbb{R}, t > 0, \]
with boundary conditions $u(x, 0) = \phi(x)$ for all $x \in \mathbb{R}$. We assume that $a$ and $\phi$ are “smooth enough” functions. A PDE problem in this form is referred to as a Cauchy problem.

We will investigate the solution to this problem using the method of characteristics. The characteristics of this PDE are curves in the $x-t$ plane defined by
\[ \dot{x}(t) = a(x(t), t), \quad x(0) = x_0. \] (1.5.1)

Let $\tilde{x} = \tilde{x}(t)$ be a solution with $\tilde{x}(0) = x_0$. Let $u$ be a solution to the PDE, we want to study the evolution of $u$ along $\tilde{x}(t)$.

\[ \dot{u}(t, \tilde{x}(t)) = u_t(t, \tilde{x}(t)) + u_x(t, \tilde{x}(t))\dot{\tilde{x}}(t) = u_t(t, \tilde{x}(t)) + u_x(t, \tilde{x}(t))a(\tilde{x}(t), t) = 0. \]

So, $u(t, x)$ is constant along the characteristic curve $\tilde{x}(t)$, that is,
\[ u(t, \tilde{x}(t)) = u(0, \tilde{x}(0)) = \phi(x_0), \quad \forall t > 0. \] (1.5.2)

Thus, if we are able to solve the ODE (1.5.3) then we would be able to find the solution to the original PDE.

**Example 1.5.1** Consider the Cauchy problem
\[ \begin{align*}
    u_t + x \ u_x &= 0, \quad x \in \mathbb{R}, t > 0 \\
    u(x, 0) &= \phi(x), \quad x \in \mathbb{R}.
\end{align*} \]
The characteristic curves are defined by
\[ \dot{x}(t) = x(t), \quad x(0) = x_0, \]
so \( x(t) = x_0 \exp(t) \). So for a given \((t, x)\) the characteristic passing through this point has initial condition \( x_0 = x \exp(-t) \). Since \( u(t, x(t)) = \phi(x_0) \) we conclude that \( u(t, x) = \phi(x \exp(-t)) \). □

First-Order Non-Homogeneous Case

Consider the following nonhomogeneous problem.

\[
\begin{align*}
  u_t(t, x) + a(t, x) u_x(t, x) &= b(t, x), \quad x \in \mathbb{R}, t > 0 \\
  u(x, 0) &= \phi(x), \quad x \in \mathbb{R}.
\end{align*}
\]

Again, the characteristic curves are given by
\[
\dot{x}(t) = a(x(t), t), \quad x(0) = x_0.
\]

(1.5.3)

Thus, for a solution \( u(t, x) \) of the PDE along a characteristic curve \( \tilde{x}(t) \) we have that
\[
\dot{u}(t, \tilde{x}(t)) = u_t(t, \tilde{x}(t)) + u_x(t, \tilde{x}(t)) \dot{\tilde{x}}(t) = u_t(t, \tilde{x}(t)) + u_x(t, \tilde{x}(t)) a(\tilde{x}(t), t) = b(t, \tilde{x}(t)).
\]

Hence, the solution to the PDE is given by
\[
u(t, \tilde{x}(t)) = \phi(x_0) + \int_0^t b(\tau, \tilde{x}(\tau)) d\tau
\]
along the characteristic \((t, \tilde{x}(t))\).

**Example 1.5.2** Consider the Cauchy problem

\[
\begin{align*}
  u_t + u_x &= x, \quad x \in \mathbb{R}, t > 0 \\
  u(x, 0) &= \phi(x), \quad x \in \mathbb{R}.
\end{align*}
\]

The characteristic curves are defined by
\[
\dot{x}(t) = 1, \quad x(0) = x_0,
\]
so \( x(t) = x_0 + t \). So for a given \((t, x)\) the characteristic passing through this point has initial condition \( x_0 = x - t \). In addition, along a characteristic \( \tilde{x}(t) = x_0 + t \) starting at \( x_0 \), we have
\[
u(t, \tilde{x}(t)) = \phi(x_0) + \int_0^t \tilde{x}(\tau) d\tau = \phi(x_0) + x_0 t + \frac{1}{2} t^2.
\]

Thus, the solution to the PDE is given by
\[
u(t, x) = \phi(x - t) + \left( x - \frac{t^2}{2} \right) t. \quad \Box
\]
Applications to Optimal Control

Given that the partial differential equation of dynamic programming is a first-order PDE, we can try to apply the method of characteristic to find the value function. In general, the HJB is not a standard first-order PDE because of the maximization that takes place. So in general, we cannot just solve a simple first-order PDE to get the value function of dynamic programming. Nevertheless, in some situations it is possible to obtain good results as the following example shows.

Example 1.5.3 (Method of Characteristics) Consider the optimal control problem

\[
\begin{align*}
\min_{\|u\| \leq 1} & \quad J(t_0, x_0, u) = \frac{1}{2}(x(\tau))^2 \\
\text{subject to} & \quad \dot{x}(t) = u(t), \quad x(t_0) = x_0
\end{align*}
\]

where \(\|u\| = \max_{0 \leq t \leq \tau} |u(t)|\).

A candidate for value function \(W(t, x)\) should satisfy the HJB equation

\[
\min_{|u| \leq 1} \{W_t(t, x) + W_x(t, x) u\} = 0,
\]

with boundary condition \(W(\tau, x) = \frac{1}{2}x^2\).

For a given \(u \in U\), let solve the PDE

\[
W_t(t, x; u) + W_x(t, x; u) u = 0, \quad W(\tau, x; u) = \frac{1}{2}x^2. \tag{1.5.4}
\]

A characteristic curve \(\tilde{x}(t)\) is found solving

\[
\dot{x}(t) = u, \quad x(0) = x_0,
\]

so \(\tilde{x}(t) = x_0 + ut\). Since the solution to the PDE is constant along the characteristic curve we have

\[
W(t, \tilde{x}(t); u) = W(\tau, \tilde{x}(\tau); u) = \frac{1}{2}(x(\tau))^2 = \frac{1}{2}(x_0 + u\tau)^2.
\]

The characteristic passing through the point \((t, x)\) has initial condition \(x_0 = x - ut\), so the general solution to the PDE (1.5.4) is

\[
W(t, x; u) = \frac{1}{2}(x + (\tau - t)u)^2.
\]

Since our objective is to minimize the terminal cost, we can identify a policy by minimizing \(W(t, x; u)\) over \(u\) above. It is straightforward to see that the optimal control (in feedback form) satisfies

\[
u^*(x, t) = \begin{cases} 
-1 & \text{if } x > \tau - t \\
\frac{x}{\tau - t} & \text{if } |x| \leq \tau - t \\
1 & \text{if } x < t - \tau.
\end{cases}
\]

The corresponding “candidate” for value function \(W^*(t, x) = W(t, x; u^*(t, x))\) satisfies

\[
W(t, x) = \frac{1}{2} \left( \max\{0; |x| - (\tau - t)\} \right)^2
\]

which we already know satisfies the HJB equation. □
1.5.2 Optimal Control and Myopic Solution

Consider the following deterministic control problem in Bolza form:

\[
\min_{u \in U} J(x(T)) + \int_0^T L(x_t, u_t) \, dt
\]

subject to \( \dot{x}(t) = f(x_t, u_t), \quad x(0) = x_0. \)

The functions \( f, J, \) and \( L \) are assumed to be “sufficiently” smooth.

The solution to this problem can be found solving the associated Hamilton-Jacobi-Bellman equation

\[
V_t(t, x) + \min_{u \in U} \{ f(x, u) V(x, t) + L(x, u) \} = 0
\]

with boundary condition \( V(T, x) = J(x) \). The value function \( V(t, x) \) represents the optimal cost-to-go starting at time \( t \) in state \( x \).

Suppose, we fix the control \( u \in U \) and solve the first-order PDE

\[
W_t(t, x; u) + f(x, u) W_x(t, x; u) + L(x, u) = 0, \quad W(T, x; u) = J(x)
\]  

(1.5.5)

using the methods of characteristics. That is, we solve the characteristic ODE \( \dot{x}(t) = f(x, u) \) and let \( x(t) = H(t; s, y, u) \) the solution passing through the point \((s, y)\), i.e., \( x(s) = H(s; s, y, u) = y \).

By construction, along a characteristic curve \((t, x(t))\) the function \( W(t, x(t); u) \) satisfies \( \dot{W}(t, x(t); u) + L(x(t), u) = 0 \). Therefore, after integration we have that

\[
W(s, x(s); u) = W(T, x(T); u) + \int_s^T L(x(t), u) \, dt = J(x(T)) + \int_s^T L(x(t), u),
\]

where the second equality follows from the boundary condition for \( W \). We can rewrite this last identity for the particular characteristic curve passing through \((t, x)\) as follows

\[
W(t, x; u) = J(H(T; t, x, u)) + \int_t^T L(H(s; t, x, u), u) \, ds.
\]

Since the control \( u \) has been fixed so far, we call \( W(t, x; u) \) the static value function associated to control \( u \). Now, if we view \( W(t, x; u) \) as a function of \( u \), we can minimize this static value function. We define

\[
u^*(t, x) = \arg\min_{u \in U} W(t, x; u) \quad \text{and} \quad \mathcal{V}(t, x) = W(t, x; u^*(t, x)).\]

Proposition 1.5.1 Suppose that \( u^*(t, x) \) is an interior solution and that \( W(t, x; u) \) is sufficiently smooth so that \( u^*(t, x) \) satisfies

\[
\left. \frac{dW(t, x; u)}{du} \right|_{u=u^*(t,x)} = 0. \quad (1.5.6)
\]

Then the function \( \mathcal{V}(t, x) \) satisfies the PDE

\[
\mathcal{V}_t(t, x) + f(x, u^*(t, x)) \mathcal{V}_x(t, x) + L(x, u^*(t, x)) = 0
\]  

(1.5.7)

with boundary condition \( \mathcal{V}(T, x) = J(x) \).
Proof: Let us rewrite the PDE in terms of \( W(t, x, u^*) \) to get
\[
\frac{\partial W(t, x, u^*)}{\partial t} + f(x, u^*) \frac{\partial W(t, x, u^*)}{\partial x} + L(x, u^*) + \left[ \frac{\partial u^*}{\partial t} + f(x, u^*) \frac{\partial u^*}{\partial x} \right] \frac{\partial W(t, x, u^*)}{\partial u^*}.
\]

We note that by construction of the function \( W \) on equation (1.5.5) the expression denoted by (a) is equal to zero. In addition, the optimality condition (1.5.6) implies that (b) is also equal to zero. Therefore, \( V(t, x) \) satisfies the PDE (1.5.7). The border condition follows again from the definition of the value function \( W \).

Given this result, the question that naturally arises is whether \( V(t, x) \) is in fact the value function (that is \( V(t, x) = V(t, x) \)) and \( u^*(t, x) \) is the corresponding optimal feedback control.

Unfortunately, this is not generally true. In fact, to prove that \( V(t, x) = V(t, x) \) we would need to show that
\[
u^*(t, x) = \arg \min_{u \in U} \{ f(x, u) V_x(t, x) + L(x, u) \}.
\]

Since we have assumed that \( u^*(t, x) \) is an interior solution then the first order optimality condition for the minimization problem above is given by
\[
f_u(x, u^*(t, x)) V_x(t, x) + L_u(x, u^*(t, x)) = 0.
\]

Using the optimality condition (1.5.6) we have that
\[
V_x(t, x) = W_x(t, x; u^*) = J'(H) H_x + \int_t^T L_x(H, u^*) H_x \, dt,
\]
where \( H = H(T; t; x, u^*) \) and \( H_x = H_x(T; t; x, u^*) \) the partial derivative of \( H(T; t; x, u^*) \) with respect to \( x \) keeping \( u^* = u^*(t, x) \) fixed. Thus, the first order optimality condition that needs to be verified is
\[
f_u(x, u^*(t, x)) \left( J'(H) H_x + \int_t^T L_x(H, u^*) H_x \, dt \right) + L_u(x, u^*(t, x)) = 0. \tag{1.5.8}
\]

On the other hand, the optimality condition (1.5.6) that \( u^*(t, x) \) satisfies is
\[
J'(H) H_u + \int_t^T \left[ L_x(H, u^*) H_u + L_u(H, u^*) \right] \, dt = 0. \tag{1.5.9}
\]

It should be clear that condition (1.5.9) does not necessarily imply condition (1.5.8) and so \( V(t, x) \) and \( u^*(t, x) \) are not guaranteed to be the value function and the optimal feedback control, respectively. The following example shows the suboptimality of \( u^*(t, x) \).

Example 1.5.4 Consider the traditional linear-quadratic control problem
\[
\min_u \left\{ x^2(T) + \int_0^T (x^2(t) + u^2(t)) \, dt \right\}
\]
subject to \( \dot{x}(t) = \alpha x(t) + \beta u(t), \quad x(0) = x_0. \)

- **Exact solution to the HJB equation:** This problem is traditionally tackled solving an associated Riccati differential equation. We suppose that the optimal control satisfies
\[
u(t, x) = -\beta k(t) x,
\]

---

CHAPTER 1. DETERMINISTIC OPTIMAL CONTROL
where the function \( k(t) \) satisfies the Riccati ODE
\[
\dot{k}(t) + 2\alpha k(t) = \beta^2 k^2(t) - 1, \quad k(T) = 1.
\]

We can get a particular solution assuming \( k(t) = \bar{k} = \text{constant} \). In this case,
\[
\beta^2 \bar{k}^2 - 2\alpha \bar{k} - 1 = 0 \quad \Rightarrow \quad \bar{k} = \frac{\alpha \pm \sqrt{\alpha^2 + \beta^2}}{\beta^2}.
\]

Now, let us define \( k(t) = z(t) + \bar{k}^+ \) then the Riccati becomes
\[
\dot{z}(t) + 2(\alpha - \beta^2 \bar{k}^+) z(t) = \beta^2 z^2(t) \quad \Rightarrow \quad \frac{\dot{z}(t)}{z^2(t)} + \frac{2(\alpha - \beta^2 \bar{k}^+)}{z(t)} = \beta^2.
\]

If we set \( w(t) = z^{-1}(t) \) then the last ODE is equivalent to
\[
\dot{w}(t) + 2(\alpha - \beta^2 \bar{k}^+) w(t) = \beta^2.
\]

This a simple linear differential equation that can be solved using the integrating factor \( \exp(2(\alpha - \beta^2 \bar{k}^+) t) \), that is,
\[
\frac{d}{dt} \left( \exp \left( 2(\alpha - \beta^2 \bar{k}^+) t \right) w(t) \right) = \exp \left( 2(\alpha - \beta^2 \bar{k}^+) t \right) \beta^2.
\]

The solution to this ODE is
\[
w(t) = \bar{k} \exp \left( -2(\alpha - \beta^2 \bar{k}^+) t \right) + \frac{\beta^2}{2(\alpha - \beta^2 \bar{k}^+)}
\]

where \( \bar{k} \) is a constant of integration. Using the fact that \( \alpha - \beta^2 \bar{k}^+ = -\sqrt{\alpha^2 + \beta^2} \) and \( k(t) = \bar{k}^+ + 1/w(t) \) we get
\[
k(t) = \frac{\alpha + \sqrt{\alpha^2 + \beta^2}}{\beta^2} + \frac{2\sqrt{\alpha^2 + \beta^2}}{2\bar{k} \sqrt{\alpha^2 + \beta^2} \exp \left( -2(\alpha - \beta^2 \bar{k}^+) t \right) - \beta^2}.
\]

The value of \( \bar{k} \) is obtained from the border condition \( k(T) = 1 \).

\textbf{Myopic Solution:} If we solve the problem using the myopic approach described at the beginning of this notes, we get that the characteristic curve is given by
\[
\dot{x}(t) = \alpha x(t) + \beta u \quad \Rightarrow \quad \ln(\alpha x(t) + \beta u) = \alpha(t + A),
\]

with \( A \) a constant of integration. The characteristic passing through the point \((t, x)\) satisfies \( A = \ln(\alpha x + \beta u) / \alpha - t \) and is given by
\[
x(\tau) = \frac{(\alpha x + \beta u) \exp(\alpha(\tau - t)) - \beta u}{\alpha}.
\]

The value of the static value function \( W(t, x; u) \) is given by
\[
W(t, x; u) = \left( \frac{(\alpha x + \beta u) \exp(\alpha(T - t)) - \beta u}{\alpha} \right)^2 + \int_t^T \left( \frac{(\alpha x + \beta u) \exp(\alpha(\tau - t)) - \beta u}{\alpha} \right)^2 + u^2 \, d\tau.
\]

If we compute the derivative of \( W(t, x; u) \) with respect to \( u \) and make it equal to zero we get, after some manipulations, that the optimal myopic solution is
\[
u^*(t, x) = \left( \frac{\alpha \beta (3 \exp(2(T - t)) - 4 \exp(T - t) + 1)}{\beta^2 (3 \exp(2(T - t)) - 8 \exp(T - t) + 5) + 2(T - t)(\alpha^2 + \beta^2)} \right) x.
\]

Interestingly, this myopic feedback control is also linear on \( x \) as in the optimal solution, however, the solution is clearly different and suboptimal. \( \square \)
The previous example shows that in general the use of a myopic policy produces suboptimal solutions. However, a question remains still open which is under what conditions is the myopic solution optimal? A general solution to this problem can be obtained by looking under what restrictions on the problem’s data the optimality condition (1.5.8) is implied by condition (1.5.9).

In what follows we present one specific case for which the optimal solution is given by the myopic solution. Consider the control problem

\[
\min_{u \in U} J(x(T)) + \int_0^T L(u(t)) \, dt
\]

subject to \( \dot{x}(t) = f(x(t), u(t)) := g(x(t)) h(u(t)), \quad x(0) = x_0. \)

In this case, it can be shown that the characteristic equation passing through the point \((t, x)\) is given by

\[
x(\tau) = G^{-1}(h(u)(\tau - t) + G(x)), \quad \text{where } G(x) := \int \frac{dx}{g(x)}
\]

In this case, the static value function is

\[
W(t, x; u) = J(G^{-1}(h(u)(T - t) + G(x))) + L(u)(T - t)
\]

and the myopic solution satisfies \( \frac{d}{du} W(t, x; u) = 0 \) or equivalently

\[
0 = J'(G^{-1}(h(u)(T - t) + G(x))) h'(u)(T - t) G^{-1}(h(u)(T - t) + G(x)) + (T - t) L'(u) \iff
0 = J'(G^{-1}(h(u)(T - t) + G(x))) f_u(x, u) G'(x) G^{-1}(h(u)(T - t) + G(x)) + L'(u) \iff
0 = f_u(x, u) W_x(t, x; u) + L'(u).
\]

The second equality uses the identities \( G'(x) = 1/f(x) \) and \( f_u(x, u) = f(x) h'(u) \). Therefore, the optimal myopic policy \( u^*(t, x) \) satisfies

\[
0 = f_u(x, u^*) V_x(t, x) + L'(u^*)
\]

i.e., the first order optimality condition (1.5.8).

**Example 1.5.5** Consider control problem

\[
\min_{u} \left\{ x^2(T) + \int_0^T u^2(t) \, dt \right\}
\]

subject to \( \dot{x}(t) = x(t) u(t), \quad x(0) = x_0. \)

In this case, the characteristic passing through \((t, x)\) is given by

\[
x(\tau) = x \exp(u(\tau - t)).
\]

The static value function is

\[
W(t, x; u) = x^2 \exp(2u(T - t)) + u^2 (T - t).
\]

Minimizing \( W \) over \( u \) implies

\[
x^2 \exp(2u^*(T - t)) + u^* = 0
\]

and the corresponding value function

\[
V(t, x) = V(t, x) = u^*(t, x) \left( u^*(t, x) (T - t) - 1 \right). \quad \Box
\]
Connecting the HJB Equation with Pontryagin Principle

We consider the optimal control problem in Lagrange form. In this case, the HJB equation is given by

\[
\min_{u \in U} \{ V_t(t, x) + V_x(t, x) f(t, x, u) + L(t, x, u) \} = 0,
\]

with boundary condition \( V(t_1, x(t_1)) = 0 \).

Let us define the so-called Hamiltonian

\[
H(t, x, u, \lambda) := \lambda f(x, t, u) - L(t, x, u).
\]

Thus, the HJB equation implies that the value function satisfies

\[
\max_{u \in U} H(t, x, u, -V_x) = 0,
\]

and so the optimal control can be found maximizing the Hamiltonian. Specifically, let \( x^*(t) \) be the optimal trajectory and let \( P(t) = -V_x(t, x^*(t)) \), then the optimal control satisfies the so-called Maximum Principle

\[
H(t, x^*(t), u^*(t), P(t)) \leq H(t, x^*(t), u, P(t)), \quad \text{for all } u \in U.
\]

In order to complete the connection with Pontryagin principle we need to derive the adjoint equations. Let \( x^*(t) \) be the optimal trajectory and consider a small perturbation \( x(t) \) such that

\[
x(t) = x^*(t) + \delta(t), \quad \text{where } |\delta(t)| < \epsilon.
\]

First, we note that the HJB equation together with the optimality of \( x^* \) and its corresponding control \( u^* \) implies that

\[
H(t, x^*(t), u^*(t), -V_x(t, x^*(t))) - V_t(t, x^*(t)) \geq H(t, x^*(t), u^*(t), -V_x(t, x(t))) - V_t(t, x(t)).
\]

Therefore, the derivative of \( H(t, x(t), u^*(t), -V_x(t, x(t))) + V_t(t, x(t)) \) with respect to \( x \) so be equal to zero at \( x^*(t) \). Using the definition of \( H \) this condition implies that

\[
-V_{xx}(t, x^*(t)) f(t, x^*(t), u^*(t)) - V_x(t, x^*(t)) f_x(t, x^*(t), u^*(t)) - L_x(t, x^*(t), u^*(t)) - V_{xt}(t, x^*(t)) = 0.
\]

In addition, using the dynamics of the system we get that

\[
\dot{V}_x(t, x^*(t)) = V_{tx}(t, x^*(t)) + V_{xx}(t, x^*(t)) f(t, x^*(t), u^*(t)),
\]

therefore

\[
\dot{V}_x(t, x^*(t)) = V_x(t, x^*(t)) f(t, x^*(t), u^*(t)) + L(t, x^*(t), u^*(t)).
\]

Finally, using the definition of \( P(t) \) and \( H \) we conclude that \( P(t) \) satisfies the adjoint condition

\[
\dot{P}(t) = \frac{\partial}{\partial x} H(t, x^*(t), u^*(t), P(t)).
\]

The boundary condition for \( P(t) \) are obtained from the boundary conditions of the HJB, that is,

\[
P(t_1) = -V_x(t_1, x(t_1)) = 0. \quad \text{(transversality condition)}
\]
Economic Interpretation of the Maximum Principle

Let us again consider the control problem in Lagrange form. In this case the performance measure is

\[ V(t, x) = \min \int_0^T L(t, x(t), u(t)) \, dt. \]

The function \( L \) corresponds to the instantaneous “cost” rate. According to our definition of \( P(t) = -V_x(t, x(t)) \), we can interpret this quantity as the marginal profit associated to a small change on the state variable \( x \). The economic interpretation of the Hamiltonian is as follows:

\[
H \, dt = P(t) f(t, x, u) \, dt - L(t, x, u) \, dt = P(t) \dot{x}(t) \, dt - L(t, x, u) \, dt.
\]

The term \( -L(t, x, u) \, dt \) corresponds to the instantaneous profit made at time \( t \) at state \( x \) if control \( u \) is selected. We can look at this profit as a direct contribution. The second term \( P(t) \, dx(t) \) represents the instantaneous profit that it is generated by changing the state from \( x(t) \) to \( x(t) + dx(t) \). We can look at this profit as an indirect contribution. Therefore \( H \, dt \) can be interpreted as the total contribution made from time \( t \) to \( t + dt \) given the state \( x(t) \) and the control \( u \).

With this interpretation, the Maximum Principle simply state that an optimal control should try to maximize the total contribution for every time \( t \). In other words, the Maximum Principle decouples the dynamic optimization problem into a series of static optimization problem, one for every time \( t \).

Note also that if we integrate the adjoint equation we get

\[ P(t) = \int_t^{t_1} H_x \, dt. \]

So \( P(t) \) is the cumulative gain obtained over \([t, t_1]\) by marginal change of the state space. In this respect, the adjoint variables behave in much the same way as dual variables in LP.

1.6 Exercises

Exercise 1.6.1 In class, we solved the following deterministic optimal control problem

\[
\min_{\|u\| \leq 1} J(t_0, x_0, u) = \frac{1}{2}(x(\tau))^2
\]

subject to \( \dot{x}(t) = u(t), \quad x(t_0) = x_0 \)

where \( \|u\| = \max_{0 \leq t \leq \tau} \{|u(t)|\} \) using the method of characteristics. In particular, we solved the open-loop HJB PDE equation

\[
W_t(t, x; u) + W_x(t, x; u) u = 0, \quad W(\tau, x; u) = \frac{1}{2} x^2.
\]

for a fixed \( u \) and then find the optimal close-loop control solving

\[
u^*(t, x) = \arg \min_{\|u\| \leq 1} W(t, x; u)
\]
and computing the value function as \( V(t, x) = W(t, x; u^*(t, x)). \)

(a) Explain why this methodology does not work in general. Provide a counter example.

(b) What specific control problems can be solve using this open-loop approach.

(c) Propose an algorithm that uses the open-loop solution to approximately solve a general deterministic optimal control problem.

**Exercise 1.6.2 (Dynamic Pricing in Discrete Time)**

Assume that we have \( x_0 \) items of a certain type that we want to sell over a period of \( N \) days. At each day, we may sell at most one item. At the \( k^{th} \) day, knowing the current number \( x_k \) of remaining unsold items, we can set the selling price \( u_k \) of a unit item to a nonnegative number of our choice; then, the probability \( q_k(u_k) \) of selling an item on the \( k^{th} \) day depends on \( u_k \) as follows:

\[
q_k(u_k) = \alpha \exp(-u_k)
\]

where \( 0 < \alpha < 1 \) is a given scalar. The objective is to find the optimal price setting policy so as to maximize the total expected revenue over \( N \) days. Let \( V_k(x_k) \) be the optimal expected cost from day \( k \) to the end if we have \( x_k \) unsold units.

(a) Assuming that for all \( k \), the value function \( V_k(x_k) \) is monotonically nondecreasing as a function of \( x_k \), prove that for \( x_k > 0 \), the optimal prices have the form

\[
\mu_k^*(x_k) = 1 + J_{k+1}(x_k) - V_{k+1}(x_k - 1)
\]

and that

\[
V_k(x_k) = \alpha \exp(-\mu_k^*(x_k)) + V_{k+1}(x_k).
\]

(b) Prove simultaneously by induction that, for all \( k \), the value function \( V_k(x_k) \) is indeed monotonically nondecreasing as a function of \( x_k \), that the optimal price \( \mu_k^*(x_k) \) is monotonically nonincreasing as a function of \( x_k \), and that \( V_k(x_k) \) is given in closed form by

\[
V_k(x_k) = \begin{cases} 
(N - k) \alpha \exp(-1) & \text{if } x_k \geq N - k, \\
\sum_{i=k}^{N-x_k} \alpha \exp(-\mu_i^*(x_k)) + x_k \alpha \exp(-1) & \text{if } 0 < x_k < N - k, \\
0 & \text{if } x_k = 0.
\end{cases}
\]

**Exercise 1.6.3** Consider a deterministic optimal control problem in which \( u \) is a scalar control and \( x \) is also scalar. The dynamics are given by

\[
f(t, x, u) = a(x) + b(x)u
\]

where \( a(x) \) and \( b(x) \) are \( C^2 \) vector functions. If

\[
P(t) b(x(t)) = 0 \text{ on a time interval } \alpha \leq t \leq \beta,
\]

the Hamiltonian does not depend on \( u \) and the problem is singular. Show that under these conditions

\[
P(t) q(x) = 0, \quad \alpha \leq t \leq \beta,
\]
where \( q(x) = b_x(x)a(x) - a_x(x)b(x) \). Show further that if

\[
P(t)[q_x(x(t)) b(x(t)) - b_x(x(t)) q(x(t))] \neq 0
\]

then

\[
u(t) = -\frac{P(t)[q_x(x(t)) a(x(t)) - a_x(x(t)) q(x(t))]}{P(t)[q_x(x(t)) b(x(t)) - b_x(x(t)) q(x(t))]}.
\]

**Exercise 1.6.4** The objective of this note is to characterize a particular family of Learning Function. These learning functions are useful modelling devices for situations where there is an agent that tries to increase his or her level of “knowledge” about a certain phenomenon (such as customers’ preferences or product quality) by applying a certain control or “effort”. To fix ideas, in what follows knowledge will be represented by the variable \( x \) while effort will be represented by the variable \( e \). For simplicity we will assume that knowledge takes values in the \([0, 1]\) interval while effort is a nonnegative real variable. The family of learning function that we are interested in this note are those than can be derived from a specific subfamily that we called Additive Learning Functions. The formal definition of an Additive Learning Function\(^1\) is as follows.

**Definition 1.6.1** Consider a function \( L : \mathbb{R}_+ \times [0, 1] \to [0, 1] \). The function \( L \) would be called Additive Learning Function if it satisfies the following properties:

- **Additivity**: \( L(e_2 + e_1, x) = L(e_2, L(e_1, x)) \) for all \( e_1, e_2 \in \mathbb{R}_+ \) and \( x \in [0, 1] \).

- **Boundary Condition**: \( L(0, x) = x \) for all \( x \in [0, 1] \).

- **Monotonicity**: \( L_e(t, x) = \frac{\partial L}{\partial e} L(e, x) > 0 \) for all \((e, x) \in \mathbb{R} \times [0, 1] \).

- **Satiation**: \( \lim_{e \to \infty} L(e, x) = 1 \) for all \( x \in [0, 1] \).

a) Prove the following. Suppose that \( L(e, x) \) is a \( C^1 \) additive learning function. Then \( L(e, x) \) satisfies

\[
L_e(e, x) - L_e(0, x) L_x(e, x) = 0
\]

where \( L_e \) and \( L_x \) are the partial derivatives of \( L(e, x) \) with respect to \( e \) and \( x \) respectively.

b) Using the method of characteristics solve the PDE of part a) as a function of

\[
H(x) = \int -\frac{1}{L_e(0, x)} \, dx
\]

and prove that the solution is of the form

\[
L(e, x) := H^{-1}(H(x) - e).
\]

Consider the following optimal control problem.

\[
V(0, x) = \max_{p_t} \int_0^T \left[ p_t \lambda(p_t) x_t \right] \, dt \quad \text{subject to} \quad \dot{x}_t = L_e(0, x_t) \lambda(p_t) \quad x_0 = x \in [0, 1] \text{ given.}
\]

\[^1\text{This name is probably not standard since I do not know the relevant literature well enough.}\]
Where \( L_e(0, x) \) is the partial derivative of the learning function \( L(e, x) \) with respect to \( e \) evaluated at \((0, x)\). This problem corresponds to the case of a seller that tries to maximize cumulative revenue during the period \([0, T]\). Potential demand rate at time \( t \) is given by \( \lambda(p_t) \) where \( p_t \) is the price set by the seller at time \( t \). However, only a fraction \( x_t \in [0, 1] \) of the potential customers buy the product at time \( t \). The dynamics of \( x_t \) are given by (1.6.2).

c) Show that equation (1.6.2) can be rewritten as

\[
x_t = L(y_t, x) \quad \text{where} \quad y_t := \int_0^t \lambda_s \, ds.
\]

and use this fact to reformulate your control problem as follows

\[
\max_{y_t} \int_0^T p(\dot{y}_t) \dot{y}_t L(y_t, x) \, dt \quad \text{subject to} \quad y_0 = 0.
\]  

(1.6.3)

d) Deduce that the optimality conditions in this case are given by

\[
\dot{y}_t^2 p'(\dot{y}_t) L(y_t, x) = \text{constant}.
\]  

(1.6.4)

e) Solve the optimality condition for the case

\[
\lambda(p) = \lambda_0 \exp(-\alpha p) \quad \text{and} \quad L(e, x) = 1 + (x - 1) \exp(-\beta e), \quad \alpha, \beta > 0.
\]

### 1.7 Exercises

**Exercise 1.7.1** Solve the problem:

\[
\min (x(T))^2 + \int_0^T (u(t))^2 \, dt
\]

subject to \( \dot{x}(t) = u(t), \quad |u(t)| \leq 1, \quad \forall t \in [0, T] \)

Calculate the cost-to-go function \( J^*(t, x) \) and verify that it satisfies the HJB equation.

A young investor has earned in the stock market a large amount of money \( S \) and plans to spend it so as to maximize his enjoyment through the rest of his life without working. He estimates that he will live exactly \( T \) more years and that his capital \( x(t) \) should be reduced to zero at time \( T \), i.e. \( x(T) = 0 \). Also, he models the evolution of his capital by the differential equation

\[
\frac{dx(t)}{dt} = \alpha x(t) - u(t),
\]

where \( x(0) = S \) is his initial capital, \( \alpha > 0 \) is a given interest rate, and \( u(t) \geq 0 \) is his rate of expenditure. The total enjoyment he will obtain is given by

\[
\int_0^T e^{-\beta t} \sqrt{u(t)} \, dt
\]

Here \( \beta \) is some positive scalar, which serves to discount future enjoyment. Find the optimal \( \{u(t)|t \in [0, T]\} \).
Exercise 1.7.2 Analyze the problem of finding a curve \( \{ x(t) | t \in [0, T] \} \) that maximizes the area under \( x \),

\[
\int_0^T x(t) \, dt,
\]

subject to the constraints

\[
x(0) = a, \quad x(T) = b, \quad \int_0^T \sqrt{1 + (\dot{x}(t))^2} \, dt = L,
\]

where \( a, b \) and \( L \) are given positive scalars. The last constraint is known as the “isoperimetric constraint”: it requires that the length of the curve be \( L \).

Hint: Introduce the system equations \( \dot{x}_1 = u, \dot{x}_2 = \sqrt{1 + u^2} \), and view the problem as a fixed terminal state problem. Show that the optimal curve \( x(t) \) satisfies the condition \( \sin \phi(t) = (C_1 - t)/C_2 \) for given constants \( C_1, C_2 \). Under some assumptions on \( a, b, \) and \( L \), the optimal curve is a circular arc.

Exercise 1.7.3 Let \( a, b \) and \( T \) be positive scalars, and let \( A = (0, a) \) and \( B = (T, b) \) be two points in a medium within which the velocity of propagation of light is proportional to the vertical coordinate. Thus the time it takes for light to propagate from \( A \) to \( B \) along curve \( \{ x(t) | t \in [0, T] \} \) is

\[
\int_0^T \frac{\sqrt{1 + (\dot{x}(t))^2}}{Cx(t)} \, dt,
\]

where \( C \) is a given positive constant. Find the curve of minimum travel time of light from \( A \) to \( B \), and show that it is an arc of a circle of the form

\[
(x(t))^2 + (t - d)^2 = D,
\]

where \( d \) and \( D \) are some constants.

Hint: Introduce the system equation \( \dot{x} = u \), and consider a fixed initial/terminal state problem \( x(0) = a \) and \( x(T) = b \).

Exercise 1.7.4 Use the discrete time Minimum Principle to solve the following problem:

A farmer annually producing \( x_k \) units of a certain crop stores \( (1 - u_k)x_k \) units of his production, where \( 0 \leq u_k \leq 1 \), and invests the remaining \( u_kx_k \) units, thus increasing the next year’s production to a level \( x_{k+1} \) given by

\[
x_{k+1} = x_k + \bar{w}u_kx_k, \quad k = 0, 1, \ldots, N - 1
\]

The scalar \( \bar{w} \) is fixed at a known deterministic value. The problem is to find the optimal investment policy that maximizes the total expected product stored over \( N \) years,

\[
x_N + \sum_{k=0}^{N-1} (1 - u_k)x_k
\]

Show the optimality of the following policy that consists of constant functions:

1. If \( \bar{w} > 1 \), \( \mu_0^*(x_0) = \cdots = \mu_{N-1}^*(x_{N-1}) = 1 \).
2. If $0 < \bar{w} < 1/N$, \[ \mu_0^*(x_0) = \cdots = \mu_{N-1}^*(x_{N-1}) = 0. \]

3. If $1/N \leq \bar{w} \leq 1$,
\[
\begin{align*}
\mu_0^*(x_0) &= \cdots = \mu_{N-\bar{k}-1}^*(x_{N-\bar{k}-1}) = 1, \\
\mu_{N-\bar{k}}^*(x_{N-\bar{k}}) &= \cdots = \mu_{N-1}^*(x_{N-1}) = 0,
\end{align*}
\]

where $\bar{k}$ is such that \[ \frac{1}{\bar{k}+1} < \bar{w} \leq \frac{1}{\bar{k}}. \]
Chapter 2

Discrete Dynamic Programming

Dynamic programming (DP) is a technique pioneered by Richard Bellman\(^1\) in the 1950’s to model and solve problems where decisions are made in stages\(^2\) in order to optimize a particular functional (e.g., minimize a certain cost) that depends (possibly) on the entire evolution (trajectory) of the system over time as well as on the decisions that were made along the way. The distinctive feature of DP (and one that is useful to keep in mind) with respect to the method of Calculus of Variations discussed in the previous chapter is that instead of thinking of an optimal trajectory as a point in an appropriate space, DP constructs this optimal trajectory sequentially over time, in essence DP is an algorithm.

A fundamental idea that emerges from DP is that in general decisions cannot be made myopically (that is, optimizing current performance) since a low cost now might mean a high cost in the future.

2.1 Discrete-Time Formulation

Let us introduce the basic DP model using one of the most classical examples in Operations Management, namely, the Inventory Control problem.

Example 2.1.1 (Inventory control) Consider the problem faced by a firm that must replenish periodically (i.e., every month) the level of inventory of a certain good. The inventory of this good is used to satisfy a (possibly stochastic) demand. The dynamics of this inventory system are depicted in Figure 3.1.1. There are two costs incurred per period: a per-unit purchasing cost \(c\), and an inventory cost incurred at the end of a period that accounts for either holding (even there is a positive amount of inventory that is carried over to the next period) or backlog costs (associated to unsatisfied demand that must be met in the future) given by a function \(r(\cdot)\). The manager of this firm must decide at the beginning of every period \(k\) the amount of inventory to order \((u_k)\) based on the initial level of inventory in period \(k\) \((x_k)\) and the available forecast of future demands, \((w_k, w_{k+1}, \ldots, w_N)\), this forecast is captured by the underlying joint probabilities distribution of these future demands.

\(^1\)For a brief historical account of the early developments of DP, including the origin of its name, see S. Dreyfus (2002). “Richard Bellman on the Birth of Dynamic Programming”, Operations Research vol. 50, No. 1, JanFeb, 4851.

\(^2\)For the most part we will consider applications in which these different stages correspond to different moments in time.
Assumptions:

1. Leadtime = 0 (i.e., instantaneous replenishment)
2. Independent demands \( w_0, w_1, \ldots, w_{N-1} \)
3. Fully backlogged demand
4. Zero terminal cost (i.e., free disposal \( g_N(x_N) = 0 \))

The objective is to minimize the total cost over \( N \) periods, i.e.

\[
\min_{u_0, \ldots, u_{N-1} \geq 0} \mathbb{E}_{w_0, w_1, \ldots, w_{N-1}} \left[ \sum_{k=0}^{N-1} (cu_k + r(x_k + u_k - w_k)) \right]
\]

We will prove that for convex cost functions \( r(\cdot) \), the optimal policy is of the "order up to" form. □

The inventory problem highlights the following main features of our Basic Model:

1. An underlying discrete-time dynamic system
2. A finite horizon
3. A cost function that is additive over time

System dynamics are described by a sequence of states driven by a system equation

\[
x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, \ldots, N - 1,
\]

where:

- \( k \) is a discrete time index
- \( f_k \) is the state transition function
- \( x_k \) is the current state of the system. It could summarize past information relevant for future optimization when the system is not Markovian.
• $u_k$ is the control; decision variable to be selected at time $k$

• $w_k$ is a random parameter ("disturbance" or "noise") described by a probability distribution $P_k(\cdot | x_k, u_k)$

• $N$ is the length of the horizon; number of periods when control is applied

The per-period cost function is given by $g_k(x_k, u_k, w_k)$. The total cost function is additive, with a total expected cost given by

$$E_{w_0, w_1, \ldots, w_{N-1}} \left[ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right],$$

where the expectation is taken over the joint distribution of the random variables $w_0, w_1, \ldots, w_{N-1}$ involved.

The sequence of events in a period $k$ is the following:

1. The system manager observes the current state $x_k$
2. Decision $u_k$ is made
3. Random noise $w_k$ is realized. It could potentially depend on $x_k$ and $u_k$ (for example, think of a case where $u_k$ is price and $w_k$ is demand).
4. Cost $g_k(x_k, u_k, w_k)$ is incurred
5. Transition $x_{k+1} = f_k(x_k, u_k, w_k)$ occurs

If we think about tackling a possible solution to a discrete DP such as the Inventory example 2.1.1, two somehow extreme strategies can be considered:

1. **Open Loop**: Select all orders $u_0, u_1, \ldots, u_{N-1}$ at time $k = 0$.
2. **Closed Loop**: Sequential decision making, place an order $u_k$ at time $k$. Here, we gain information about the realization of demand on the fly.

Intuitively, in a deterministic DP settings in which the values of $(w_0, w_1, \ldots, w_{N-1})$ are known at time 0, open and closed loop strategies are equivalent because no uncertainty is revealed over time and hence there is no gain from waiting. However, in a stochastic environment postponing decision can have a significant impact on the overall performance of a particular strategy. So closed-loop optimization are generally needed to solve a stochastic DP problem to optimality. In closed-loop optimization, we want to find an optimal rule (i.e., a policy) for selecting action $u_k$ in period $k$, as a function of the state $x_k$. So, we want to find a sequence of functions $\mu_k(x_k) = u_k, k = 0, 1, \ldots, N-1$. The sequence $\pi = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$ is a policy or control law. For each policy $\pi$, we can associate a trajectory $x^\pi = (x^\pi_0, x^\pi_1, \ldots, x^\pi_N)$ that describes the evolution of the state of the system (e.g., units in inventory at the beginning of every period in Example 2.1.1) over time when the policy $\pi$ has been chosen. Note that in genera $x^\pi$ is a stochastic process. The corresponding performance of policy $\pi = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$ is given by

$$J_\pi = E_{w_0, w_1, \ldots, w_{N-1}} \left[ g_N(x^\pi_N) + \sum_{k=0}^{N-1} g_k(x^\pi_k, \mu_k(x^\pi_k), w_k) \right].$$
If the initial state is fixed, i.e., $x_0^\pi = x_0$ for all feasible policy $\pi$ then we denote the performance of policy $\pi$ by $J_\pi(x_0)$.

The objective of dynamic programming is to optimize $J_\pi$ over all policies $\pi$ that satisfy the constraints of the problem.

### 2.1.1 Markov Decision Processes

There are situations where the state $x_k$ is naturally discrete, and its evolution can be modeled by a Markov chain. In these cases, the state transition function is described by the transition probabilities matrix between the states:

$$p_{ij}(u,k) = P\{x_{k+1} = j | x_k = i, u_k = u\}$$

Claim: Transition probabilities $\iff$ System equation

**Proof:** $\Rightarrow$ Given a transition probability representation,

$$p_{ij}(u,k) = P\{x_{k+1} = j | x_k = i, u_k = u\},$$

we can cast it in terms of the basic DP framework as

$$x_{k+1} = w_k,$$  where $P\{w_k = j | x_k = i, u_k = u\} = p_{ij}(u,k).$ \quad (1)

$\Leftarrow$ Given a discrete-state system equation $x_{k+1} = f_k(x_k, u_k, w_k)$, and a probability distribution for $w_k$, $P_k(w_k|x_k, u_k)$, we can get the following transition probability representation:

$$p_{ij}(u,k) = P_k\{W_k(i,u,j)| x_k = i, u_k = u\},$$

where the event $W_k$ is defined as

$$W_k(i,u,j) = \{w | j = f_k(i,u,w)\}. \quad \square$$

#### Example 2.1.2 (Scheduling)

- **Objective:** Find the optimal sequence of operations $A, B, C, D$ to produce a certain product.
- **Precedence constraints:** $A \rightarrow B, C \rightarrow D$.
- **State definition:** Set of operations already performed.
- **Costs:** Startup costs $S_A$ and $S_B$ incurred at time $k = 0$, and setup transition costs $C_{nm}$ from operation $m$ to $n$.

This example is represented in Figure 2.1.2. The optimal solution is described by a path of minimum cost that starts at the initial state and ends at some state at the terminal time. The cost of a path is the sum of the labels in the arcs plus the terminal cost (label in the leaf). $\square$
CHAPTER 2. DISCRETE DYNAMIC PROGRAMMING

Example 2.1.3 (Chess Game)

• Objective: Find the optimal two-game chess match strategy that maximizes the winning chances.

• Description of the match:
  Each game can have two outcomes: Win (1 point for winner, 0 for loser), and Draw (1/2 point for each player)
  If the score is tied after two games, the match continues until one of them wins a game (“sudden death”).

• State: vector with the two scores attained so far. It could also be the net score (difference between the scores).

• Each player has two playing styles, and can choose one of the two at will in each game:
  – **Timid play:** draws with probability \( p_d > 0 \), and loses w.p. \( 1 - p_d \).
  – **Bold play:** wins w.p. \( p_w > 0 \), and loses w.p. \( 1 - p_w \).

• Observations: If there is a tie after the 2nd game, the player must play **Bold**. So, from an analytical perspective, the problem is a two-period one. Also note that this is not a “game theory” setting, since there is no best response here. The other player’s strategy is somehow captured by the corresponding probabilities.

Using the equivalence between system equation and transition probability function mentioned above, in Figure 2.1.3 we show the transition probabilities for period \( k = 0 \). In Figure 2.1.4 we show the transition
Note that these numbers are negative because maximizing the probability of winning $p$ is equivalent to minimizing $-p$ (recall that we are working with min problems so far). One interesting feature of this problem (to be verified later) is that even if $p_w < 1/2$, the player could still have more than 50% chance of winning the match. □

2.2 Deterministic DP and the Shortest Path Problem

In this section, we focus on deterministic problems, i.e., problems where the value of each disturbance $w_k$ is known in advance at time 0. In deterministic problems, using feedback results does not help in terms of cost reduction and hence open-loop and closed-loop policies are equivalent.

Claim: In deterministic problems, minimizing cost over admissible policies $\{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$ (i.e., sequence of functions) leads to the same optimal cost as minimizing over sequences of control vectors $\{u_0, u_1, \ldots, u_{N-1}\}$.
CHAPTER 2. DISCRETE DYNAMIC PROGRAMMING

Prof. R. Caldentey

Proof: Given a policy \( \{ \mu_0, \mu_1, \ldots, \mu_{N-1} \} \), and an initial state \( x_0 \), the future states are perfectly predictable through the equation

\[
x_{k+1} = f_k(x_k, \mu(x_k)), \quad k = 0, 1, \ldots, N - 1,
\]

and the corresponding controls are perfectly predictable through the equation

\[
u_k = \mu_k(x_k), \quad k = 0, 1, \ldots, N - 1.
\]

Thus, the cost achieved by an admissible policy \( \{ \mu_0, \mu_1, \ldots, \mu_{N-1} \} \) for a deterministic problem is also achieved by the control sequence \( \{ u_0, u_1, \ldots, u_{N-1} \} \) defined above. 

Hence, we may restrict attention to sequences of controls without loss of optimality.

2.2.1 Deterministic finite-state problem

This type of problems can be represented by a graph (see Figure 2.2.1), where:

- States \( \iff \) Nodes
- Each control applied over a state \( x_k \) \( \iff \) Arc from node \( x_k \).
  So, every outgoing arc from a node \( x_k \) represents one possible control \( u_k \). Among them, we have to choose the best one \( u_k^* \) (i.e., the one that minimizes the cost from node \( x_k \) onwards).
- Control sequences (open-loop) \( \iff \) paths from initial state \( s \) to terminal states
- Final stage \( \iff \) Artificial terminal node \( t \)
- Each state \( x_N \) at stage \( N \) \( \iff \) Connected to the terminal node \( t \) with an arc having cost \( g_N(x_N) \).
- One-step costs \( g_k(i, u_k) \) \( \iff \) Cost of an arc \( a^k_{ij} \) (cost of transition from state \( i \in S_k \) to state \( j \in S_{k+1} \) at time \( k \), viewed as the “length of the arc”) if \( u_k \) forces the transition \( i \to j \).
  Define \( a_{it} \) as the terminal cost of state \( i \in S_N \).
  Assume \( a_{it} = \infty \) if there is no control that drives from \( i \) to \( j \).
- Cost of control sequence \( \iff \) Cost of the corresponding path (view it as “length of the path”)
- The deterministic finite-state problem is equivalent to finding a shortest path from \( s \) to \( t \).

2.2.2 Backward and forward DP algorithms

The usual backward DP algorithm takes the form:

\[
J_N(i) = a_{it}^N, \quad i \in S_N,
\]

\[
J_k(i) = \min_{j \in S_{k+1}} \left[ a^k_{ij} + J_{k+1}(j) \right], \quad i \in S_k, \quad k = 0, 1, \ldots, N - 1.
\]

The optimal cost is \( J_0(s) \) and is equal to the length of the shortest path from \( s \) to \( t \).
Observation: An optimal path $s \rightarrow t$ is also an optimal path $t \rightarrow s$ in a “reverse” shortest path problem where the direction of each arc is reversed and its length is left unchanged.

The previous observation leads to the forward DP algorithm:

$$
\tilde{J}_N(j) = a_{sj}^0, \quad j \in S_1,
$$

$$
\tilde{J}_k(j) = \min_{i \in S_{N-k}} \left[ a_{ij}^{N-k} + \tilde{J}_{k+1}(i) \right], \quad j \in S_{N-k+1}, k = 0, 1, \ldots, N - 1.
$$

The optimal cost is

$$
\tilde{J}_0(t) = \min_{i \in S_N} \left[ a_{it}^N + \tilde{J}_1(i) \right].
$$

Note that both algorithms yield the same result: $J_0(s) = \tilde{J}_0(t)$. Take $\tilde{J}_k(j)$ as the optimal cost-to-arrive to state $j$ from initial state $s$.

The following observations apply to the forward DP algorithm:

- There is no forward DP algorithm for stochastic problems.
- Mathematically, for stochastic problems, we cannot restrict ourselves to open-loop sequences, so the shortest path viewpoint fails.
- Conceptually, in the presence of uncertainty, the concept of “optimal cost-to-arrive” at a state $x_k$ does not make sense. The reason is that it may be impossible to guarantee (w.p. 1) that any given state can be reached.
- By contrast, even in stochastic problems, the concept of “optimal cost-to-go” from any state $x_k$ (in expectation) makes clear sense.

**Conclusion:** A deterministic finite-state problem is equivalent to a special type of shortest path problem and can be solved by either the ordinary (backward) DP algorithm or by an alternative forward DP algorithm.
2.2.3 Generic shortest path problems

Here, we are converting a shortest path problem to a deterministic finite-state problem. More formally, given a graph, we want to compute the shortest path from each node $i$ to the final node $t$. How to cast this into the DP framework?

- Let $\{1, 2, \ldots, N, t\}$ be the set of nodes of a graph, where $t$ is the destination node.
- Let $a_{ij}$ be the cost of moving from node $i$ to node $j$.
- Objective: Find a shortest (minimum cost) path from each node $i$ to node $t$.
- Assumption: All cycles have nonnegative length. Then, an optimal path need not take more than $N$ moves (depth of a tree).
- We formulate the problem as one where we require exactly $N$ moves but allow degenerate moves from a node $i$ to itself with cost $a_{ii} = 0$.
- In terms of the DP framework, we propose a formulation with $N$ stages labeled 0, 1, \ldots, $N-1$. Denote:
  
  $J_k(i) = \text{Optimal cost of getting from } i \text{ to } t \text{ in } N - k \text{ moves}$
  
  $J_0(i) = \text{Cost of the optimal path from } i \text{ to } t \text{ in } N \text{ moves}$
  
- DP algorithm:
  
  $J_k(i) = \min_{j = 1, 2, \ldots, N} \{a_{ij} + J_{k+1}(j)\}, \ k = 0, 1, \ldots, N - 2,$
  
  with $J_{N-1}(i) = a_{it}, \ i = 1, 2, \ldots, N$.
  
- The optimal policy when at node $i$ after $k$ moves is to move to a node $j^*$ such that
  
  $j^* = \arg\min_{1 \leq j \leq N} \{a_{ij} + J_{k+1}(j)\}$
  
- If the optimal path from the algorithm contains degenerate moves from a node to itself, it means that the path in reality involves less than $N$ moves.

**Demonstration of the algorithm**

Consider the problem exhibited in Figure 2.2.2 where the costs $a_{ij}$ with $i \neq j$ are shown along the connecting line segments. The graph is represented as a non-directed one, meaning that the arc costs are the same in both directions, i.e., $a_{ij} = a_{ji}$.

**Running the algorithm:**

In this case, we have $N = 4$, so it is a 3-stage problem with 4 states:

1. Starting from stage $N - 1 = 3$, we compute $J_{N-1}(i) = a_{it}$, for $i = 1, 2, 3, 4$ and $t = 5$:
   
   $J_3(1) = \text{cost of getting from node 1 to node 5} = 2$
   
   $J_3(2) = \text{cost of getting from node 2 to node 5} = 7$
   
   $J_3(3) = \text{cost of getting from node 3 to node 5} = 5$
   
   $J_3(4) = \text{cost of getting from node 4 to node 5} = 3$

   The numbers above represent the cost of getting from $i$ to $t$ in $N - (N - 1) = 1$ move.
Figure 2.2.2: Shortest path problem data. There are $N = 4$ states, and a destination node $t = 5$.

2. Proceeding backwards to stage $N - 2 = 2$, we have:

$$J_2(1) = \min_{j=1,2,3,4} \{a_{1j} + J_3(j)\} = \min \{a_{11} + J_3(1), a_{12} + J_3(2), a_{13} + J_3(3), a_{14} + J_3(4)\} = 2$$

$$J_2(2) = \min_{j=1,2,3,4} \{a_{2j} + J_3(j)\} = \min \{a_{21} + J_3(1), a_{22} + J_3(2), a_{23} + J_3(3), a_{24} + J_3(4)\} = 5.5$$

$$J_2(3) = \min_{j=1,2,3,4} \{a_{3j} + J_3(j)\} = \min \{a_{31} + J_3(1), a_{32} + J_3(2), a_{33} + J_3(3), a_{34} + J_3(4)\} = 4$$

$$J_2(4) = \min_{j=1,2,3,4} \{a_{4j} + J_3(j)\} = \min \{a_{41} + J_3(1), a_{42} + J_3(2), a_{43} + J_3(3), a_{44} + J_3(4)\} = 3$$

3. Proceeding backwards to stage $N - 3 = 1$, we have:

$$J_1(1) = \min_{j=1,2,3,4} \{a_{1j} + J_2(j)\} = \min \{a_{11} + J_2(1), a_{12} + J_2(2), a_{13} + J_2(3), a_{14} + J_2(4)\} = 2$$

$$J_1(2) = \min_{j=1,2,3,4} \{a_{2j} + J_2(j)\} = \min \{a_{21} + J_2(1), a_{22} + J_2(2), a_{23} + J_2(3), a_{24} + J_2(4)\} = 4.5$$

$$J_1(3) = \min_{j=1,2,3,4} \{a_{3j} + J_2(j)\} = \min \{a_{31} + J_2(1), a_{32} + J_2(2), a_{33} + J_2(3), a_{34} + J_2(4)\} = 4$$

$$J_1(4) = \min_{j=1,2,3,4} \{a_{4j} + J_2(j)\} = \min \{a_{41} + J_2(1), a_{42} + J_2(2), a_{43} + J_2(3), a_{44} + J_2(4)\} = 3$$

4. Finally, proceeding backwards to stage 0, we have:

$$J_0(1) = \min_{j=1,2,3,4} \{a_{1j} + J_1(j)\} = \min \{a_{11} + J_1(1), a_{12} + J_1(2), a_{13} + J_1(3), a_{14} + J_1(4)\} = 2$$

$$J_0(2) = \min_{j=1,2,3,4} \{a_{2j} + J_1(j)\} = \min \{a_{21} + J_1(1), a_{22} + J_1(2), a_{23} + J_1(3), a_{24} + J_1(4)\} = 4.5$$

$$J_0(3) = \min_{j=1,2,3,4} \{a_{3j} + J_1(j)\} = \min \{a_{31} + J_1(1), a_{32} + J_1(2), a_{33} + J_1(3), a_{34} + J_1(4)\} = 4$$

$$J_0(4) = \min_{j=1,2,3,4} \{a_{4j} + J_1(j)\} = \min \{a_{41} + J_1(1), a_{42} + J_1(2), a_{43} + J_1(3), a_{44} + J_1(4)\} = 3$$
CHAPTER 2. DISCRETE DYNAMIC PROGRAMMING

Prof. R. Caldentey

Figure 2.2.3 shows the outcome of the shortest path (DP-type) algorithm applied over the graph in Figure 2.2.2.

Figure 2.2.3: Outcome of the shortest path algorithm. The arcs represent the optimal control to follow from a given state $i$ (node $i$ in the graph) at a particular stage $k$ (where stage $k$ means $4 - k$ transitions left to reach $t = 5$). When there is more than one arc going out of a node, it represents the availability of more than one optimal control. The label next to each node shows the cost-to-go starting at the corresponding (stage, state) position.

2.2.4 Some shortest path applications

Hidden Markov models and the Viterbi algorithm

Consider a Markov chain for which we do not observe the outcome of the transitions but rather we observe a signal or proxy that relates to that transition. The setting of the problem is the following:

- Markov chain (discrete time, finite number of states) with transition probabilities $p_{ij}$.
- State transition are hidden from view.
- For each transition, we get an independent observation.
- Denote $\pi_i$: Probability that the initial state is $i$.
- Denote $r(z;i,j)$: Probability that the observation takes the value $z$ when the state transition is from $i$ to $j$.$^3$
- Trajectory estimation problem: Given the observation sequence $Z_N = \{z_1, z_2, \ldots, z_N\}$, what is the most likely (unobservable) transition sequence $\hat{X}_N = \{\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_N\}$? More formally: We are looking for the transition sequence $\hat{X}_N = \{\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_N\}$ that maximizes $p(X_N|Z_N)$ over all $X_N = \{x_0, x_1, \ldots, x_N\}$. We are using the notation $\hat{X}_N$ to emphasize the fact that this is an estimated sequence. We do not observe the true sequence, but just a proxy for it given by $Z_N$.

$^3$The probabilities $p_{ij}$ and $r(z;i,j)$ are assumed to be independent of time for notational convenience, but the methodology could be extended to time-dependent probabilities.
Viterbi algorithm

We know from conditional probability that
\[
\mathbb{P}(X_N|Z_N) = \frac{\mathbb{P}(X_N, Z_N)}{\mathbb{P}(Z_N)},
\]
for unconditional probabilities \(\mathbb{P}(X_N, Z_N)\) and \(\mathbb{P}(Z_N)\). Since \(\mathbb{P}(Z_N)\) is a positive constant once \(Z_N\) is known, we can just maximize \(\mathbb{P}(X_N, Z_N)\), where

\[
\mathbb{P}(X_N, Z_N) = \mathbb{P}(x_0, x_1, \ldots, x_N, z_1, z_2, \ldots, z_N) = \pi_0 \mathbb{P}(x_1, \ldots, x_N, z_1, z_2, \ldots, z_N|x_0) = \pi_0 \mathbb{P}(x_1, z_1|x_0) \mathbb{P}(x_2, \ldots, x_N, z_2, \ldots, z_N|x_0, x_1, z_1) = \pi_0 p_{x_0x_1} r(z_1|x_0, x_1) \mathbb{P}(x_2, \ldots, x_N, z_2, \ldots, z_N|x_0, x_1, z_1) \cdot
\]

Continuing in the same manner we obtain:
\[
\mathbb{P}(X_N, Z_N) = \pi_0 \prod_{k=1}^{N} p_{x_{k-1}x_k} r(z_k; x_{k-1}, x_k)
\]
Instead of working with this function, we will maximize \(\log \mathbb{P}(X_N, Z_N)\), or equivalently:
\[
\min_{x_0, x_1, \ldots, x_N} \left\{ -\log(\pi_0) - \sum_{k=1}^{N} \log(p_{x_{k-1}x_k} r(z_k; x_{k-1}, x_k)) \right\}
\]

The outcome of this minimization problem will be the sequence \(\hat{X}_N = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N\}\).

**Transformation into a shortest path problem in a trellis diagram**

We build the trellis diagram shown in Figure 2.2.4 as follows:

- Arc \((s, x_0)\) \rightarrow Cost = −\log(\pi_0).
- Arc \((x_N, t)\) \rightarrow Cost = 0.
- Arc \((x_{k-1}, x_k)\) \rightarrow Cost = −\log(p_{x_{k-1}x_k} r(z_k; x_{k-1}, x_k)).

The shortest path defines the estimated state sequence \(\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_N\).

In practice, the shortest path is most conveniently constructed sequentially by forward DP: Suppose that we have already computed the shortest distances \(D_k(x_k)\) from \(s\) to all states \(x_k\), on the basis of the observation sequence \(z_1, \ldots, z_k\), and suppose that we observe \(z_k+1\). Then:
\[
D_{k+1}(x_{k+1}) = \min_{\{x_k: p_{x_kx_{k+1}} > 0\}} \left\{ D_k(x_k) - \log(p_{x_kx_{k+1}} r(z_{k+1}; x_k, x_{k+1})) \right\}, \quad k = 1, \ldots, N - 1,
\]
starting from \(D_0(x_0) = -\log(\pi_0)\).

Observations:

- Final estimated sequence \(\hat{X}_N\) corresponds to the shortest path from \(s\) to the final state \(\hat{x}_N\) that minimizes \(D_N(x_N)\) over the final set of possible states \(x_N\).
Figure 2.2.4: State estimation of a hidden Markov model viewed as a problem of finding a shortest path from $s$ to $t$. There are $N + 1$ copies of the state space (recall that the number of states is finite). So, $x_k$ stands for any state in the copy $k$ of the state space. An arc connects $x_{k-1}$ with $x_k$ if $p_{x_{k-1}x_k} > 0$.

- **Advantage:** It can be computed in real time, as new observations arrive.
- **Applications of the Viterbi algorithm:**
  - Speech recognition, where the goal is to transcribe a spoken word sequence in terms of elementary speech units called “phonemes”.
  - **Setting:**
    - States of the hidden Markov model: phonemes.
    - Given a sequence of recorded phonemes $Z_N = \{z_1, \ldots, z_N\}$ (i.e., a noisy representation of words) try to find a phonemic sequence $\hat{X}_N = \{\hat{x}_1, \ldots, \hat{x}_N\}$ that maximizes over all possible $X_N = \{x_1, \ldots, x_N\}$ the conditional probability $P(X_N|Z_N)$.
    - The probabilities $p_{x_{k-1}x_k}$ and $r(z_k; x_{k-1}, x_k)$ can be experimentally obtained.
  - Computerized recognition of handwriting.

### 2.2.5 Shortest path algorithms

Computational implications of the equivalence $\text{shortest path problems} \iff \text{deterministic finite-state DP}$:

- **We can use DP to solve general shortest path problems.**
  
  Although there are other methods with superior worst-case performance, DP could be preferred because it is highly parallelizable.

- **There are many non-DP shortest path algorithms that can be used to solve deterministic finite-state problems.**
They may be preferable than DP if they avoid calculating the optimal cost-to-go at every state.

This is essential for problems with huge state spaces (e.g., combinatorial optimization problems).

Example 2.2.1 (An Example with very large number of nodes: TSP)

The Traveling Salesman Problem (TSP) is about finding a tour (cycle) that passes exactly once for each city (node) of a graph, and that minimizes the total cost. Consider for instance the problem described in Figure 2.2.5.

![Figure 2.2.5: Basic graph for the TSP example with four cities.](image)

To convert a TSP problem over a map (graph) with \( N \) nodes to a shortest path problem, build a new execution graph as follows:

- Pick a city and set it as the initial node \( s \).
- Associate a node with every sequence of \( n \) distinct cities, \( n \leq N \).
- Add an artificial terminal node \( t \).
- A node representing a sequence of cities \( c_1, c_2, \ldots, c_n \) is connected with a node representing a sequence \( c_1, c_2, \ldots, c_n, c_{n+1} \) with an arc with weight \( a_{c_n c_{n+1}} \) (length of the arc in the original graph).
- Each sequence of \( N \) cities is connected to the terminal node through an arc with same cost as the cost of the arc connecting the last city of the sequence and city \( s \) in the original graph.

Figure 2.2.6 shows the construction of the execution graph for the example described in Figure 2.2.5.

2.2.6 Alternative shortest path algorithms: Label correcting methods

Working on the shortest path execution graph as the one in Figure 2.2.6, the idea of these methods is to progressively discover shorter paths from the origin \( s \) to every other node \( i \).

- Given: Origin \( s \), destination \( t \), lengths \( a_{ij} \geq 0 \).
Figure 2.2.6: Structure of the shortest path execution graph for the TSP example.

- Notation:
  - Label $d_i$: Length of the shortest path found (initially $d_s = 0$, $d_i = \infty$ for $i \neq s$).
  - Variable UPPER: Label $d_t$ of the destination.
  - Set OPEN: Contains nodes that are currently active in the sense that they are candidates for further examination (initially, OPEN:=\{s\}). It is sometimes called candidate list.
  - Function ParentOf($j$): Saves the predecessor of $j$ in the shortest path found so far from $s$ to $j$. At the end of the algorithm, proceeding backward from node $t$, it allows to rebuild the shortest path from $s$.

Label Correcting Algorithm (LCA)

Step 1 Node removal: Remove a node $i$ from OPEN and for each child $j$ of $i$, do Step 2.

Step 2 Node insertion test: If $d_i + a_{ij} < \min\{d_j, \text{UPPER}\}$, set $d_j := d_i + a_{ij}$ and set $i := \text{ParentOf}(j)$.

In addition, if $j \neq t$, set OPEN:=OPEN\{j\}; while if $j = t$, set UPPER:= $d_t$.

Step 3 Termination test: If OPEN is empty, terminate; else go to Step 1.

As a clarification for Step 2, note that since OPEN is a set, if $j, j \neq t$, is already in OPEN, then OPEN remains the same. Also, when $j = t$, note that UPPER takes the new value $d_t = d_i + a_{it}$ that has just been updated. Figure 2.2.7 sketches the Label Correcting Algorithm.

The execution of the algorithm over the TSP example above is represented in Figure 2.2.8 and Table 2.1. Interestingly, note that several nodes of the execution graph never enter the OPEN set. Indeed, this computational reduction with respect to DP is what makes this method appealing.

The following proposition establishes the validity of the Label Correcting Algorithm.
Figure 2.2.7: Sketch of the Label Correcting Algorithm.

Table 2.1: The optimal solution ABCD is found after examining nodes 1 through 10 in Figure 2.2.8, in that order. The table also shows the successive contents of the OPEN list, the value of UPPER at the end of an iteration, and the actions taken during each iteration.

Proposition 2.2.1 If there exists at least one path from the origin to the destination in the execution graph, the label correcting algorithm terminates with UPPER equal to the shortest distance from the origin to the destination. Otherwise, the algorithm terminates with UPPER=∞.

Proof: We proceed in three steps:

1. The algorithm terminates
   Each time a node \( j \) enters OPEN, its label \( d_j \) is decreased and becomes equal to some path from \( s \) to \( j \). The number of distinct lengths of paths from \( s \) to \( j \) that are smaller than any given number is finite. Hence, there can only be a finite number of label reductions.

2. Suppose that there is no path \( s \rightarrow t \).
Then a node \( i \) such that \((i, t)\) is an arc cannot enter the OPEN list, because if that happened, since the paths are built starting from \( s \), it would mean that there is a path \( s \to i \), which jointly with the arc \((i, t)\) would determine a path \( s \to t \), which is a contradiction. Since this holds for all \( i \) adjacent to \( t \) in the basic graph, UPPER can never be reduced from its initial value \( \infty \).

3. Suppose that there is a path \( s \to t \). Then, there is a shortest path \( s \to t \). Let \((s, j_1, j_2, \ldots, j_k, t)\) be a shortest path, and let \( d^* \) be the corresponding shortest distance. We will see that UPPER=\( d^* \) upon termination.

Each subpath \((s, j_1, j_2, \ldots, j_m)\), \( m = 1, \ldots, k \), must be a shortest path \( s \to j_m \). If UPPER\(> d^* \) at termination, then same occurs throughout the algorithm (because UPPER is decreasing during the execution). So, UPPER is bigger than the length of all paths \( s \to j_m \) (due to the nonnegative arc length assumption).

In particular, node \( j_k \) will never enter the OPEN list with \( d_{jk} \) equal to the shortest distance \( s \to j_k \). To see this, suppose \( j_k \) enters OPEN. When at some point the algorithm picks \( j_k \) from OPEN, it will set \( d_t = d_{jk} + a_{jt} \), and UPPER= \( d^* \).

Similarly, node \( j_{k-1} \) will never enter OPEN with \( d_{jk-1} \) equal to the shortest distance \( s \to j_{k-1} \). Proceeding backward, \( j_1 \) never enters OPEN with \( d_{j_1} \) equal to the shortest distance \( s \to j_1 \); i.e. \( a_{s j_1} \). However, this happens at the first iteration of the algorithm, leading to a contradiction.

Therefore, UPPER will be equal to the shortest distance \( s \to t \). \( \blacksquare \)
Specific Label Correcting Methods

Making the method efficient:

- Reduce the value of UPPER as quickly as possible (i.e., try to discover “good” $s \rightarrow t$ paths early in the course of the algorithm).

- Keep the number of reentries into OPEN low.
  
  - Try to remove from OPEN nodes with small label first.
  
  - Heuristic rationale: if $d_i$ is small, then $d_j$ when set to $d_i + a_{ij}$ will be accordingly small, so reentrance of $j$ in the OPEN list is less likely.

- Reduce the overhead for selecting the node to be removed from OPEN.

- These objectives are often in conflict. They give rise to a large variety of distinct implementations.

Node selection methods:

- **Breadth-first search**: Also known as the Bellman-Ford method. The set OPEN is treated as an ordered list. FIFO policy.

- **Depth-first search**: The set OPEN is treated as an ordered list. LIFO policy. It often requires relatively little memory, specially for sparse (i.e., tree-like) graphs. It reduces UPPER quickly.

- **Best-first search**: Also known as the Dijkstra method. Remove from OPEN a node $j$ with minimum value of label $d_j$. In this way, each node will be inserted in OPEN at most once.

Advanced initialization:

In order to get a small starting value of UPPER, instead of starting from $d_i = \infty$ for all $i \neq s$, we can initialize the value of the labels $d_i$ and the set OPEN as follows:

Start with

$$d_i := \text{length of some path from } s \text{ to } i.$$ 

If there is no such path, set $d_i = \infty$. Then, set OPEN:=\{ $i \neq t | d_i < \infty$\}.

- No node with shortest distance greater or equal than the initial value of UPPER will enter OPEN.

- Good practical idea:
  
  - Run a heuristic to get a “good” starting path $P$ from $s$ to $t$.
  
  - Use as UPPER the length of $P$, and as $d_i$ the path distances of all nodes $i$ along $P$. 

58
2.2.7 Exercises

Exercise 2.2.1 A decision maker must continually choose between two activities over a time interval $[0, T]$. Choosing activity $i$ at time $t$, where $i = 1, 2$, earns reward at a rate $g_i(t)$, and every switch between the two activities costs $c > 0$. Thus, for example, the reward for starting with activity 1, switching to 2 at time $t_1$, and switching back to 1 at time $t_2 > t_1$, earns total reward

$$
\int_0^{t_1} g_1(t) \, dt + \int_{t_1}^{t_2} g_2(t) \, dt + \int_{t_2}^{T} g_1(t) \, dt - 2c
$$

We want to find a set of switching times that maximize the total reward. Assume that the function $g_1(t) - g_2(t)$ changes sign a finite number of times in the interval $[0, T]$. Formulate the problem as a finite horizon problem and write the corresponding DP algorithm.

Exercise 2.2.2 Assume that we have a vessel whose maximum weight capacity is $z$ and whose cargo is to consist of different quantities of $N$ different items. Let $v_i$ denote the value of the $i$th type of the item, $w_i$ the weight of $i$th type of item, and $x_i$ the number of items of type $i$ that are loaded in the vessel. The problem is to find the most valuable cargo subject to the capacity constraint. Formulate this problem in terms of DP.

Exercise 2.2.3 Find a shortest path from each node to node 6 for the graph below by using the DP algorithm:

Exercise 2.2.4 Air transportation is available between $n$ cities, in some cases directly and in others through intermediate stops and change of carrier. The airfare between cities $i$ and $j$ is denoted by $a_{ij}$. We assume that $a_{ij} = a_{ji}$, and for notational convenience, we write $a_{ij} = \infty$ if there is no direct flight between $i$ and $j$. The problem is to find the cheapest airfare for going between two cities perhaps through intermediate stops. Let $n = 6$ and

$$
a_{12} = 30, \quad a_{13} = 60, \quad a_{14} = 25, \quad a_{15} = a_{16} = \infty, \\
a_{23} = a_{24} = a_{25} = \infty, \quad a_{26} = 50, \\
a_{34} = 35, \quad a_{35} = a_{36} = \infty, \\
a_{45} = 15, \quad a_{46} = \infty, \\
a_{56} = 15.
$$

Find the cheapest airfare from every city to every other city by using the DP algorithm.
Exercise 2.2.5 Label correcting with negative arc lengths. Consider the problem of finding a shortest path from node \( s \) to node \( t \), and assume that all cycle lengths are nonnegative (instead of all arc lengths being nonnegative). Suppose that a scalar \( u_j \) is known for each node \( j \), which is an underestimate of the shortest distance from \( j \) to \( t \) (\( u_j \) can be taken \( -\infty \) if no underestimate is known). Consider a modified version of the typical iteration of the label correcting algorithm discussed above, where Step 2 is replaced by the following:

Modified Step 2: If 

\[
d_i + a_{ij} < \min\{d_j, \text{UPPER} - u_j\},
\]

set 

\[
d_j = d_i + a_{ij} \quad \text{and set } i := \text{ParentOf}(j).
\]

In addition, if \( j \neq t \), place \( j \) in OPEN if it is not already in OPEN, while if \( j = t \), set \( \text{UPPER} \) to the new value \( d_i + a_{it} \) of \( d_t \).

1. Show that the algorithm terminates with a shortest path, assuming there is at least one path from \( s \) to \( t \).

2. Why is the Label Correcting Algorithm given in class a special case of the one here?

Exercise 2.2.6 We have a set of \( N \) objects, denoted \( 1, 2, \ldots, N \), which we want to group in clusters that consist of consecutive objects. For each cluster \( i, i+1, \ldots, j \), there is an associated cost \( a_{ij} \). We want to find a grouping of the objects in clusters such that the total cost is minimum. Formulate the problem as a shortest path problem, and write a DP algorithm for its solution. (Note: An example of this problem arises in typesetting programs, such as \( \text{TEX/LATEX} \), that break down a paragraph into lines in a way that optimizes the paragraph’s appearance).

2.3 Stochastic Dynamic Programming

We present here a general problem of decision making under stochastic uncertainty over a finite number of stages. The components of the formulation are listed below:

- The discrete time dynamic system evolves according to the system equation

\[
x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, \ldots, N - 1,
\]

where

- the state \( x_k \) is an element of a space \( S_k \),
- the control \( u_k \) verifies \( u_k \in U_k(x_k) \subset C_k \), for a space \( C_k \), and
- the random disturbance \( w_k \) is an element of a space \( D_k \).

- The random disturbance \( w_k \) is characterized by a probability distribution \( P_k(\cdot|x_k, u_k) \) that may depend explicitly on \( x_k \) and \( u_k \). For now, we assume independent disturbances \( w_0, w_1, \ldots, w_{N-1} \). In this case, since the system evolution from state to state is independent of the past, we have a Markov decision model.

- Admissible policies \( \pi = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\} \), where \( \mu_k \) maps states \( x_k \) into controls \( u_k = \mu_k(x_k) \), and is such that \( \mu_k(x_k) \in U_k(x_k) \) for all \( x_k \in S_k \).
• Given an initial state $x_0$ and an admissible policy $\pi = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$, the states $x_k$ and disturbances $w_k$ are random variables with distributions defined through the system equation

$$x_{k+1} = f_k(x_k, \mu_k(x_k), w_k), \quad k = 0, 1, \ldots, N - 1,$$

• The stage cost function is given by $g_k(x_k, \mu_k(x_k), w_k)$.

• The expected cost of a policy $\pi$ starting at $x_0$ is

$$J_\pi(x_0) = E\left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k)\right],$$

where the expectation is taken over the r.v. $w_k$ and $x_k$.

An optimal policy $\pi^*$ is one that minimizes this cost; that is,

$$J_{\pi^*}(x_0) = \min_{\pi \in \Pi} J_\pi(x_0), \quad (2.3.1)$$

where $\Pi$ is the set of all admissible policies.

Observations:

• It is useful to see $J^*$ as a function that assigns to each initial state $x_0$ the optimal cost $J^*(x_0)$, and call it the optimal cost function or optimal value function.

• When produced by DP, $\pi^*$ is typically independent of $x_0$, because

$$\pi^* = \{\mu_0^*(x_0), \ldots, \mu_{N-1}^*(x_{N-1})\},$$

and $\mu_0^*(x_0)$ must be defined for all $x_0$.

• Even though we will be using the “min” operator in $(2.3.1)$, it should be understood that the correct formal formulation should be $J_{\pi^*}(x_0) = \inf_{\pi \in \Pi} J_\pi(x_0)$.

Example 2.3.1 (Control of a single server queue)

Consider a single server queueing system with the following features:

• Waiting room for $n - 1$ customers; an arrival finding $n$ people in the system ($n - 1$ waiting and one in the server) leaves.

• Discrete random service time belonging to the set $\{1, 2, \ldots, N\}$.

• Probability $p_m$ of having $m$ arrivals at the beginning of a period, with $m = 0, 1, 2, \ldots$.

• System offers two types of service, that can be chosen at the beginning of each period:
  - Fast, with cost per period $c_f$, and that finishes at the end of the current period w.p. $q_f$.
  - Slow, with cost per period $c_s$, and that finishes at the end of the current period w.p. $q_s$. 

61
Assume \( q_f > q_s \) and \( c_f > c_s \).

- A recent arrival cannot be immediately served.
- The system incurs two costs in every period: The service cost (either \( c_f \) or \( c_s \)), and a waiting time cost \( r(i) \) if there are \( i \) customers waiting at the beginning of a period.
- There is a terminal cost \( R(i) \) if there are \( i \) customers waiting in the final period (i.e., in period \( N \)).
- Problem: Choose the type of service at the beginning of each period (fast or slow) in order to minimize the total expected cost over \( N \) periods.

Intuitive optimal strategy must be of the threshold type: “When there are more than \( i \) customers in the system use fast; otherwise, use slow”.

In terms of DP terminology, we have:

- State \( x_k \): Number of customers in the system at the start of period \( k \).
- Control \( u_k \): Type of service provided; either \( u_k = u_f \) (fast) or \( u_k = u_s \) (slow).
- Cost per period \( k \): For \( 0 \leq k \leq N - 1 \), \( g_k(i, u_k, w_k) = r(i) + c_f 1_{\{u_k = u_f\}} + c_s 1_{\{u_k = u_s\}} \).
  \( g_N(i) = R(i) \).

According to the claim above, since states are discrete, transition probabilities are enough to describe the system dynamics:

- If the system is empty, then:
  \[ p_{0j}(u_f) = p_{0j}(u_s) = p_j, \quad j = 0, 1, \ldots, n - 1; \quad \text{and} \quad p_{0n}(u_f) = p_{0n}(u_s) = \sum_{m=n}^{\infty} p_m \]
  In words, since customers cannot be served immediately, they accumulate and system jumps to state \( j < n \) (if there were less than \( n \) arrivals), or to state \( n \) (if there are \( n \) or more).
- When the system is not empty (i.e., \( x_k = i > 0 \)), then
  \[ p_{ij}(u_f) = 0, \quad \text{if} \quad j < i - 1 \quad \text{(we cannot have more than one service completion per period)}, \]
  \[ p_{ij}(u_f) = q_f p_0, \quad \text{if} \quad j = i - 1 \quad \text{(current customer finishes in this period and nobody arrives)}, \]
  \[ p_{ij}(u_f) = \mathbb{P}\{j - i + 1 \text{ arrivals, service completed}\} + \mathbb{P}\{j - i \text{ arrivals, service not completed}\} \]
  \[ = q_f p_{j-i+1} + (1 - q_f) p_{j-i}, \quad \text{if} \quad i - 1 < j < n - 1, \]
  \[ p_{i(n-1)}(u_f) = \mathbb{P}\{\text{at least } n - i \text{ arrivals, service completed}\} \]
  \[ + \mathbb{P}\{n - 1 - i \text{ arrivals, service uncompleted}\} \]
  \[ = q_f \sum_{m=n-i}^{\infty} p_m + (1 - q_f) p_{n-1-i}, \]
  \[ p_{in}(u_f) = \mathbb{P}\{\text{at least } n - i \text{ arrivals, service not completed}\} \]
  \[ = (1 - q_f) \sum_{m=n-i}^{\infty} p_m \]

For control \( u_s \) the formulas are analogous, with \( u_s \) and \( q_s \) replacing \( u_f \) and \( q_f \), respectively. □

\( 4 \)Here, \( 1_{\{A\}} \) is the indicator function, taking the value one if event \( A \) occurs, and zero otherwise.
2.4 The Dynamic Programming Algorithm

The DP technique rests on a very intuitive idea, the Principle of Optimality. The name is due to Bellman (New York 1920 - Los Angeles 1984).

Principle of optimality. Let \( \pi^* = \{\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\} \) be an optimal policy for the basic problem, and assume that when using \( \pi^* \) a given state \( x_i \) occurs at time \( i \) with some positive probability. Consider the “tail subproblem” whereby we are at \( x_i \) at time \( i \) and wish to minimize the cost-to-go from time \( i \) to time \( N \),

\[
E \left[ g_N(x_N) + \sum_{k=i}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right].
\]

Then, the truncated (tail) policy \( \{\mu_i^*, \mu_{i+1}^*, \ldots, \mu_{N-1}^*\} \) is optimal for this subproblem.

Figure 2.4.1 illustrates the intuition of the Principle of Optimality. The DP algorithm is based on this idea: It first solves all tail subproblems of final stage, and then proceeds backwards solving all tail subproblems of a given time length using the solution of the tail subproblems of shorter time length. Next, we introduce the DP algorithm with two examples.

Solution to Scheduling Example 2.1.2:

Consider the graph of costs for previous Example 2 given in Figure 2.4.2. Applying the DP algorithm from the terminal nodes (stage \( N = 3 \)), and proceeding backwards, we get the representation in Figure 2.4.3. At each state-time pair, we record the optimal cost-to-go and the optimal decision. For example, node \( AC \) has a cost of 5, and the optimal decision is to proceed to \( ACB \) (because it has the lowest stage \( k = 3 \) cost, starting from \( k = 2 \) and state \( AC \)). In terms of our formal notation for the cost, \( g_3(ACB) = 1 \), for a terminal state \( x_3 = ACB \).

Solution to Inventory Example 2.1.1:

Consider again the stochastic inventory problem described in Example 1. The application of the DP algorithm is very similar to the deterministic case, except for the fact that now costs are computed as expected values.
• Tail subproblems of length 1: The optimal cost for the last period is

\[ J_{N-1}(x_{N-1}) = \min_{u_{N-1} \geq 0} E_{w_{N-1}} \left[ c u_{N-1} + r(x_{N-1} + u_{N-1} - w_{N-1}) \right] \]

Note:
- \( u_{N-1} = \mu_{N-1}(x_{N-1}) \) depends on \( x_{N-1} \).
- \( J_{N-1}(x_{N-1}) \) may be computed numerically and stored as a column of a table.

• Tail subproblems of length \( N - k \): The optimal cost for period \( k \) is

\[
J_k(x_k) = \min_{u_k \geq 0} \mathbb{E}_{w_k} \left[ c u_k + r(x_k + u_k - w_k) + J_{k+1}(x_k + u_k - w_k) \right],
\]

where \( x_{k+1} = x_k + u_k - w_k \) is the initial inventory of the next period.

• The value \( J_0(x_0) \) is the optimal expected cost when the initial stock at time 0 is \( x_0 \).

If the number of attainable states \( x_k \) is discrete with finite support \([0, S]\), the output of the DP algorithm could be stored in two tables (one for the optimal cost \( J_k \), and one for the optimal control \( u_k \)), each table consisting of \( N \) columns labeled from \( k = 0 \) to \( k = N - 1 \), and \( S + 1 \) rows labeled from 0 to \( S \). The tables are filled by the DP algorithm from right to left.

**DP algorithm:** Start with

\[ J_N(x_N) = g_N(x_N), \]

and go backwards using the recursion

\[
J_k(x_k) = \min_{u_k \in U_k(x_k)} \mathbb{E}_{w_k} \left[ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right], \quad k = 0, 1, \ldots, N - 1, \tag{2.4.1}
\]

where the expectation is taken with respect to the probability distribution of \( w_k \), which may depend on \( x_k \) and \( u_k \).

**Proposition 2.4.1** For every initial state \( x_0 \), the optimal cost \( J^*(x_0) \) of the basic problem is equal to \( J_0(x_0) \), given by the last step of the DP algorithm. Furthermore, if \( u^*_k = \mu^*_k(x_k) \) minimizes the RHS of (2.4.1) for each \( x_k \) and \( k \), the policy \( \pi^* = \{\mu^*_0, \ldots, \mu^*_{N-1}\} \) is optimal.
Figure 2.4.3: Transition graph for Example 2. Next to each node/state we show the cost of optimally completing the scheduling starting from that state. This is the optimal cost of the corresponding tail subproblem. The optimal cost for the original problem is equal to 10. The optimal schedule is \textit{CABD}.

Observations:

- Justification: Proof by induction that $J_k(x_k)$ is equal to $J^*_k(x_k)$, defined as the optimal cost of the tail subproblem that starts at time $k$ at state $x_k$.

- All the tail subproblems are solved in addition to the original problem. Observe the intensive computational requirements. The worst-case computational complexity is $\sum_{k=0}^{N-1} |S_k| |U_k|$, where $|S_k|$ is the size of the state space in period $k$, and $|U_k|$ is the size of the control space in period $k$. In particular, note that potentially we could need to search over the whole control space, although we just store the optimal one in each period-state pair.

Proof of Proposition 2.4.1

For this version of the proof, we need the following additional assumptions:

- The disturbance $w_k$ takes a finite or countable number of values
- The expected values of all stage costs are finite for every admissible policy $\pi$
- The functions $J_k(x_k)$ generated by the DP algorithm are finite for all states $x_k$ and periods $k$.

Informal argument

Let $\pi_k = \{\mu_k, \mu_{k+1}, \ldots, \mu_{N-1}\}$ denote a tail policy from time $k$ onward.

- Border case: For $k = N$, define $J^*_N(x_N) = J_N(x_N) = g_N(x_N)$. 


For any admissible policy \( \pi \) generated by the DP algorithm and the optimal \( J^*_{k+1}(x_{k+1}) \), assume that \( J_{k+1}(x_{k+1}) = J^*_{k+1}(x_{k+1}) \). Then

\[
J_k^* (x_k) = \min_{(\mu_k, \pi_{k+1})} E_{w_k,w_{k+1},\ldots,w_{N-1}} \left[ g_k(x_k, \mu_k(x_k), w_k) + g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right] \\
= \min_{\mu_k} E_{w_k} \left[ g_k(x_k, \mu_k(x_k), w_k) + \min_{\pi_{k+1},\ldots,w_{N-1}} \left[ g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right] \right]
\]

(this is the “informal step”, since we are moving the min inside the \( E[\cdot] \))

\[
= \min_{\mu_k} \left[ g_k(x_k, \mu_k(x_k), w_k) + J^*_{k+1}(f_k(x_k, \mu_k(x_k), w_k)) \right] \quad \text{(by def. of } J^*_{k+1} \text{)}
\]

\[
= \min_{\mu_k} \left[ g_k(x_k, \mu_k(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k(x_k), w_k)) \right] \quad \text{(by induction hypothesis (III))}
\]

\[
= \min_{u_k \in U_k(x_k)} E_{w_k} \left[ g_k(x_k, \mu_k(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k(x_k), w_k)) \right] \\
= J_k(x_k),
\]

where the second to last equality follows from converting a minimization problem over functions \( \mu_k \) to a minimization problem over scalars \( u_k \). In symbols, for any function \( F \) of \( x \) and \( u \), we have

\[
\min_{\mu \in M} F(x, \mu(x)) = \min_{u \in U(x)} F(x, u),
\]

where \( M \) is the set of all functions \( \mu(x) \) such that \( \mu(x) \in U(x) \) for all \( x \). 

\[\blacksquare\]

A more formal argument

For any admissible policy \( \pi = \{\mu_0, \ldots, \mu_{N-1}\} \) and each \( k = 0, 1, \ldots, N-1 \), denote

\[
\pi_k = \{\mu_k, \mu_{k+1}, \ldots, \mu_{N-1}\}.
\]

For \( k = 0, 1, \ldots, N-1 \), let \( J_k^*(x_k) \) be the optimal cost for the \((N-k)\)-stage problem that starts at state \( x_k \) and time \( k \), and ends at time \( N \); that is

\[
J_k^*(x_k) = \min_{\pi_k} E \left[ g_N(x_N) + \sum_{i=k}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right].
\]

For \( k = N \), we define \( J_N^*(x_N) = g_N(x_N) \). We will show by backward induction that the functions \( J_k^* \) are equal to the functions \( J_k \) generated by the DP algorithm, so that for \( k = 0 \) we get the desired result.

Start by defining for any \( \epsilon > 0 \), and for all \( k \) and \( x_k \), an admissible control \( \mu_k^*(x_k) \in U_k(x_k) \) for the DP recursion (2.4.1) such that

\[
E_{w_k} \left[ g_k(x_k, \mu_k^*(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k^*(x_k), w_k)) \right] \leq J_k(x_k) + \epsilon \tag{2.4.2}
\]

Because of our former assumption, \( J_{k+1}(x_k) \) generated by the DP algorithm is well defined and finite for all \( k \) and \( x_k \in S_k \). Let \( J_k^*(x_k) \) be the expected cost when using the policy \( \{\mu_k^*, \ldots, \mu_{N-1}^*\} \).

We will show by induction that for all \( x_k \) and \( k \), it must hold that

\[
J_k(x_k) \leq J_k^*(x_k) \leq J_k^*_k(x_k) \leq J_k^*_k(x_k) + (N-k)\epsilon, \tag{2.4.3}
\]

\[
J_k^*_k(x_k) \leq J_k^*_k(x_k) \leq J_k^*_k(x_k) + (N-k)\epsilon, \tag{2.4.4}
\]

\[
J_k(x_k) = J_k^*_k(x_k) \tag{2.4.5}
\]
• For $k = N - 1$, we have

$$J_{N-1}(x_{N-1}) = \min_{u_{N-1} \in U_{N-1}(x_{N-1})} E_{w_{N-1}} \left[ g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) + g_N(x_N) \right]$$

$$J_{N-1}^*(x_{N-1}) = \min_{\pi_{N-1}} E_{w_{N-1}} \left[ g_{N-1}(x_{N-1}, \mu_{N-1}(x_{N-1}), w_{N-1}) + g_N(x_N) \right]$$

Both minimizations guarantee the LHS inequalities in (2.4.3) and (2.4.4) when comparing versus

$$J_{N-1}^*(x_{N-1}) = E_{w_{N-1}} \left[ g_{N-1}(x_{N-1}, \mu_{N-1}^*(x_{N-1}), w_{N-1}) + g_N(x_N) \right],$$

with $\mu_{N-1}^*(x_{N-1}) \in U_{N-1}(x_{N-1})$. The RHS inequalities there hold just by the construction in (2.4.2). By taking $\epsilon \to 0$ in equations (2.4.3) and (2.4.4), it is also seen that $J_{N-1}(x_{N-1}) = J_{N-1}^*(x_{N-1})$.

• Suppose that equations (2.4.3)-(2.4.5) hold for period $k + 1$. For period $k$, we have:

$$J_k^*(x_k) = \min \left[ g_k(x_k, \mu_k^*(x_k), w_k) + J_{k+1}^*(f_k(x_k, \mu_k^*(x_k), w_k)) \right] \quad \text{(by definition of } J_k^*(x_k))$$

$$J_k(x_k) \leq \min_{u_k \in U_k(x_k)} \left[ g_k(x_k, \mu_k^*(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k^*(x_k), w_k)) \right] \quad \text{(by IH)}$$

$$J_k(x_k) \leq J_k(x_k) + \epsilon + (N - k - 1)\epsilon \quad \text{(by equation (2.4.2))}$$

$$J_k(x_k) = J_k(x_k) + (N - k)\epsilon$$

We also have

$$J_k(x_k) = \min \left[ g_k(x_k, \mu_k^*(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k^*(x_k), w_k)) \right] \quad \text{(by min over all admissible controls)}$$

Combining the preceding two relations, we see that equation (2.4.3) holds.

In addition, for every policy $\pi = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$, we have

$$J_k^*(x_k) = \min \left[ g_k(x_k, \mu_k^*(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k^*(x_k), w_k)) \right] \quad \text{(by definition of } J_k^*(x_k))$$

$$J_k(x_k) \leq \min_{u_k \in U_k(x_k)} \left[ g_k(x_k, \mu_k(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k(x_k), w_k)) \right] \quad \text{(by IH)}$$

$$J_k(x_k) \leq J_k(x_k) + \epsilon + (N - k - 1)\epsilon \quad \text{(by equation (2.4.2))}$$

$$J_k(x_k) \leq J_k(x_k) + (N - k)\epsilon \quad \text{(since } \mu_k(x_k) \text{ is an admissible control for period } k)$$

(Note that by IH, $J_{k+1}$ is the optimal cost starting from period $k + 1$)

$$J_k(x_k) \leq J_k(x_k) + (N - k)\epsilon \quad \text{(where } \pi^{k+1} \text{ is an admissible policy starting from period } k + 1)$$

$$J_k(x_k) = J_{\pi^k}(x_k) + (N - k)\epsilon, \quad \text{for } \pi^k = (\mu_k, \pi^{k+1})$$

Since $\pi^k$ is any admissible policy, taking the minimum over $\pi^k$ in the preceding relation, we obtain for all $x_k$,

$$J_k^*(x_k) \leq J_k^*(x_k) + (N - k)\epsilon.$$

We also have by the definition of the optimal cost $J_k^*$, for all $x_k$,

$$J_k^*(x_k) \leq J_k^*(x_k).$$
Combining the preceding two relations, we see that equation (2.4.4) holds for period $k$. Finally, equation (2.4.5) follows from equations (2.4.3) and (2.4.4) by taking $\epsilon \to 0$, and the induction is complete.

**Example 2.4.1 (Linear-quadratic example)**

A certain material is passed through a sequence of two ovens (see Figure 2.4.4). Let

- $x_0$: Initial temperature of the material,
- $x_k, k = 1, 2$: Temperature of the material at the exit of Oven $k$,
- $u_0$: Prevailing temperature in Oven 1,
- $u_1$: Prevailing temperature in Oven 2,

![Figure 2.4.4: System dynamics of Example 5.](image)

Consider a system equation

$$x_{k+1} = (1 - a)x_k + au_k, \quad k = 0, 1,$$

where $a \in (0, 1)$ is a given scalar. Note that the system equation is linear in the control and the state. The objective is to get a final temperature $x_2$ close to a given target $T$, while expending relatively little energy. This is expressed by a total cost function of the form

$$r(x_2 - T)^2 + u_0^2 + u_1^2,$$

where $r$ is a scalar. In this way, we are penalizing quadratically a deviation from the target $T$. Note that the cost is quadratic in both controls and states.

We can cast this problem into a DP framework by setting $N = 2$, and a terminal cost $g_2(x_2) = r(x_2 - T)^2$, so that the border condition for the algorithm is

$$J_2(x_2) = r(x_2 - T)^2.$$

Proceeding backwards, we have

$$J_1(x_1) = \min_{u_1 \geq 0} \{u_1^2 + J_2(x_2)\}$$

$$= \min_{u_1 \geq 0} \{u_1^2 + J_2((1 - a)x_1 + au_1)\}$$

$$= \min_{u_1 \geq 0} \{u_1^2 + r((1 - a)x_1 + au_1 - T)^2\} \quad (2.4.6)$$
This is a quadratic function in $u_1$ that we can solve by setting to zero the derivative with respect to $u_1$. We will get an expression $u_1 = \mu^*_1(x_1)$ depending linearly on $x_1$. By substituting back this expression for $u_1$ into (2.4.6), we obtain a closed form expression for $J_1(x_1)$, which is quadratic in $x_1$.

Proceeding backwards further,

$$J_0(x_0) = \min_{u_0 \geq 0} \{ u_0^2 + J_1((1-a)x_0 + au_0) \}$$

Since $J_1(\cdot)$ is quadratic in its argument, then it is quadratic in $u_0$. We minimize with respect to $u_0$ by setting the correspondent derivative to zero, which will depend on $x_0$. The optimal temperature of the first oven will be a function $\mu^*_0(x_0)$. The optimal cost is obtained by substituting this expression in the formula for $J_0$. \Box

### 2.4.1 Exercises

**Exercise 2.4.1** Consider the system

$$x_{k+1} = x_k + u_k + w_k, \; k = 0, 1, 2, 3,$$

with initial state $x_0 = 5$, and cost function

$$\sum_{k=0}^{3} (x_k^2 + u_k^2)$$

Apply the DP algorithm for the following three cases:

(a) The control constraint set is $U_k(x_k) = \{ u | 0 \leq x_k + u \leq 5, \; u \; \text{integer} \}$, for all $x_k$ and $k$, and the disturbance verifies $w_k = 0$ for all $k$.

(b) The control constraint and the disturbance $w_k$ are as in part (a), but there is in addition a constraint $x_4 = 5$ on the final state.

*Hint:* For this problem you need to define a state space for $x_4$ that consists of just the value $x_4 = 5$, and to redefine $U_3(x_3)$. Alternatively, you may use a terminal cost $g_4(x_4)$ equal to a very large number for $x_4 \neq 5$.

(c) The control constraint is as in part (a) and the disturbance $w_k$ takes the values $-1$ and $1$ with probability $1/2$ each, for all $x_k$ and $w_k$, except if $x_k + w_k$ is equal to 0 or 5, in which case $w_k = 0$ with probability 1.

**Note:** In this exercise (and in the exercises below), when the output of the DP algorithm is requested, submit the tables describing state $x_k$, optimal cost $J_k(x_k)$, and optimal control $\mu_k(x_k)$, for periods $k = 0, 1, \ldots, N - 1$ (e.g., in this case, $N = 4$).

**Exercise 2.4.2** Suppose that we have a machine that is either running or is broken down. If it runs throughout one week, it makes a gross profit of $100$. If it fails during the week, gross profit is zero. If it is running at the start of the week and we perform preventive maintenance, the probability that it will fail during the week is 0.4. If we do not perform such maintenance, the probability of failure is 0.7. However, maintenance will cost $20.
When the machine is broken down at the start of the week, it may either be repaired at a cost of $40, in which case it will fail during the week with a probability of 0.4, or it maybe replaced at a cost of $150 by a new machine that is guaranteed to run its first week of operation.

Find the optimal repair, replacement, and maintenance policy that maximizes total profit over four weeks, assuming a new machine at the start of the first week.

**Exercise 2.4.3** In the framework of the basic problem, consider the case where the cost is of the form

\[
E_{w_k} = \alpha^k g_k(x_k, u_k, w_k) + \sum_{k=0}^{N-1} \alpha^{N-k} g_N(x_N) + \sum_{k=0}^{N-1} \alpha^{N-k} g_k(x_k, u_k, w_k),
\]

where \(\alpha\) is a discount factor with \(0 < \alpha < 1\). Show that an alternate form of the DP algorithm is given by

\[
V_N(x_N) = g_N(x_N),
V_k(x_k) = \min_{u_k \in U_k(x_k)} \left\{ g_k(x_k, u_k, w_k) + \alpha V_{k+1}(f_k(x_k, u_k, w_k)) \right\}
\]

**Exercise 2.4.4** In the framework of the basic problem, consider the case where the system evolution terminates at time \(i\) when a given value \(\bar{w}\) of the disturbance at time \(i\) occurs, or when a termination decision \(\bar{u}_i\) is made by the controller. If termination occurs at time \(i\), the resulting cost is

\[
T + \sum_{k=0}^{i} g_k(x_k, u_k, w_k),
\]

where \(T\) is a termination cost. If the process has not terminated up to the final time \(N\), the resulting cost is \(g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k)\). Reformulate the problem into the framework of the basic problem.

**Hint:** Augment the state space with a special termination state.

**Exercise 2.4.5** For the Stock Option problem discussed in class, where the time index runs backwards (i.e., period 0 is the terminal stage), prove the following statements:

1. The optimal cost function \(J_n(s)\) is increasing and continuous in \(s\).
2. The optimal cost function \(J_n(s)\) is increasing in \(n\).
3. If \(\mu_F \geq 0\) and we do not exercise the option if expected profit is zero, then the option is never exercised before maturity.

**Exercise 2.4.6** Consider a device consisting of \(N\) stages connected in series, where each stage consists of a particular component. The components are subject to failure, and to increase the reliability of the device duplicate components are provided. For \(j = 1, 2, \ldots, N\), let \((1 + m_j)\) be the number of components for the \(j\)th stage (one mandatory component, and \(m_j\) backup ones), let \(p_j(m_j)\) be the probability of successful operation when \((1 + m_j)\) components are used, and let \(c_j\) denote the cost of a single backup component at the \(j\)th stage. Formulate in terms of DP the problem of finding the number of components at each stage that maximizes the reliability of the device expressed by the product

\[
p_1(m_1) \cdot p_2(m_2) \cdots p_N(m_N),
\]

subject to the cost constraint \(\sum_{j=1}^{N} c_j m_j \leq A\), where \(A > 0\) is given.
Exercise 2.4.7 (Monotonicity Property of DP) An evident, yet very important property of the DP algorithm is that if the terminal cost \( g_N \) is changed to a uniformly larger cost \( \bar{g}_N \) (i.e., \( g_N(x_N) \leq \bar{g}_N(x_N), \forall x_N \)), then clearly the last stage cost-to-go \( J_{N-1}(x_{N-1}) \) will be uniformly increased (i.e., \( J_{N-1}(x_{N-1}) \leq \bar{J}_{N-1}(x_{N-1}) \)).

More generally, given two functions \( J_{k+1} \) and \( \bar{J}_{k+1} \), with \( J_{k+1}(x_{k+1}) \leq \bar{J}_{k+1}(x_{k+1}) \) for all \( x_{k+1} \), we have, for all \( x_k \) and \( u_k \in U_k(x_k) \),

\[
E_{w_k} \left[ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right] \leq E_{w_k} \left[ g_k(x_k, u_k, w_k) + \bar{J}_{k+1}(f_k(x_k, u_k, w_k)) \right].
\]

Suppose now that in the basic problem the system and cost are time invariant; that is, \( S_k \triangleq S, C_k \triangleq C, D_k \triangleq D, f_k \triangleq f, U_k \triangleq U, P_k \triangleq P \), and \( g_k \triangleq g \). Show that if in the DP algorithm we have \( J_{N-1}(x) \leq J_N(x) \) for all \( x \in S \), then

\[
J_k(x) \leq J_{k+1}(x), \quad \text{for all } x \in S \text{ and } k.
\]

Similarly, if we have \( J_{N-1}(x) \geq J_N(x) \) for all \( x \in S \), then

\[
J_k(x) \geq J_{k+1}(x), \quad \text{for all } x \in S \text{ and } k.
\]

### 2.5 Linear-Quadratic Regulator

#### 2.5.1 Preliminaries: Review of linear algebra and quadratic forms

We will be using some results of linear algebra. Here is a summary of them:

1. Given a matrix \( A \), we let \( A' \) be its transpose. It holds that \( (AB)' = B'A' \), and \( (A^n)' = (A')^n \).

2. The rank of a matrix \( A \in \mathbb{R}^{m \times n} \) is equal to the maximum number of linearly independent row (column) vectors. The matrix is said to be full rank if rank\((A) = \min\{m, n\} \). A square matrix is of full rank if and only if it is nonsingular.

3. rank\((A) = \text{rank}(A') \).

4. Given a matrix \( A \in \mathbb{R}^{n \times n} \), the determinant of the matrix \( \gamma I - A \), where \( I \) is the \( n \times n \) identity matrix and \( \gamma \) is a scalar, is an \( n \)th degree polynomial. The \( n \) roots of this polynomial are called the eigenvalues of \( A \). Thus, \( \gamma \) is an eigenvalue of \( A \) if and only if the matrix \( \gamma I - A \) is singular (i.e., it does not have an inverse), or equivalently, if there exists a vector \( v \neq 0 \) such that \( Av = \gamma v \). Such vector \( v \) is called an eigenvector corresponding to \( \gamma \).

The eigenvalues and eigenvectors of \( A \) can be complex even if \( A \) is real.

A matrix \( A \) is singular if and only if it has an eigenvalue that is equal to zero.

If \( A \) is nonsingular, then the eigenvalues of \( A^{-1} \) are the reciprocals of the eigenvalues of \( A \).

The eigenvalues of \( A \) and \( A' \) coincide.

5. A square symmetric \( n \times n \) matrix \( A \) is said to be positive semidefinite if \( x'Ax \geq 0, \forall x \in \mathbb{R}^n, x \neq 0 \). It is said to be positive definite if \( x'Ax > 0, \forall x \in \mathbb{R}^n, x \neq 0 \). We will denote \( A \geq 0 \) and \( A > 0 \) to denote positive semidefiniteness and definiteness, respectively.
6. If $A$ is an $n \times n$ positive semidef. symmetric matrix and $C$ is an $m \times n$ matrix, then the matrix $CAC'$ is positive semidef. symmetric. If $A$ is positive def. symmetric, and $C$ has rank $m$ (equivalently, $m \leq n$ and $C$ has full rank), then $CA'C$ is positive def. symmetric.

7. An $n \times n$ positive def. matrix $A$ can be written as $CC'$ where $C$ is a square invertible matrix.

8. The expected value of a random vector $x = x_1, \ldots, x_n$ is the vector:

$$E[x] = (E[x_1], \ldots, E[x_n]).$$

The covariance matrix of a random vector $x$ with expected value $E[\bar{x}]$ is defined to be the $n \times n$ positive semidefinite matrix

$$
\begin{pmatrix}
E[(x_1 - \bar{x}_1)^2] & \cdots & E[(x_1 - \bar{x}_1)(x_n - \bar{x}_n)] \\
\vdots & \ddots & \vdots \\
E[(x_n - \bar{x}_n)(x_1 - \bar{x}_1)] & \cdots & E[(x_n - \bar{x}_n)^2]
\end{pmatrix}
$$

9. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a quadratic form

$$f(x) = \frac{1}{2} x'Qx + b'x,$$

where $Q$ is a symmetric $n \times n$ matrix and $b \in \mathbb{R}^n$. Its gradient is given by

$$\nabla f(x) = Qx + b.$$

The function $f$ is convex if and only if $Q$ is positive semidefinite. If $Q$ is positive definite, then $f$ is convex and $Q$ is invertible, so a vector $x^*$ minimizes $f$ if and only if

$$\nabla f(x^*) = Qx^* + b = 0,$$

or equivalently, $x^* = -Q^{-1}b$.

### 2.5.2 Problem setup

**System equation:** $x_{k+1} = A_k x_k + B_k u_k + w_k$ [Linear in both state and control.]

**Quadratic cost:**

$$E_{w_0, \ldots, w_{N-1}} \left\{ x_N'Q_Nx_N + \sum_{k=0}^{N-1} x_k'Q_kx_k + u_k'R_ku_k \right\},$$

where:

- $Q_k \geq 0$ are square, symmetric, positive semidefinite matrices with appropriate dimension,
- $R_k > 0$ are square, symmetric, positive definite matrices with appropriate dimension,
- Disturbances $w_k$ are independent with $E[w_k] = 0$, and do not depend on $x_k$ nor on $u_k$ (the case $E[w_k] \neq 0$ will be discussed later, in Section 2.5.7),
- Controls $u_k$ are unconstrained.
DP Algorithm:

\[ J_N(x_N) = x_N^T Q_N x_N \]  
\[ J_k(x_k) = \min_{u_k} E_{w_k} \{ x_k^T Q_k x_k + u_k^T R_k u_k + J_{k+1}(A_k x_k + B_k u_k + w_k) \} \]  

Intuition: The purpose of this DP is to bring the state closer to \( x_k = 0 \) as soon as possible. Any deviation from zero is penalized quadratically.

2.5.3 Properties

- \( J_k(x_k) \) is quadratic in \( x_k \)
- Optimal policy \( \{ \mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^* \} \) is linear, i.e. \( \mu_k^*(x_k) = L_k x_k \)
- Similar treatment to several variants of the problems as follows

**Variant 1:** Nonzero mean \( w_k \).

**Variant 2:** Shifted problem, i.e., set the target in a vector \( \bar{x}_N \) rather than in zero:

\[ E[(x_N - \bar{x}_N)^T Q_N(x_N - \bar{x}_N)] + \sum_{k=0}^{N-1} ((x_N - \bar{x}_k)^T Q_k x_k - \bar{x}_k) + u_k^T R_k u_k. \]

2.5.4 Derivation

By induction, we want to verify that \( \mu_k^*(x_k) = L_k x_k \) and \( J_k(x_k) = x_k^T K_k x_k + \text{constant} \), where \( L_k \) are gain matrices\(^5\) given by

\[ L_k = -(B_k^T K_{k+1} B_k + R_k)^{-1} B_k^T K_{k+1} A_k, \]

and where \( K_k \) are symmetric positive semidefinite matrices given by

\[ K_N = Q_N \]

\[ K_k = A_k^T (K_{k+1} - K_{k+1} B_k (B_k^T K_{k+1} B_k + R_k)^{-1} B_k^T K_{k+1}) A_k + Q_k. \] \hspace{1cm} (2.5.3)

The above equation is called the discrete time Riccati equation. Just like DP, it starts at the terminal time \( N \) and proceeds backwards.

We will show that the optimal policy (but not the optimal cost) is the same as when \( w_k \) is replaced by \( E[w_k] = 0 \) (property known as certainty equivalence).

Induction argument proceeds as follows. Write equation (4.2.5) for \( N-1 \):

\[ J_{N-1}(x_{N-1}) = \min_{u_{N-1}} E_{w_{N-1}} \{ x_{N-1}^T Q_{N-1} x_{N-1} + u_{N-1}^T R_{N-1} u_{N-1} \}
+ (A_{N-1} x_{N-1} + B_{N-1} u_{N-1} + w_{N-1})^T Q_N (A_{N-1} x_{N-1} + B_{N-1} u_{N-1} + w_{N-1}) \}
= x_{N-1}^T Q_{N-1} x_{N-1} + \min_{u_{N-1}} \{ u_{N-1}^T R_{N-1} u_{N-1} + u_{N-1}^T B_{N-1}^T Q_N B_{N-1} u_{N-1} \}
+ 2 x_{N-1} A_{N-1}^T Q_N B_{N-1} u_{N-1} + x_{N-1}^T A_{N-1}^T Q_N A_{N-1} x_{N-1} + E[w_{N-1}^T Q_N w_{N-1}] \} \] \hspace{1cm} (2.5.4)

\(^5\)The idea is that \( L_k \) represent how much we gain in our path towards the target zero.
where in the expansion of the last term in the second line we are using the fact that since $E[w] = 0$, then $E[w_{N-1}Q_N(A_{N-1}x_{N-1} + B_{N-1}u_{N-1})] = 0$.

By differentiating the equation w.r.t $u_{N-1}$, and setting the derivative equal to 0; we get

\[
( R_{N-1} + B'_{N-1}Q_NB_{N-1} )u_{N-1} = -B'_{N-1}Q_NA_{N-1}x_{N-1},
\]

leading to

\[
u^*_N = -(R_{N-1} + B'_{N-1}Q_NB_{N-1})^{-1}B'_{N-1}Q_NA_{N-1}x_{N-1}.
\]

By substitution into (4.2.6), we get

\[
J_{N-1}(x_{N-1}) = x'_{N-1}K_{N-1}x_{N-1} + E[w'_{N-1}Q_Nw_{N-1}]
\]

where $K_{N-1} = A'_{N-1}(Q_N - Q_NB_{N-1}(B'_{N-1}Q_NB_{N-1} + R_{N-1})^{-1}B'_{N-1}Q_N)A_{N-1} + Q_N - 1$.

Facts:

- The matrix $K_{N-1}$ is symmetric, since $K_{N-1} = K'_{N-1}$.
- **Claim:** $K_{N-1} \geq 0$ (we need this result to prove that the next matrix $L_{N-2}$ is invertible).

**Proof:** From (4.2.7) we have

\[
x'_{N-1}K_{N-1}x_{N-1} = J_{N-1}(x_{N-1}) - E[w'_{N-1}Q_Nw_{N-1}].
\]

So,

\[
x'K_{N-1}x = x'Q_{N-1}x + \min_u \left\{ u'R_{N-1}u + (A_{N-1}x + B_{N-1}u)'Q_N(A_{N-1}x + B_{N-1}u) \right\}_{\geq 0}.
\]

Note that the $E[w'_{N-1}Q_Nw_{N-1}]$ in $J_{N-1}(x_{N-1})$ cancels out with the one in (2.5.6). Thus, the expression in brackets is nonnegative for every $u$. The minimization over $u$ preserves nonnegativity, showing that $K_{N-1}$ is also positive semidefinite.

In conclusion,

\[
J_{N-1}(x_{N-1}) = x'_{N-1}K_{N-1}x_{N-1} + \text{constant}
\]

is a positive semidefinite quadratic function (plus an inconsequential constant term), we may proceed backward and obtain from DP equation (4.2.5) the optimal law for stage $N - 2$. As earlier, we show that $J_{N-2}$ is positive semidef. and by proceeding sequentially we obtain

\[
u^*_K(x_k) = L_kx_k,
\]

where

\[
L_k = -(B'_{k}K_{k+1}B_{k} + R_{k})^{-1}B'_{k}K_{k+1}A_{k},
\]

and where the symmetric $\geq 0$ matrices $K_k$ are given recursively by

\[K_N = Q_N,
\]

74
\[ K_k = \left( A_k' (K_{k+1} - K_{k+1} K_{k+1} B_k + R_k)^{-1} B_k' K_{k+1} A_k + Q_k \right). \]

Just like DP, this algorithm starts at the terminal time \( N \) and proceeds backwards. The optimal cost is given by

\[
J_0(x_0) = x_0' K_0 x_0 + \sum_{k=0}^{N-1} \mathbb{E} \left[ w_k' K_{k+1} w_k \right].
\]

The control \( u_k^* \) and the system equation lead to the linear feedback structure represented in Figure 2.5.1.

2.5.5 Asymptotic behavior of the Riccati equation

The objective of this section is to prove the following result: If matrices \( A_k, B_k, Q_k \) and \( R_k \) are constant and equal to \( A, B, Q, R \) respectively, then \( K_k \rightarrow K \) as \( k \rightarrow -\infty \) (i.e., when we have many periods ahead), where \( K \) satisfies the algebraic Riccati equation:

\[
K = A' (K - KB(B'KB + R)^{-1} B'K) A + Q,
\]

where \( K \geq 0 \) and is unique (within the class of positive semidefinite matrices) solution. This property indicates that for the system

\[
x_{k+1} = Ax_k + Bu_k + w_k, \quad k = 0, 1, 2, \ldots, N - 1,
\]

and a large \( N \), one can approximate the control \( \mu^*_k(x_k) = L_k x_k \) by the steady state control:

\[
\mu^*_k(x) = L x,
\]

where

\[
L = -(B'KB + R)^{-1} B'KA.
\]

Before proving the above result, we need to introduce three notions: controllability, observability, and stability.

**Definition 2.5.1** A pair of matrices \( (A, B) \), where \( A \in \mathbb{R}^{n \times n} \) and \( B \in \mathbb{R}^{n \times m} \), is said to be controllable if the \( n \times (n, m) \) matrix: \( [B, AB, A^2 B, \ldots, A^{n-1} B] \) has full rank.

**Definition 2.5.2** A pair \( (A, C) \), \( A \in \mathbb{R}^{n \times n}, C \in \mathbb{R}^{m \times n} \) is said to be observable if the pair \( (A', C') \) is controllable.
The next two claims provide intuition for the previous definitions:

**Claim:** If the pair \((A, B)\) is controllable, then for any initial state \(x_0\) there exists a sequence of control vectors \(u_0, u_1, \ldots, u_{N-1}\), that forces state \(x_n\) of the system: \(x_{k+1} = Ax_k + Bu_k\) to be equal to zero at time \(n\).

**Proof:** By successively applying the equation \(x_{k+1} = Ax_k + Bu_k\), for \(k = n - 1, n - 2, \ldots, 0\), we obtain

\[
x_n = A^n x_0 + Bu_{n-1} + ABu_{n-2} + \cdots + A^{n-1}Bu_0,
\]

or equivalently

\[
x_n - A^n x_0 = (B, AB, \ldots, A^{n-1}B)(u_{n-1}, u_{n-2}, \ldots, u_1, u_0)'
\]

Since \((A, B)\) is controllable, \((B, AB, \ldots, A^{n-1}B)\) has full rank and spans the whole space \(\mathbb{R}^n\). Hence, we can find \((u_{n-1}, u_{n-2}, \ldots, u_1, u_0)\) \(\in \mathbb{R}^n\) such that

\[
(B, AB, \ldots, A^{n-1}B)(u_{n-1}, u_{n-2}, \ldots, u_1, u_0)' = v,
\]

for any vector \(v \in \mathbb{R}^n\). In particular, by setting \(v = -A^n x_0\), we obtain \(x_n = 0\) in equation (3.3.4).

In words: The system equation \(x_{k+1} = Ax_k + Bu_k\) under controllable matrices \((A, B)\) in the space \(\mathbb{R}^n\) warrants convergence to the zero vector in exactly \(n\) steps.

**Claim:** Suppose that \((A, C)\) is observable (i.e., \((A', C')\) is controllable). In the context of estimation problems, given measurements \(z_0, z_1, \ldots, z_{n-1}\) of the form

\[
\begin{array}{ccl}
z_k & \in & \mathbb{R}^{m \times 1} \\
\frac{C_k}{C} & \in & \mathbb{R}^{m \times n} \\
\frac{x_k}{x} & \in & \mathbb{R}^{n \times 1}
\end{array}
\]

it is possible to uniquely infer the initial state \(x_0\) of the system \(x_{k+1} = Ax_k\).

**Proof:** In view of the relation

\[
\begin{aligned}
z_0 &= Cx_0 \\
x_1 &= Ax_0 \\
z_1 &= Cx_1 = CAx_0 \\
x_2 &= Ax_1 = A^2x_0 \\
z_2 &= Cx_1 = CA^2x_0 \\
\vdots & \quad \vdots \\
z_{n-1} &= Cx_{n-1} = CA^{n-1}x_0,
\end{aligned}
\]

or in matrix form, in view of

\[
(z_0, z_1, \ldots, z_{n-1})' = (C, CA, \ldots, CA^{n-1})'x_0,
\]

where \((C, CA, \ldots, CA^{n-1})\) has full rank \(n\), there is a unique \(x_0\) that satisfies (2.5.8).

To get the previous result we are using the following: If \((A, C)\) is observable, then \((A', C')\) is controllable. So, if we denote

\[
\alpha \overset{\Delta}{=} (C', A'C', (A')^2C', \ldots, (A')^{n-1}C') = (C', (CA)',$}(CA^2)'$,\ldots,(CA^{n-1})'),
\]
then $\alpha$ is full rank, and therefore $\alpha'$ has full rank, where

$$
\alpha' = \begin{pmatrix} C \\
CA \\
CA^2 \\
\vdots \\
CA^{n-1} \end{pmatrix},
$$

which completes the argument.

In words: The system equation $x_{k+1} = Ax_k$ under observable matrices $(A, C)$ allow to infer the initial state of a sequence of observations $z_0, z_1, \ldots, z_{n-1}$ given by $z_k = Cx_k$.

**Stability:** The concept of stability refers to the fact that in the absence of random disturbance, the dynamics of the system driven by the control $\mu(x) = Lx$, bring the state $x_{k+1} = Ax_k + Bu_k = (A + BL)x_k$, $k = 0, 1, \ldots,$ towards zero as $k \to \infty$. For any $x_0$, since $x_k = (A + BL)^k x_0$, it follows that the closed-loop system is stable if and only if $(A + BL)^k \to 0$, or equivalently, if and only if the eigenvalues of the matrix $(A + BL)$ are strictly within the unit circle of the complex plane.

Assume time-independent system and cost per stage, and some technical assumptions: controllability of $(A, B)$ and observability of $(A, C)$ where $Q = C'C$. The Riccati equation (2.5.3) converges $\lim_{k \to -\infty} K_k = K$, where $K$ is positive definite, and is the unique (within the class of positive semidef. matrices) solution of the algebraic Riccati equation

$$
K = A'(K - KB(B'KB + R)^{-1}B'K)A + Q.
$$

The following proposition formalizes this result. To simplify notation, we reverse the time indexing of the Riccati equation. Thus, $P_k$ corresponds to $K_{N-k}$ in (2.5.3).

**Proposition 2.5.1** Let $A$ be an $n \times n$ matrix, $B$ be an $n \times m$ matrix, $Q$ be an $n \times n$ positive semidef. symmetric matrix, and $R$ be an $m \times m$ positive definite symmetric matrix. Consider the discrete-time Riccati equation

$$
P_{k+1} = A'(P_k - P_k B (B'P_k B + R)^{-1}B'P_k)A + Q, \quad k = 0, 1, \ldots, \tag{2.5.9}
$$

where the initial matrix $P_0$ is an arbitrary positive semidef. symmetric matrix. Assume that the pair $(A, B)$ is controllable. Assume also that $Q$ may be written as $Q = C'C$, where the pair $(A, C)$ is observable. Then,

(a) There exists a positive def. symmetric matrix $P$ such that for every positive semidef. symmetric initial matrix $P_0$ we have $\lim_{k \to -\infty} P_k = P$. Furthermore, $P$ is the unique solution of the algebraic matrix equation

$$
P = A'(P - PB(B'PB + R)^{-1}B'P)A + Q
$$

within the class of positive semidef. symmetric matrices.
(b) The corresponding closed-loop system is stable; that is, the eigenvalues of the matrix

\[ D = A + BL, \]

where

\[ L = -(B'PB + R)^{-1}B'PA, \]

are strictly within the unit circle of the complex plane.

Observations:

- The implication of the observability assumption in the proposition is that in the absence of control, if the state cost per stage \( x'_kQx_k \to 0 \) as \( k \to \infty \), or equivalently \(Cx_k \to 0\), then also \( x_k \to 0\).

- We could replace the statement in Proposition 2.5.1, part (b), by

\[(A + BL)^k \to 0 \quad \text{as} \quad k \to \infty. \]

Since \( x_{k+1} = (A + BL)^k x_0 \), then \( x_k \to 0 \) as \( k \to \infty \).

Graphical proof of Proposition 2.5.1 for the scalar case

We provide here a proof for a limited version of the statement in the proposition, where we assume a one-dimensional state and control. For \( A \neq 0, B \neq 0, Q > 0 \), and \( R > 0 \), the Riccati equation in (2.5.9) is given by

\[ P_{k+1} = A^2 \left( P_k - \frac{B^2P_k^2}{B^2P_k + R} \right) + Q, \]

which can be equivalently written as

\[ P_{k+1} = F(P_k), \quad \text{where} \quad F(P) = \frac{A^2RP}{B^2P + R} + Q. \quad (2.5.10) \]

Figure 2.5.2 illustrates this recursion.

Facts about Figure 2.5.2:

- \( F \) is concave and monotonic increasing in the range \((-R/B^2, \infty)\).

- The equation \( P = F(P) \) has one solution \( P^* > 0 \) and one solution \( \tilde{P} < 0 \).

- The Riccati iteration \( P_{k+1} = F(P_k) \) converges to \( P^* > 0 \) starting anywhere in \((\tilde{P}, \infty)\).

Technical note: Going back to the matrix case: If controllability of \((A, B)\) and observability of \((A, C)\) are replaced by two weaker assumptions:

- Stabilizability, i.e., there exists a feedback gain matrix \( G \in \mathbb{R}^{m \times n} \) such that the closed-loop system \( x_{k+1} = (A + BG)x_k \) is stable.

- Detectability, i.e., \( A \) is such that if \( u_k \to 0 \) and \( Cx_k \to 0 \), then it follows that \( x_k \to 0 \), and that \( x_{k+1} = (A + BL)x_k \) is stable.

Then, the conclusions of the proposition hold with the exception of positive def. of the limit matrix \( P \), which can now only be guaranteed to be positive semidef.
2.5.6 Random system matrices

Setting:

- Suppose that \( \{A_0, B_0\}, \ldots, \{A_{N-1}, B_{N-1}\} \) are not known but rather are independent random matrices that are also independent of \( w_0, \ldots, w_{N-1} \).
- Assume that their probability distribution are given, and have finite variance.
- To cast this problem into the basic DP framework, define disturbances \( (A_k, B_k, w_k) \).

The DP algorithm is:

\[
J_N(x_N) = x_N'Q_Nx_N
\]

\[
J_k(x_k) = \min_{u_k} E_{A_k, B_k, w_k} \left[ x_k'Q_kx_k + u_k'R_ku_k + J_{k+1}(A_kx_k + B_ku_k + w_k) \right]
\]

In this case, similar calculations to those for the deterministic matrices give:

\[
\mu_k^*(x_k) = L_kx_k,
\]

where the gain matrices are given by

\[
L_k = -(R_k + E[B_k'K_{k+1}B_k])^{-1}E[B_k'K_{k+1}A_k],
\]

and where the matrices \( K_k \) are given by the generalized Riccati equation

\[ K_N = Q_N, \]

\[ K_k = E[A_k'K_{k+1}A_k] - E[A_k'K_{k+1}B_k](R_k + E[B_k'K_{k+1}B_k])^{-1}E[B_k'K_{k+1}A_k] + Q_k. \quad (2.5.11) \]

In the case of a stationary system and constant matrices \( Q_k \) and \( R_k \), it is not necessarily true that the above equation converges to a steady-state solution. This is illustrated in Figure 2.5.3. In the
case of a scalar stationary system (one-dimensional state and control), using $P_k$ in place of $K_{N-k}$, this equation is written as

$$P_{k+1} = \tilde{F}(P_k),$$

where the function $\tilde{F}$ is given by

$$\tilde{F}(P) = \frac{E[A^2]R}{E[B^2]P} + Q + \frac{TP^2}{E[B^2]P + R},$$

and where


If $T = 0$, as in the case where $A$ and $B$ are not random, the Riccati equation becomes identical with the one of Figure 2.5.2 and converges to a steady-state. Convergence also occurs when $T$ has a small positive value. However, as illustrated in the figure, for $T$ large enough, the graph of the function $\tilde{F}$ and the 45-degree line that passes through the origin do not intersect at a positive value of $P$, and the Riccati equation diverges to infinity.

**Interpretation:** $T$ is a measure of the uncertainty in the system. If there is a lot of uncertainty, optimization over a long horizon is meaningless. This phenomenon has been called the *uncertainty threshold principle*.

### 2.5.7 On certainty equivalence

Consider the optimization problem:

$$\min_u E_w[(ax + bu + w)^2],$$

where $a, b$ are scalars, $x$ is known, and $w$ is random. We have

$$E_w[(ax + bu + w)^2] = E[(ax + bu)^2 + w^2 + 2(ax + bu)w]$$

$$= (ax + bu)^2 + 2(ax + bu)E[w] + E[w^2]$$
Taking derivative with respect to \( u \) gives
\[
2(ax + bu)b + 2bE[w] = 0,
\]
and hence the minimizer is
\[
u^* = -\frac{a}{b}x - \frac{1}{b}E[w].
\]
Observe that \( u^* \) depends on \( w \) only through the mean \( E[w] \). In particular, the result of the optimization problem is the same as for the corresponding deterministic problem where \( w \) is replaced by \( E[w] \). This property is called the certainty equivalence principle.

In particular,

- For example, when \( A_k \) and \( B_k \) are known, the certainty equivalence holds (the optimal control is still linear in the state \( x_k \)).
- When \( A_k \) and \( B_k \) are random, certainty equivalence does not hold.

### 2.5.8 Exercises

**Exercise 2.5.1** Consider a linear-quadratic problem where \( A_k, B_k \) are known, for the case where at the beginning of period \( k \) we have a forecast \( y_k \in \{1, 2, \ldots, n\} \) consisting of an accurate prediction that \( w_k \) will be selected in accordance with a particular probability distribution \( P_{k|y_k} \). The vectors \( w_k \) need not have zero mean under the distribution \( P_{k|y_k} \). Show that the optimal control law is of the form
\[
u_k(x_k, y_k) = -(B'_kK_{k+1}B_k + R_k)^{-1}B'_kK_{k+1}(Ax_k + E[w_k|y_k]) + \alpha_k,
\]
where the matrices \( K_k \) are given by the discrete time Riccati equation, and \( \alpha_k \) are appropriate vectors.

**Hint:**
- System equation: \( x_{k+1} = A_kx_k + B_ku_k + w_k, \quad k = 0, 1, \ldots, N - 1, \)
- Cost = \( E_{w_0, \ldots, w_{N-1}} \left[ x_N'Q_Nx_N + \sum_{k=0}^{N-1} (x_k'Q_kx_k + u_k'R_ku_k) \right] \)

Let
\[
y_k = \text{Forecast available at the beginning of period } k
\]
\[
P_{k|y_k} = \text{p.d.f. of } w_k \text{ given } y_k
\]
\[
p_{y_k} = \text{a priori p.d.f. of } y_k \text{ at stage } k
\]

We have the following DP algorithm:
\[
J_N(x_N, y_N) = x_N'Q_Nx_N
\]
\[
J_k(x_k, y_k) = \min_{u_k} E_{w_k} \left[ x_k'Q_kx_k + u_k'R_ku_k + \sum_{i=1}^{n} p_{y_k}^k + 1 J_{k+1}(x_{k+1}; i)|y_k \right],
\]
where the noise \( w_k \) is explained by \( P_{k|y_k} \).
Prove the following result by induction. The control \( u^*_k(x_k, y_k) \) should be derived on the way.

**Proposition:** Under the conditions of the problem:

\[
J_k(x_k, y_k) = x_k' K_k x_k + x_k' b_k(y_k) + c_k(y_k), \quad k = 0, 1, \ldots, N,
\]

where \( b_k(y_k) \) is an \( n \)-dimensional vector, \( c_k(y_k) \) is a scalar, and \( K_k \) is generated by the discrete time Riccati equation.

**Exercise 2.5.2** Consider a scalar linear system

\[
x_{k+1} = a_k x_k + b_k u_k + w_k, \quad k = 0, 1, \ldots, N - 1,
\]

where \( a_k, b_k \in \mathbb{R} \), and each \( w_k \) is a Gaussian random variable with zero mean and variance \( \sigma^2 \). We assume no control constraints and independent disturbances.

1. Show that the control law \( \{u_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\} \) that minimizes the cost function

\[
E\left[ \exp\left\{ x_N^2 + \sum_{k=0}^{N-1} (x_k^2 + \rho u_k^2) \right\} \right], \quad \rho > 0,
\]

is linear in the state variable, assuming that the optimal cost is finite for every \( x_0 \).

2. Show by example that the Gaussian assumption is essential for the result to hold.

**Hint 1:** Note from integral tables that

\[
\int_{-\infty}^{+\infty} e^{-(ax^2+bx+c)} \, dx = \sqrt{\frac{\pi}{a}} e^{(b^2-4ac)/(4a)}, \quad a > 0
\]

Let \( w \) be a normal random variable with zero mean and variance \( \sigma^2 < 1/(2\beta) \). Using this definite integral, prove that

\[
E\left[ e^{\beta(a+w)^2} \right] = \frac{1}{\sqrt{1 - 2\beta \sigma^2}} \exp\left\{ \frac{\beta a^2}{1 - 2\beta \sigma^2} \right\}
\]

Then, prove that if the DP algorithm has a finite minimizing value at each step, then

\[
J_N(x_N) = e^{x_N^2},
\]

\[
J_k(x_k) = \alpha_k e^{\beta_k x_k^2}, \quad \text{for constants } \alpha_k, \beta_k > 0, \quad k = 0, 1, \ldots, N - 1.
\]

**Hint 2:** In particular for \( w_{N-1} \), consider the discrete distribution

\[
\mathbb{P}(w_{N-1} = \xi) = \begin{cases} 
1/4, & \text{if } |\xi| = 1 \\
1/2, & \text{if } \xi = 0
\end{cases}
\]

Find a functional form for \( J_{N-1}(x_{N-1}) \), and check that \( u^*_{N-1} \neq \gamma_{N-1} x_{N-1} \), for a constant \( \gamma_{N-1} \).

### 2.6 Modular functions and monotone policies

Now we go back to the basic DP setting on problems with perfect state information. We will identify conditions on a parameter \( \theta \) (e.g., \( \theta \) could be related to the state of a system) under which the optimal action \( D^*(\theta) \) varies monotonically with it. We start with some technical definitions and relevant properties.
2.6.1 Lattices

**Definition 2.6.1** Given two points \( x = (x_1, \ldots, x_n) \) and \( y = (y_1, \ldots, y_n) \) in \( \mathbb{R}^n \), we define

- Meet of \( x \) and \( y \): \( x \land y = (\min\{x_1, y_1\}, \ldots, \min\{x_n, y_n\}) \),
- Join of \( x \) and \( y \): \( x \lor y = (\max\{x_1, y_1\}, \ldots, \max\{x_n, y_n\}) \).

Then
\[
x \land y \leq x \leq x \lor y
\]

**Definition 2.6.2** A set \( X \subset \mathbb{R}^n \) is said to be a sublattice of \( \mathbb{R}^n \) if \( \forall x, y \in X, x \land y \in X \) and \( x \lor y \in X \).

**Examples:**

- \( I = \{(x, y) \in \mathbb{R}^2 | 0 \leq x \leq 1, 0 \leq y \leq 1\} \) is a sublattice.
- \( H = \{(x, y) \in \mathbb{R}^2 | x + y = 1\} \) is not a sublattice, because for example \((1, 0)\) and \((0, 1)\) are in \( H \), but not \((0, 0)\) nor \((1, 1)\) are in \( H \).

**Definition 2.6.3** A point \( x^* \in X \) is said to be a greatest element of a sublattice \( X \) if \( x^* \geq x, \forall x \in X \). A point \( \hat{x} \in X \) is said to be a least element of a sublattice \( X \) if \( \hat{x} \leq x, \forall x \in X \).

**Theorem 2.6.1** Suppose \( X \neq \emptyset \), \( X \) a compact (i.e., closed and bounded) sublattice of \( \mathbb{R}^n \). Then, \( X \) has a least and a greatest element.

2.6.2 Supermodularity and increasing differences

Let \( S \subset \mathbb{R}^n \), \( \Theta \subset \mathbb{R}^l \). Suppose that both \( S \) and \( \Theta \) are sublattices.

**Definition 2.6.4** A function \( f : S \times \Theta \to \mathbb{R} \) is said to be supermodular in \((x, \theta)\) if for all \( z = (x, \theta) \) and \( z' = (x', \theta') \) in \( S \times \Theta \):
\[
f(z) + f(z') \leq f(z \lor z') + f(z \land z').
\]

Similarly, \( f \) is submodular if
\[
f(z) + f(z') \geq f(z \lor z') + f(z \land z').
\]

**Example:** Let \( S = \Theta = \mathbb{R}_+ \), and let \( f : S \times \Theta \to \mathbb{R} \) be given by \( f(x, \theta) = x\theta \). We will show that \( f \) is supermodular in \((x, \theta)\).

Pick any \((x, \theta)\) and \((x', \theta')\) in \( S \times \Theta \), and assume w.l.o.g. \( x \geq x' \). There are two cases to consider:

1. \( \theta \geq \theta' \Rightarrow (x, \theta) \lor (x', \theta') = (x, \theta) \), and \((x, \theta) \land (x', \theta') = (x', \theta')\). Then,
\[
\begin{align*}
\underbrace{f(x, \theta)}_{x\theta} + \underbrace{f(x', \theta')}_{x'\theta'} & \leq \underbrace{f((x, \theta) \lor (x', \theta'))}_{(x, \theta)} + \underbrace{f((x, \theta) \land (x', \theta'))}_{x'\theta'}
\end{align*}
\]
2. \( \theta < \theta' \Rightarrow (x, \theta) \lor (x', \theta') = (x, \theta') \), and \((x, \theta) \land (x', \theta') = (x', \theta)\). Then,

\[
\frac{f((x, \theta) \lor (x', \theta'))}{x'\theta} + f((x, \theta) \land (x', \theta')) = x'\theta + x\theta'
\]

and we would have

\[
\frac{f(x, \theta)}{x\theta} + \frac{f(x', \theta')}{x'\theta'} \leq \frac{f((x, \theta) \lor (x', \theta'))}{x'\theta} + \frac{f((x, \theta) \land (x', \theta'))}{x\theta},
\]

if and only if

\[
x\theta + x'\theta' \leq x'\theta + x\theta' \]

\[\iff x(\theta' - \theta) - x'(\theta' - \theta) \geq 0\]

\[\iff (x - x')(\theta' - \theta) \geq 0, \quad 0 > 0\]

which is indeed the case.

Therefore, \(f(x, \theta) = x\theta\) is supermodular in \(S \times \Theta\).

**Definition 2.6.5** For \(S, \Theta \subset \mathbb{R}\), a function \(f : S \times \Theta \to \mathbb{R}\) is said to satisfy increasing differences in \((x, \theta)\) if for all pairs \((x, \theta)\) and \((x', \theta')\) in \(S \times \Theta\), if \(x \geq x'\) and \(\theta \geq \theta'\), then

\[
f(x, \theta) - f(x', \theta) \geq f(x', \theta') - f(x', \theta').
\]

If the inequality becomes strict whenever \(x \geq x'\) and \(\theta \geq \theta'\), then \(f\) is said to satisfy strictly increasing differences.

In other words, \(f\) has increasing differences in \((x, \theta)\) if the difference

\[
f(x, \theta) - f(x', \theta), \quad \text{for } x \geq x',
\]

is increasing in \(\theta\).

**Theorem 2.6.2** Let \(S, \Theta \subset \mathbb{R}\), and suppose \(f : S \times \Theta \to \mathbb{R}\) is supermodular in \((x, \theta)\). Then

1. \(f\) is supermodular in \(x\), for each fixed \(\theta \in \Theta\) (i.e., for any fixed \(\theta \in \Theta\), and for any \(x, x' \in S\), we have \(f(x, \theta) + f(x', \theta) \leq f(x \lor x', \theta) + f(x \land x', \theta)\)).

2. \(f\) satisfies increasing differences in \((x, \theta)\).

**Proof:** For part (1), fix \(\theta\). Let \(z = (x, \theta), z' = (x', \theta)\). Since \(f\) is supermodular in \((x, \theta)\):

\[
f(x, \theta) + f(x', \theta) \leq f(x \lor x', \theta) + f(x \land x', \theta),
\]

or equivalently

\[
f(z) + f(z') \leq f(z \lor z') + f(z \land z'),
\]

and the result holds.
For part (2), pick any \( z = (x, \theta) \) and \( z' = (x', \theta') \) that satisfy \( x \geq x' \) and \( \theta \geq \theta' \). Let \( w = (x, \theta') \) and \( w' = (x', \theta) \). Then, \( w \lor w' = z \) and \( w \land w' = z' \). Since \( f \) is supermodular on \( S \times \Theta \),

\[
f(w) + f(w') \leq f(w \lor w') + f(w \land w').
\]

Rearranging terms,

\[
f(x, \theta) - f(x', \theta') \geq f(x, \theta') - f(x', \theta'),
\]

and so \( f \) also satisfies increasing differences, as claimed.

**Remark:** We will prove later on that the reverse of part (2) in the theorem also holds.

**Recall:** A function \( f : S \subset \mathbb{R}^n \to \mathbb{R}^n \) is said to be of class \( C^k \) if the derivatives \( f^{(1)}, f^{(2)}, \ldots, f^{(k)} \) exist and are continuous (the continuity is automatic for all the derivatives except the last one, \( f^{(k)} \)). Moreover, if \( f \) is \( C^k \), then the cross-partial derivatives satisfy

\[
\frac{\partial^2}{\partial z_i \partial z_j} f(z) = \frac{\partial^2}{\partial z_j \partial z_i} f(z).
\]

**Theorem 2.6.3** Let \( Z \) be an open sublattice of \( \mathbb{R}^n \). A \( C^2 \) function \( h : Z \to \mathbb{R} \) is supermodular on \( Z \) if and only if for all \( z \in Z \), we have

\[
\frac{\partial^2}{\partial z_i \partial z_j} h(z) \geq 0, \quad i, j = 1, \ldots, n, \; i \neq j.
\]

Similarly, \( h \) is submodular if and only if for all \( z \in Z \), we have

\[
\frac{\partial^2}{\partial z_i \partial z_j} h(z) \leq 0, \quad i, j = 1, \ldots, n, \; i \neq j.
\]

**Proof:** We prove here the result for supermodularity for the case \( n = 2 \).

\( \Leftarrow \) If

\[
\frac{\partial^2}{\partial z_i \partial z_j} h(z) \geq 0, \quad i, j = 1, \ldots, n, \; i \neq j,
\]

then for \( x_1 > x_2 \) and \( y_1 > y_2 \),

\[
\int_{y_2}^{y_1} \int_{x_2}^{x_1} \frac{\partial^2}{\partial x \partial y} h(x, y) \, dx \, dy \geq 0
\]

So,

\[
\int_{y_2}^{y_1} \frac{\partial}{\partial y} (h(x_1, y) - h(x_2, y)) \, dy \geq 0,
\]

and thus,

\[
h(x_1, y_1) - h(x_2, y_1) - (h(x_1, y_2) - h(x_2, y_2)) \geq 0,
\]

or equivalently,

\[
h(x_1, y_1) - h(x_2, y_1) \geq h(x_1, y_2) - h(x_2, y_2),
\]

which shows that \( h \) satisfies increasing differences and hence is supermodular.
⇒) Suppose \( h \) is supermodular. Then, it satisfies increasing differences and so, for \( x_1 > x_2, y_1 > y, \)

\[
\frac{h(x_1, y_1) - h(x_1, y)}{y_1 - y} \geq \frac{h(x_2, y_1) - h(x_2, y)}{y_1 - y}.
\]

Letting \( y_1 \to y \), we have

\[
\frac{\partial}{\partial y} h(x_1, y) \geq \frac{\partial}{\partial y} h(x_2, y), \quad \text{when} \quad x_1 \geq x_2,
\]

implying that

\[
\frac{\partial^2}{\partial x \partial y} h(x, y) \geq 0.
\]

Note that the limit above defines a left derivative, but since \( f \) is differentiable, it is also the right
derivative.

2.6.3 Parametric monotonicity

Suppose \( S, \Theta \subset \mathbb{R} \), \( f : S \times \Theta \to \mathbb{R} \), and consider the optimization problem

\[
\max_{x \in S} f(x, \theta).
\]

Here, by \textit{parametric monotonicity} we mean that the higher the value of \( \theta \), the higher the maxi-
mizer \( x^*(\theta) \).

Let’s give some intuition for \textit{strictly increasing differences} implying \textit{parametric monotonicity}. We argue by contradiction. Suppose that in this maximization problem a solution exists for all \( \theta \in \Theta \) (e.g., suppose that \( f(\cdot, \theta) \) is continuous on \( S \) for each fixed \( \theta \), and that \( S \) is compact). Pick any two values \( \theta_1, \theta_2 \) with \( \theta_1 > \theta_2 \). Let \( x_1, x_2 \) be values that are optimal at \( \theta_1 \) and \( \theta_2 \), respectively. Thus,

\[
f(x_1, \theta_1) - f(x_2, \theta_1) \geq 0 \geq f(x_1, \theta_2) - f(x_2, \theta_2).
\]

Suppose \( f \) satisfies strictly increasing differences, and that \( \theta_1 > \theta_2 \), but parametric monotonicity
fails. Furthermore, assume \( x_1 < x_2 \). So, the vectors \( (x_2, \theta_1) \) and \( (x_1, \theta_2) \) satisfy \( x_2 > x_1 \) and \( \theta_1 > \theta_2 \).

By strictly increasing differences,

\[
f(x_2, \theta_1) - f(x_1, \theta_1) > f(x_2, \theta_2) - f(x_1, \theta_2),
\]

contradicting (4.2.3). So, we must have \( x_1 \geq x_2 \), where \( x_1, x_2 \) were arbitrary selections from the
sets of optimal actions at \( \theta_1 \) and \( \theta_2 \), respectively.

In summary, if \( S, \Theta \subset \mathbb{R} \), \textit{strictly increasing differences} imply \textit{monotonicity of optimal actions in
the parameter \( \theta \) \in \Theta.}

\^6Note that this concept is different from what is stated in the Envelope Theorem, which studies the marginal
change in the value of the maximized function, and not of the optimizer of that function:

\textbf{Envelope Theorem}: Consider a maximization problem: \( M(\theta) = \max_x f(x, \theta) \). Let \( x^*(\theta) \) be the argmax value
of \( x \) that solves the problem in terms of \( \theta \), i.e., \( M(\theta) = f(x^*(\theta), \theta) \). Assume that \( f \) is continuously differentiable
in \( (x, \theta) \), and that \( x^* \) is continuously differentiable in \( \theta \). Then,

\[
\frac{\partial}{\partial \theta} M(\theta) = \frac{\partial}{\partial \theta} f(y, \theta) \bigg|_{y = x^*(\theta)}.
\]
CHAPTER 2. DISCRETE DYNAMIC PROGRAMMING

Note: This result also holds for \( S \subset \mathbb{R}^n, n \geq 2 \), but the proof is different and requires additional assumptions. The problem of the extension of the previous argument to higher dimensional settings is that we cannot say anymore that \( x_1 \not\geq x_2 \) implies \( x_1 < x_2 \).

The following theorem relaxes the “strict” condition of the increasing differences to guarantee parametric monotonicity.

**Theorem 2.6.4** Let \( S \) be a compact sublattice of \( \mathbb{R}^n \), \( \Theta \) be a sublattice of \( \mathbb{R}^l \), and \( f : S \times \Theta \rightarrow \mathbb{R} \) be a continuous function on \( S \) for each fixed \( \theta \). Suppose that \( f \) satisfies increasing differences in \( (x, \theta) \), and is supermodular in \( x \) for each fixed \( \theta \). Let the correspondence \( D^* : \Theta \rightarrow S \) be defined by

\[
D^*(\theta) = \text{arg max}\{f(x, \theta) | x \in S\}.
\]

Then,

1. For each \( \theta \in \Theta \), \( D^*(\theta) \) is a nonempty compact sublattice of \( \mathbb{R}^n \), and admits a greatest element, denoted \( x^*(\theta) \).
2. \( x^*(\theta_1) \geq x^*(\theta_2) \) whenever \( \theta_1 > \theta_2 \).
3. If \( f \) satisfies strictly increasing differences in \( (x, \theta) \), then \( x_1 \geq x_2 \) for any \( x_1 \in D(\theta_1) \) and \( x_2 \in D(\theta_2) \), whenever \( \theta_1 > \theta_2 \).

**Proof:** For part (1): Since \( f \) is continuous on \( S \) for each fixed \( \theta \), and since \( S \) is compact, \( D^*(\theta) \neq \emptyset \) for each \( \theta \). Fix \( \theta \) and take a sequence \( \{x_p\} \) in \( D^*(\theta) \) converging to \( x \in S \). Then, for any \( y \in S \), since \( x_p \) is optimal, we have

\[
f(x_p, \theta) \geq f(y, \theta).
\]

Taking limit as \( p \rightarrow \infty \), and using the continuity of \( f(\cdot, \theta) \), we obtain

\[
f(x, \theta) \geq f(y, \theta),
\]

so \( x \in D^*(\theta) \). Therefore, \( D^*(\theta) \) is closed, and as a closed subset of a compact set \( S \), it is also compact. Now, we argue by contradiction: Let \( x \) and \( x' \) be distinct elements of \( D^*(\theta) \). If \( x \wedge x' \not\in D^*(\theta) \), we must have

\[
f(x \wedge x', \theta) < f(x, \theta) = f(x', \theta).
\]

However, supermodularity in \( x \) means

\[
\frac{f(x, \theta) + f(x', \theta)}{2} < f(x' \vee x, \theta) + f(x' \wedge x, \theta) < f(x' \vee x, \theta) + f(x, \theta),
\]

which implies

\[
f(x' \vee x, \theta) > f(x, \theta) = f(x', \theta),
\]

which in turn contradicts the presumed optimality of \( x \) and \( x' \) at \( \theta \). A similar argument also establishes that \( x \wedge x' \in D^*(\theta) \). Thus, \( D^*(\theta) \) is a sublattice of \( \mathbb{R}^n \), and as a nonempty compact sublattice of \( \mathbb{R}^n \), admits a greatest element \( x^*(\theta) \).
For part (2): Let $\theta_1$ and $\theta_2$ be given with $\theta_1 > \theta_2$. Let $x_1 \in D^*(\theta_1)$, and $x_2 \in D^*(\theta_2)$. Then, we have

\[
0 \leq f(x_1, \theta_1) - f(x_1 \lor x_2, \theta_1) \quad \text{(by optimality of } x_1 \text{ at } \theta_1)
\]
\[
\leq f(x_1 \lor x_2, \theta_1) - f(x_2, \theta_1) \quad \text{(by supermodularity in } x) \\
\leq f(x_1 \lor x_2, \theta_2) - f(x_2, \theta_2) \quad \text{(by increasing differences in } (x, \theta)) \\
\leq 0 \quad \text{(by optimality of } x_2 \text{ at } \theta_2),
\]

so equality holds at every point in this string. Now, suppose $x_1 = x^*(\theta_1)$ and $x_2 = x^*(\theta_2)$. Since equality holds at all points in the string, using the first equality we have

\[
f(x_1 \lor x_2, \theta_1) = f(x_1, \theta_1),
\]

and so $x_1 \lor x_2$ is also an optimal action at $\theta_1$. If $x_1 \not\geq x_2$, then we would have $x_1 \lor x_2 > x_1$, and this contradicts the definition of $x_1$ as the greatest element of $D^*(\theta_1)$. Thus, we must have $x_1 \geq x_2$.

For part (3): Suppose that $x_1 \in D^*(\theta_1), x_2 \in D^*(\theta_2)$. Suppose that $x_1 \not\geq x_2$. Then, $x_2 > x_1 \land x_2$. If $f$ satisfies strictly increasing differences, then since $\theta_1 > \theta_2$, we have

\[
f(x_2, \theta_1) - f(x_1 \land x_2, \theta_1) > f(x_2, \theta_2) - f(x_1 \land x_2, \theta_2),
\]

so the third inequality in the string above becomes strict, contradicting the equality.  

\[
\Box
\]

Remark: For the case where $S, \Theta \subset \mathbb{R}$, from Theorem 2.6.2 it can be seen that if $f$ is supermodular, it automatically verifies the hypotheses of Theorem 2.6.4, and therefore in principle supermodularity in $R^2$ constitutes a sufficient condition for parametric monotonicity. For a more general case in $\mathbb{R}^n, n > 2$, a related result follows.

For this general case, the definition of increasing differences is: For all $z \in Z$, for all distinct $i, j \in \{1, \ldots, n\}$, and for all $z'_i, z'_j$ such that

\[
z'_i \geq z_i, \quad \text{and} \quad z'_j \geq z_j;
\]

it is the case that

\[
f(z_{-ij}, z'_i, z'_j) - f(z_{-ij}, z_i, z'_j) \geq f(z_{-ij}, z'_i, z_j) - f(z_{-ij}, z_i, z_j).
\]

In words, $f$ has increasing differences on $Z$ if it has increasing differences in each pair $(z_i, z_j)$ when all other coordinates are held fixed at some value.

Theorem 2.6.5 A function $f : Z \subset \mathbb{R}^n \rightarrow \mathbb{R}$ is supermodular on $Z$ if and only if $f$ has increasing differences on $Z$.

Proof: The implication “$\Rightarrow$” can be proved by a slight modification of part (2) in Theorem 2.6.2. To prove “$\Leftarrow$”, pick any $z$ and $z'$ in $Z$. We are required to show that

\[
f(z) + f(z') \leq f(z \lor z') + f(z \land z').
\]
If \( z \geq z' \) or \( z \leq z' \), the inequality trivially holds. So, suppose \( z \) and \( z' \) are not comparable under \( \geq \). For notational convenience, arrange the coordinates of \( z \) and \( z' \) so that
\[
z \lor z' = (z'_1, \ldots, z'_k, z_{k+1}, \ldots, z_n),
\]
and
\[
z \land z' = (z_1, \ldots, z_k, z'_{k+1}, \ldots, z'_n).
\]
Note that since \( z \) and \( z' \) are not comparable under \( \geq \), we must have \( 0 < k < n \).

Now, for \( 0 \leq i \leq j \leq n \), define
\[
z^{i,j} = (z'_1, \ldots, z'_i, z_{i+1}, \ldots, z_j, z_{j+1}, \ldots, z'_n).
\]
Then, we have
\[
z^{0,k} = z \land z', \quad z^{k,n} = z \lor z', \quad z^{0,n} = z, \quad z^{k,k} = z'. \tag{2.6.2}
\]
Since \( f \) has increasing differences on \( Z \), it is the case that for all \( 0 \leq i < k \leq j < n \),
\[
f(z^{i+1,j+1}) - f(z^{i,j+1}) \geq f(z^{i+1,j}) - f(z^{i,j}).
\]
Therefore, we have for \( k \leq j < n \),
\[
f(z^{k,j+1}) - f(z^{0,j+1}) = \sum_{i=0}^{k-1} [f(z^{i+1,j+1}) - f(z^{i,j+1})] \geq \sum_{i=0}^{k-1} [f(z^{i+1,j}) - f(z^{i,j})] = f(z^{k,j}) - f(z^{0,j}).
\]
Since this inequality holds for all \( j \) satisfying \( k \leq j < n \), it follows that the LHS is at its highest value at \( j = n - 1 \), while the RHS is at its lowest value when \( j = k \). Therefore,
\[
f(z^{k,n}) - f(z^{0,n}) \geq f(z^{k,k}) - f(z^{0,k}).
\]
From (2.6.2), this is precisely the statement that
\[
f(z \lor z') - f(z) \geq f(z') - f(z \land z').
\]
Since \( z \) and \( z' \) were chosen arbitrarily, \( f \) is shown to be supermodular on \( Z \).

**Remark:** From Theorem 2.6.5, it is sufficient to prove supermodularity (or increasing differences) to prove parametric monotonicity.

### 2.6.4 Applications to DP

We include here a couple of examples that show how useful the concept of parametric monotonicity could be to characterize monotonicity properties of the optimal policy.
Example 2.6.1 (A gambling model with changing win probabilities)

- Consider a gambler who is allowed to bet any amount up to his present fortune at each play.
- He will win or lose that amount according to a given probability $p$.
- Before each gamble, the value of $p$ changes ($p \sim F$).
- Control: On each play, the gambler must decide, after the win probability is announced, how much to bet.
- Consider a sequence of $N$ gambles.
- Objective: Maximize the expected value of a given utility function $G$ of his final fortune $x$, where $G(x)$ is continuously differentiable and nondecreasing in $x$.
- State: $(x,p)$, where $x$ is his current fortune, and $p$ is the current win probability.
  - Assume indices run backward in time.

DP formulation:

Define the value function $V_k(x,p)$ as the maximal expected final utility for state $(x,p)$ when there are $k$ games left.

The DP algorithm is:

$$
V_0(x,p) = G(x),
$$

and for $k = N, N-1, \ldots, 1$,

$$
V_k(x,p) = \max_{0 \leq u \leq x} \left\{ p \int_0^1 V_{k-1}(x + u, \alpha) dF(\alpha) + (1-p) \int_0^1 V_{k-1}(x - u, \alpha) dF(\alpha) \right\}.
$$

Let $u_k(x,p)$ be the largest $u$ that maximizes this equation. Let $g_k(u,p)$ be the expression to maximize above, i.e.,

$$
g_k(u,p) = p \int_0^1 V_{k-1}(x + u, \alpha) dF(\alpha) + (1-p) \int_0^1 V_{k-1}(x - u, \alpha) dF(\alpha).
$$

Intuitively, for given $k$ and $x$, the optimal amount $u_k(x,p)$ to bet should be increasing in $p$. So, we would like to prove parametric monotonicity of $u_k(x,p)$ in $p$. To this end, it would be enough to prove increasing differences of $g_k(u,p)$ in $(u,p)$, or equivalently, it would be enough to prove supermodularity of $g_k(u,p)$ in $(u,p)$. Or it would be enough to prove

$$
\frac{\partial^2}{\partial u \partial p} g_k(u,p) \geq 0.
$$

The derivation proceeds as follows:

$$
\frac{\partial}{\partial p} g_k(u,p) = \int_0^1 V_{k-1}(x + u, \alpha) dF(\alpha) - \int_0^1 V_{k-1}(x - u, \alpha) dF(\alpha).
$$
Then, by the Leibniz rule\footnote{We would need to prove that $V_k(x,p)$ and $\frac{\partial}{\partial x}V_k(x,p)$ are continuous in $x$. A sufficient condition or that is that $\mu_k(x,p)$ is continuously differentiable in $x$.}

$$\frac{\partial^2}{\partial u \partial p} g_k(u,p) = \int_0^1 \frac{\partial}{\partial u} V_{k-1}(x + u, \alpha) dF(\alpha) - \int_0^1 \frac{\partial}{\partial u} V_{k-1}(x - u, \alpha) dF(\alpha).$$

Then,

$$\frac{\partial^2}{\partial u \partial p} g_k(u,p) \geq 0$$

if for all $\alpha$,

$$\frac{\partial}{\partial u} [V_{k-1}(x + u, \alpha) - V_{k-1}(x - u, \alpha)] \geq 0,$$

or equivalently, if for all $\alpha$,

$$V_{k-1}(x + u, \alpha) - V_{k-1}(x - u, \alpha)$$

increases in $u$, which follows if $V_{k-1}(z, \alpha)$ is increasing in $z$, which immediately holds because for $z' > z$,

$$V_{k-1}(z', \alpha) \geq V_{k-1}(z, \alpha),$$

since in the former we are maximizing over a bigger domain.

For $V_0(\cdot, \alpha)$, it holds because $G(z') \geq G(z)$. □

Example 2.6.2 (An optimal allocation problem subject to penalty costs)

- There are $N$ stages to construct $I$ components sequentially.
- At each stage, we allocate $u$ dollars for the construction of one component.
- If we allocate $u$, then the component constructed will be a success w.p. $P(u)$ (continuous, nondecreasing, with $P(0) = 0$).
- After each component is constructed, we are informed as to whether or not it is successful.
- If at the end of $N$ stages we are $j$ components short, we incur a penalty cost $C(j)$ (increasing, with $C(j) = 0$ for $j \leq 0$).
- Control: How much money to allocate in each stage to minimize the total expected cost (construction + penalty).
- State: Number of successful components still needed.
- Indices run backward in time.

**DP formulation:**

Define the value function $V_k(i)$ as the minimal expected remaining cost when state is $i$ and $k$ stages remain.

The DP algorithm is:

$$V_0(i) = C(i),$$

where $C(i)$ is the penalty cost for being $i$ components short.
and for \( k = N, N - 1, \ldots, 1, \) and \( i > 0, \)
\[
V_k(i) = \min_{u \geq 0} \{ u + P(u)V_{k-1}(i-1) + (1 - P(u))V_{k-1}(i) \}.
\] (2.6.3)

We set \( V_k(i) = 0, \forall i \leq 0, \) and for all \( k. \)

It follows immediately from the definition of \( V_k(i) \) and the monotonicity of \( C(i) \) that \( V_k(i) \) increases in \( i \) and decreases in \( k. \)

Let \( u_k(i) \) be the minimizer of (2.6.3). Two intuitive results should follow:

1. “The more we need, the more we should invest” (i.e., \( u_k(i) \) is increasing in \( i). \)

2. “The more time we have, the less we need to invest at each stage” (i.e., \( u_k(i) \) is decreasing in \( k). \)

Let’s determine conditions on \( C(\cdot) \) that make the previous two intuitions valid.

Define
\[
g_k(i, u) = u + P(u)V_{k-1}(i-1) + (1 - P(u))V_{k-1}(i).
\]

Minimizing \( g_k(i, u) \) is equivalent to maximizing \( (-g_k(i, u)). \) Then, in order to prove \( u_k(i) \) increasing in \( i, \) it is enough to prove \( (-g_k(i, u)) \) supermodular in \( (i, u), \) or \( g_k(i, u) \) submodular in \( (i, u). \) Note that here we are treating \( i \) as a continuous quantity.

So, \( u_k(i) \) increases in \( i \) if
\[
\frac{\partial^2}{\partial i \partial u} g_k(i, u) \leq 0.
\]

We compute this cross-partial derivative. First, we calculate
\[
\frac{\partial}{\partial u} g_k(i, u) = 1 + P'(u)[V_{k-1}(i-1) - V_{k-1}(i)],
\]
and then
\[
\frac{\partial^2}{\partial i \partial u} g_k(i, u) = P'(u) \frac{\partial}{\partial i} [V_{k-1}(i-1) - V_{k-1}(i)] \leq 0,
\]
so that \( u_k(i) \) increases in \( i \) if \( [V_{k-1}(i-1) - V_{k-1}(i)] \) decreases in \( i. \) Similarly, \( u_k(i) \) decreases in \( k \) if \( [V_{k-1}(i-1) - V_{k-1}(i)] \) increases in \( k. \) Therefore, submodularity gives a sufficient condition on \( g_k(i, u), \) which ensures the desired monotonicity of the optimal policy. For this example, we show below that if \( C(i) \) is convex in \( i, \) then \( [V_{k-1}(i-1) - V_{k-1}(i)] \) decreases in \( i \) and increases in \( k, \) ensuring the desired structure of the optimal policy.

Two results are easy to verify:

- \( V_k(i) \) is increasing in \( i, \) for a given \( k. \)
- \( V_k(i) \) is decreasing in \( k, \) for a given \( i. \)

**Proposition 2.6.1** If \( C(i + 2) - C(i + 1) \geq C(i + 1) - C(i) \) \( \forall i \) (i.e., \( C(\cdot) \) convex), then \( u_k(i) \) increases in \( i \) and decreases in \( k. \)
We proceed by induction on $n = k + i$. For $n = 0$ (i.e., $k = i = 0$):

\[ A_{0,0} : \quad V_1(1) - V_1(0) \leq V_0(1) - V_0(0), \]

\[ = 0 \text{ from (2.6.3)} \]

\[ B_{0,0} : \quad V_1(0) - V_0(0) \leq V_0(1), \]

\[ C_{0,0} : \quad V_0(1) - V_0(0) \leq V_0(2) - V_0(1), \]

where the last inequality holds because $C(\cdot)$ is convex.

IH: The 3 inequalities above are true for $k + i < n$.

Suppose now that $k + i = n$. We proceed by proving one inequality at a time.

1. For $A_{i,k}$:
   - If $i = 0$ $\Rightarrow A_{0,k} : V_{k+1}(1) - V_{k+1}(0) \leq V_k(1) - V_k(0)$, which holds because $V_k(i)$ is decreasing in $k$.
   - If $i > 0$, then there is $\bar{u}$ such that
     \[ V_{k+1}(i) = \bar{u} + P(\bar{u})V_k(i - 1) + (1 - P(\bar{u}))V_k(i). \]

   Thus,
   \[ V_{k+1}(i) - V_k(i) = \bar{u} + P(\bar{u})[V_k(i - 1) - V_k(i)] \quad (2.6.4) \]

   Also, since $\bar{u}$ is the minimizer just for $V_{k+1}(i)$,
   \[ V_{k+1}(i + 1) \leq \bar{u} + P(\bar{u})V_k(i) + (1 - P(\bar{u}))V_k(i + 1). \]

   Then,
   \[ V_{k+1}(i + 1) - V_k(i + 1) \leq \bar{u} + P(\bar{u})[V_k(i) - V_k(i + 1)] \quad (2.6.5) \]

   Note that from $C_{i-1,k}$ (which holds by IH because $i - 1 + k = n - 1$), we get
   \[ V_k(i) - V_k(i + 1) \leq V_k(i - 1) - V_k(i) \]

   Then, using the RHS of (4.2.5) and (5.3.2), we have
   \[ V_{k+1}(i + 1) - V_k(i + 1) \leq V_{k+1}(i) - V_k(i), \]

   or equivalently,
   \[ V_{k+1}(i + 1) - V_{k+1}(i) \leq V_k(i + 1) - V_k(i), \]

   which is exactly $A_{i,k}$. 

\[ \]
2. For $B_{i,k}$:

Note that for some $\bar{u}$,

$$V_{k+2}(i) = \bar{u} + P(\bar{u})V_{k+1}(i) + (1 - P(\bar{u}))V_{k+1}(i),$$

or equivalently,

$$V_{k+2}(i) - V_{k+1}(i) = \bar{u} + P(\bar{u})[V_{k+1}(i) - V_{k+1}(i)].$$

(2.6.6)

Also, since $\bar{u}$ is the minimizer for $V_{k+2}(i)$,

$$V_{k+1}(i) \leq \bar{u} + P(\bar{u})V_k(i) + (1 - P(\bar{u}))V_k(i),$$

so that

$$V_{k+1}(i) - V_k(i) \leq \bar{u} + P(\bar{u})[V_k(i) - V_k(i)].$$

(2.6.7)

By IH, $A_{i-1,k}$, for $i - 1 + k = n - 1$, holds. So,

$$V_{k+1}(i) - V_k(i) \leq V_k(i) - V_k(i - 1),$$

or equivalently,

$$V_k(i - 1) - V_k(i) \leq V_{k+1}(i - 1) - V_{k+1}(i).$$

Plugging it in (3.3.4), and using the RHS of (4.2.7), we obtain

$$V_{k+1}(i) - V_k(i) \leq V_{k+2}(i) - V_{k+1}(i),$$

which is exactly $B_{i,k}$.

3. For $C_{i,k}$, we first note that $B_{i+1,k-1}$ (already proved since $i + 1 + k - 1 = n$) states that

$$V_k(i + 1) - V_{k-1}(i + 1) \leq V_{k+1}(i + 1) - V_k(i + 1),$$

or equivalently,

$$2V_k(i + 1) \leq V_{k+1}(i + 1) + V_{k-1}(i + 1).$$

(2.6.8)

Hence, if we can show that,

$$V_{k-1}(i + 1) + V_{k+1}(i + 1) \leq V_k(i) + V_k(i + 2),$$

(2.6.9)

then from (2.6.8) and (2.6.9) we would have

$$2V_k(i + 1) \leq V_k(i) + V_k(i + 2),$$

or equivalently,

$$V_k(i + 1) - V_k(i) \leq V_k(i + 2) - V_k(i + 1),$$

which is exactly $C_{i,k}$.

Now, for some $\bar{u}$,

$$V_k(i + 2) = \bar{u} + P(\bar{u})V_{k-1}(i + 1) + (1 - P(\bar{u}))V_{k-1}(i + 2),$$

which implies

$$V_k(i + 2) - V_{k-1}(i + 1) = \bar{u} + P(\bar{u})V_{k-1}(i + 1) + (1 - P(\bar{u}))V_{k-1}(i + 2) - V_{k-1}(i + 1)$$

$$= \bar{u} + (1 - P(\bar{u}))[V_{k-1}(i + 2) - V_{k-1}(i + 1)].$$

(2.6.10)
Moreover, since $\bar{u}$ is the minimizer of $V_k(i + 2)$:

$$V_{k+1}(i + 1) \leq \bar{u} + P(\bar{u})V_k(i) + (1 - P(\bar{u}))V_k(i + 1).$$

Subtracting $V_k(i)$ from both sides:

$$V_{k+1}(i + 1) - V_k(i) \leq \bar{u} + (1 - P(\bar{u}))[V_k(i + 1) - V_k(i)].$$

Then, equation (2.6.9) will follow if we can prove that

$$V_k(i + 1) - V_k(i) \leq V_{k-1}(i + 2) - V_{k-1}(i + 1),$$

because then

$$V_{k+1}(i + 1) - V_k(i) \leq \bar{u} + (1 - P(\bar{u}))[V_{k-1}(i + 2) - V_{k-1}(i + 1)]
= V_k(i + 2) - V_{k-1}(i + 1).$$

Now, from $A_{i,k-1}$ (which holds by IH), it follows that

$$V_k(i + 1) - V_k(i) \leq V_{k-1}(i + 1) - V_{k-1}(i).$$

(2.6.12)

Also, from $C_{i,k-1}$ (which holds by IH), it follows that

$$V_{k-1}(i + 1) - V_{k-1}(i) \leq V_{k-1}(i + 2) - V_{k-1}(i + 1).$$

(2.6.13)

Finally, (2.6.12) and (2.6.13) $\Rightarrow$ (2.6.11) $\Rightarrow$ (2.6.9), and we close this case.

In the end, the three inequalities hold, and the proof is completed. ▪

2.7 Extensions

2.7.1 The Value of Information

The value of information is the reduction in cost between optimal closed-loop and open-loop policies. To illustrate its computation, we revisit the two-game chess match example.

Example 2.7.1 (Two-game chess match)

- **Closed-Loop**: Recall that the optimal policy when $p_d > p_w$ is to play timid if and only if one is ahead in the score. Figure 2.7.1 illustrates this. The optimal payoff under the closed-loop policy is the sum of the payoffs in the leaves of Figure 2.7.1. This is because the four payoffs correspond to four mutually exclusive outcomes. The total payoff is

$$\mathbb{P}(\text{win}) = p_wp_d + p_w^2(1 - p_d) + (1 - p_w)(1 - p_w)p_d.$$  

(2.7.1)

For example, if $p_w = 0.45$ and $p_d = 0.9$, we know that $\mathbb{P}(\text{win}) = 0.53$. 
Figure 2.7.1: Optimal closed-loop policy for the two-game chess match. The payoffs included next to the leaves represent the total cumulative payoff for following that particular branch from the root.

- Open-Loop: There are four possible policies:

  Note that the latter two policies lead to the same payoff, and that this payoff dominates the first policy (i.e., playing (timid-timid)) because

  \[ p_w p_d + p_w^2 (1 - p_d) \geq p_d^2 p_w. \]

  Therefore, the maximum open-loop probability of winning the match is:

  \[
  \max \{ p_w^2 (3 - 2 p_w), \quad p_w p_d + p_w^2 (1 - p_d) \} = p_w^2 + p_w (1 - p_w) \max \{ 2 p_w, p_d \} \quad (2.7.2)
  \]

  So,

  - if \( p_d > 2 p_w \), then the optimal policy is to play either (timid,bold) or (bold, timid);
  - if \( p_d \leq 2 p_w \), then the optimal policy is to play (bold,bold).

  Again, if \( p_w = 0.45 \) and \( p_d = 0.9 \), then \( P(\text{win}) = 0.425 \)

- For the aforementioned probabilities \( p_w = 0.45 \) and \( p_d = 0.9 \), the value of information is the difference between both optimal payoffs: \( 0.53 - 0.425 = 0.105 \).

  More generally, by subtracting (2.7.1)-(2.7.2):

  \[
  \text{Value of Information} = p_w^2 (2 - p_w) + p_w (1 - p_w) p_d - p_w^2 - p_w (1 - p_w) \max \{ 2 p_w, p_d \} = p_w (1 - p_w) \min \{ p_w, p_d - p_w \}. \quad \square
  \]

2.7.2 State Augmentation

In the basic DP formulation, the random noise is independent across all periods, and the control depends just on the current state. In this regard, the system is of the Markovian type. In order to deal with a more general situation, we enlarge the state definition so that the current state captures information of the past. In some applications, this past information could be helpful for the future.
Figure 2.7.2: Open-loop policies for the two-game chess match. The payoffs included next to the leaves represent the total cumulative payoff for following that particular branch from the root.

Time Lags

Suppose that the next state $x_{k+1}$ depends on the last two states $x_k$ and $x_{k-1}$, and on the last two controls $u_k$ and $u_{k-1}$. For instance,

$$x_{k+1} = f_k(x_k, x_{k-1}, u_k, u_{k-1}, w_k), \quad k = 1, \ldots, N - 1,$$

$$x_1 = f_0(x_0, u_0, w_0).$$

We redefine the system equation as follows:

$$
\begin{pmatrix}
  x_{k+1} \\
  y_{k+1} \\
  s_{k+1} \\
  \bar{x}_{k+1}
\end{pmatrix}
= 
\begin{pmatrix}
f_k(x_k, y_k, u_k, s_k, w_k) \\
x_k \\
u_k \\
\bar{f}_k(\bar{x}_k, u_k, w_k)
\end{pmatrix},
$$

where $\bar{x}_k = (x_k, y_k, s_k) = (x_k, x_{k-1}, u_{k-1})$.

**DP algorithm**
When the DP algorithm for the reformulated problem is translated in terms of the variables of the original problem, it takes the form:

\[
J_N(x_N) = g_N(x_N),
\]

\[
J_{N-1}(x_{N-1}, x_{N-2}, u_{N-2}) = \min_{u_{N-1} \in U_{N-1}(x_{N-1})} \text{E}_{w_{N-1}} \left\{ g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) + J_N \left( f_{N-1}(x_{N-1}, x_{N-2}, u_{N-1}, w_{N-1}) \right) \right\},
\]

\[
J_k(x_k, x_{k-1}, \ldots, u_{k-1}) = \min_{u_k \in U_k(x_k)} \text{E}_{w_k} \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, x_{k-1}, u_k, u_{k-1}, w_k), x_k, u_k) \right\},
\]

\[
J_0(x_0) = \min_{u_0 \in U_0(x_0)} \text{E}_{w_0} \left\{ g_0(x_0, u_0, w_0) + J_1(f_0(x_0, u_0, w_0), x_0, u_0) \right\}.
\]

**Correlated Disturbances**

Assume that \( w_0, w_1, \ldots, w_{N-1} \) can be represented as the output of a linear system driven by independent r.v. For example, suppose that disturbances can be modeled as:

\[
w_k = C_k y_{k+1}, \text{ where } y_{k+1} = A_k y_k + \xi_k, \quad k = 0, 1, \ldots, N - 1,
\]

where \( C_k, A_k \) are matrices of appropriate dimension, and \( \xi_k \) are independent random vectors. By viewing \( y_k \) as an additional state variable; we obtain the new system equation:

\[
\begin{pmatrix}
    x_{k+1} \\
    y_{k+1}
\end{pmatrix} =
\begin{pmatrix}
    f_k(x_k, u_k, C_k(A_k y_k + \xi_k)) \\
    A_k y_k + \xi_k
\end{pmatrix},
\]

for some initial \( y_0 \). In period \( k \), this correlated disturbance can be represented as the output of a linear system driven by independent random vectors:

\[
\xi_k \rightarrow y_{k+1} = A_k y_k + \xi_k \rightarrow y_{k+1} \rightarrow C_k \rightarrow w_k
\]

Observation: In order to have perfect state information, the controller must be able to observe \( y_k \). This occurs for instance when \( C_k \) is the identity matrix, and therefore \( w_k = y_{k+1} \). Since \( w_k \) is realized at the end of period \( k \), its known value is carried over the next period \( k + 1 \) through the state component \( y_{k+1} \).

**DP algorithm**

\[
J_N(x_N, y_N) = g_N(x_N)
\]

\[
J_k(x_k, y_k) = \min_{u_k \in U_k(x_k)} \text{E}_{\xi_k} \left\{ g_k(x_k, u_k, C_k(A_k y_k + \xi_k)) + J_{k+1} \left( f_k(x_k, u_k, C_k(A_k y_k + \xi_k), x_{k+1}, y_{k+1} \right) \right\}
\]

98
2.7.3 Forecasts

Suppose that at time \( k \), the controller has access to a forecast \( y_k \) that results in a reassessment of the probability distribution of \( w_k \).

In particular, suppose that at the beginning of period \( k \), the controller receives an accurate prediction that the next disturbance \( w_k \) will be selected according to a particular prob. distribution from a collection \( \{ Q_1, Q_2, \ldots, Q_m \} \). For example if a forecast is \( i \), then \( w_k \) is selected according to a probability distribution \( Q_i \). The a priori probability that the forecast will be \( i \) is denoted by \( p_i \) and is given.

System equation:
\[
\begin{pmatrix}
  x_{k+1} \\
  y_{k+1}
\end{pmatrix} =
\begin{pmatrix}
  f_k(x_k, u_k, w_k) \\
  \xi_k
\end{pmatrix},
\]
where \( \xi_k \) is the r.v. taking value \( i \) w.p. \( p_i \). So, when \( \xi_k \) takes the value \( i \), then \( w_{k+1} \) will occur according to distribution \( Q_i \). Note that there are two sources of randomness now: \( \tilde{w}_k = (w_k, \xi_k) \), where \( w_k \) stands for the outcome of the previous forecast in the current period, and \( \xi_k \) passes the new forecast to the next period.

DP algorithm
\[
J_N(x_N, y_N) = g_N(x_N)
\]
\[
J_k(x_k, y_k) = \min_{u_k \in U_k(x_k)} \mathbb{E}_{w_k} \left[ g_k(x_k, u_k, w_k) + \sum_{i=1}^m p_i J_{k+1}(f_k(x_k, u_k, w_k), i)/y_k \right]
\]
So, in current period \( k \), the forecast \( y_k \) is known (given), and it determines the distribution for the current noise \( w_k \). For the future, the forecast is \( i \) w.p. \( p_i \).

2.7.4 Multiplicative Cost Functional

The basic formulation of the DP problem assumes that the cost functional is additive over time. That is, every period (depending on states, actions and uncertainty) the system generates a cost and it is the sum of these single-period costs that we are interested to minimize. It should be relatively clear by now why this additivity assumption is crucial for the DP method to work. However, what we really need is an appropriate form of separability of the cost functional into its single-period components and additivity is one convenient (and most natural form most practical applications) form to ensure this separability, but is not the only one. The following exercises clarify this point.

Exercise 2.7.1 In the framework of the basic problem, consider the case where the cost is of the form
\[
\mathbb{E}_w \left[ \exp \left( g_N(x_N) + \sum_{k=1}^{N-1} g_k(x_k, u_k, w_k) \right) \right].
\]
a) Show that the optimal cost and optimal policy can be obtained from the DP-like algorithm
\[
J_N(x_N) = \exp(g_N(x_N)), \quad J_k(x_k) = \min_{u_k \in U_k} \mathbb{E} \left[ J_{k+1}(f_k(x_k, u_k, w_k)) \exp(g_k(x_k, u_k, w_k)) \right].
\]
b) Define the functions $V_k(x_k) = \ln(J_k(x_k))$. Assume also that $g_k(x, u, w) = g_k(x, u)$, that is, the $g_k$ are independent of $w_k$. Show that the above algorithm can be rewritten as follows:

$$V_N(x_N) = g_N(x_N),$$

$$V_k(x_k) = \min_{u_k \in U_k} \{g_k(x_k, u_k) + \ln(\mathbb{E}\{\exp(V_{k+1}(f_k(x_k, u_k, w_k)))\})\}.$$ 

**Exercise 2.7.2** Consider the case where the cost has the following multiplicative form

$$\mathbb{E}_w \left[ g_N(x_N) \prod_{k=1}^{N} g_k(x_k, u_k, w_k) \right].$$

Develop a DP-like algorithm for this problem assuming $g_k(x_k, u_k, w_k) \geq 0$ for all $x_k, u_k$ and $w_k$. 


Chapter 3

Applications

3.1 Inventory Control

In this section, we study the inventory control problem discussed in Example 2.1.1.

3.1.1 Problem setup

We assume the following:

• Excess demand in each period is backlogged and is filled when additional inventory becomes available, i.e.,

\[ x_{k+1} = x_k + u_k - w_k, \quad k = 0, 1, \ldots, N - 1. \]

• Demands \( w_k \) take values within a bounded interval and are independent.

• Cost of state \( x \):

\[ r(x) = p \max\{0, -x\} + h \max\{0, x\}, \]

where \( p \geq 0 \) is the per-unit backlog cost, and \( h \geq 0 \) is the per-unit holding cost.

• Per-unit purchasing cost \( c \).

• Total expected cost to be minimized:

\[ E \left[ \sum_{k=0}^{N-1} (cu_k + p \max\{0, w_k - x_k - u_k\} + h \max\{0, x_k + u_k - w_k\}) \right], \]

where the costs are incurred based on the inventory (potentially, negative) available at the end of each period \( k \).

• Suppose that \( p > c \) (otherwise, if \( c \geq p \), it would never be optimal to buy stock in the last period \( N - 1 \) and possibly in the earlier periods).

• Most of the subsequent analysis generalizes to the case where \( r(\cdot) \) is a convex function that grows to infinity with asymptotic slopes \( p \) and \( h \) as its argument tends to \( -\infty \) and \( \infty \), respectively.
Figure 3.1.1 illustrates the problem setup and the dynamics of the system. By applying the DP algorithm, we have

\[ J_N(x_N) = 0, \]

\[ J_k(x_k) = \min_{u_k \geq 0} \{ cu_k + H(x_k + u_k) + E_{w_k}[J_{k+1}(x_k + u_k - w_k)] \}, \] (3.1.1)

where

\[ H(y) = E[r(y - w_k)] = pE[(w_k - y)^+] + hE[(y - w_k)^+]. \]

If the probability distribution of \( w_k \) is time-varying, then \( H \) depends on \( k \). To simplify notation in what follows, we will assume that all demands are identically distributed.

By defining \( y_k = x_k + u_k \geq 0 \) (i.e., \( y_k \) is the inventory level right after getting the new units, and before demand for the period is realized), we could write

\[ J_k(x_k) = \min_{y_k \geq x_k} \{ cy_k + H(y_k) + E_{w_k}[J_{k+1}(y_k - w_k)] \} - cx_k. \] (3.1.2)

### 3.1.2 Structure of the cost function

- Note that \( H(y) \) is convex, since for a given \( w_k \), both terms in its definition are convex (Figure 3.1.2 illustrates this) \( \Rightarrow \) the sum is convex \( \Rightarrow \) taking expectation on \( w_k \) preserves convexity.

- Assume \( J_{k+1}(\cdot) \) is convex (to be proved later), then the function \( G_k(y) \) minimized in (3.1.2) is convex. Suppose for now that there is an unconstrained minimum \( S_k \) (existence to be verified); that is, for each \( k \), the scalar \( S_k \) minimizes the function

\[ G_k(y) = cy + H(y) + E_{w_k}[J_{k+1}(y - w)]. \]

In addition, if \( G_k(y) \) has the shape shown in Figure 3.1.3, then the minimizer of \( G_k(y) \), for \( y_k \geq x_k \), is

\[ y_k^* = \begin{cases} S_k & \text{if } x_k < S_k \\ x_k & \text{if } x_k \geq S_k. \end{cases} \]
\[ \text{Max}\{0,w_k \cdot y\} \]
\[ \text{Max}\{0,y-w_k\} \]

**Figure 3.1.2:** Graphical illustration of the two terms in the \(H\) function.

\[ G_k(y) \]
\[ x_{k1} \quad S_k \quad x_{k2} \]

**Figure 3.1.3:** The function \(G_k(y)\) has a “bowl shape”. The minimum for \(y\) \(\geq x_{k1}\) is \(S_k\); the minimum for \(y\) \(\geq x_{k2}\) is \(x_{k2}\).

- Using the reverse transformation \(u_k = y_k - x_k\) (recall that \(u_k\) is the amount ordered), then an optimal policy is determined by a sequence of scalars \(\{S_0, S_1, \ldots, S_{N-1}\}\) and has the form

\[
\mu_k^*(x_k) = \begin{cases} 
S_k - x_k & \text{if } x_k < S_k \\
0 & \text{if } x_k \geq S_k.
\end{cases}
\]  

(3.1.3)

This control is known as *basestock policy*, with *basestock level* \(S_k\).

To complete the proof of the optimality of the control policy (3.1.3), we need to prove the next result:

**Proposition 3.1.1** The following three facts hold:

1. The value function \(J_{k+1}(y)\) (and hence, \(G_k(y)\)) is convex in \(y\), \(k = 0, 1, \ldots, N - 1\).
2. \(\lim_{|y| \to \infty} G_k(y) = \infty, \forall k\).
3. \(\lim_{|y| \to \infty} J_k(y) = \infty, \forall k\).
PROOF: By induction. For \( k = N - 1 \),

\[
G_{N-1}(y) = cy + H(y) + E_w[J_N(y - w)],
\]

and since \( H(\cdot) \) is convex, \( G_{N-1}(y) \) is convex. For \( y \) “very negative”, \( \frac{\partial}{\partial y} H(y) = -p \), and so \( \frac{\partial}{\partial y} G_{N-1}(y) = c - p < 0 \). For \( y \) “very positive”, \( \frac{\partial}{\partial y} H(y) = h \), and so \( \frac{\partial}{\partial y} G_{N-1}(y) = c + h > 0 \). So, \( \lim_{|y| \to \infty} G_{N-1}(y) = \infty \). Hence, the optimal control for the last period turns out to be

\[
\mu^*_{N-1}(x_{N-1}) = \begin{cases} 
  S_{N-1} - x_{N-1} & \text{if } x_{N-1} < S_{N-1} \\
  0 & \text{if } x_{N-1} \geq S_{N-1},
\end{cases}
\]

and from the DP algorithm in (3.1.1), we get

\[
J_{N-1}(x_{N-1}) = \begin{cases} 
  c(S_{N-1} - x_{N-1}) + H(S_{N-1}) & \text{if } x_{N-1} < S_{N-1} \\
  H(x_{N-1}) & \text{if } x_{N-1} \geq S_{N-1}.
\end{cases}
\]

Before continuing, we need the following auxiliary result:

**Claim:** \( J_{N-1}(x_{N-1}) \) is convex in \( x_{N-1} \).

**Proof:** Note that we can write

\[
J_{N-1}(x) = \begin{cases} 
  -cx + cS_{N-1} + H(S_{N-1}) & \text{if } x < S_{N-1} \\
  H(x) & \text{if } x \geq S_{N-1}.
\end{cases}
\]

Figure 3.1.4 illustrates the convexity of the function \( G_{N-1}(y) = cy + H(y) \). Recall that we had denoted \( S_{N-1} \) the unconstrained minimizer of \( G_{N-1}(y) \). The unconstrained minimizer \( H^* \) of the function \( H(y) \) occurs to the right of \( S_{N-1} \). To verify this, compute

\[
\frac{\partial}{\partial y} G_{N-1}(y) = c + \frac{\partial}{\partial y} H(y).
\]

\footnote{Note that \( G_{N-1}(y) \) is shifted one index back in the argument to show convexity, since given the convexity of \( J_N(\cdot) \), it turns out to be convex. However, we still need to prove the convexity of \( J_{N-1}(\cdot) \).}
Evaluating the derivative at $S_{N-1}$, we get
\[
\frac{\partial}{\partial y} G_{N-1}(S_{N-1}) = c + \frac{\partial}{\partial y} H(S_{N-1}) = 0,
\]
and therefore, $\frac{\partial}{\partial y} H(S_{N-1}) = -c < 0$; that is, $H(\cdot)$ is decreasing at $S_{N-1}$, and thus its minimum $H^*$ occurs to its right.

Figure 3.1.5 plots $J_{N-1}(x_{N-1})$. Note that according to (3.1.5), the function is linear to the left of $S_{N-1}$, and tracks $H(x_{N-1})$ to the right of $S_{N-1}$. The minimum value of $J_{N-1}(\cdot)$ occurs at $x_{N-1} = H^*$, but we should be cautious on how to interpret this fact: This is the “best possible state” that the controller can reach, however, the purpose of DP is to prescribe the best course of action for any initial state $x_{N-1}$ at period $N - 1$, which is given by the optimal control (3.1.4) above.

Continuing with the proof of Proposition 3.1.1, so far we have that given the convexity of $J_N(x)$, we prove the convexity of $G_{N-1}(x)$, and then the convexity of $J_{N-1}(x)$. Furthermore, Figure 3.1.5 also shows that
\[
\lim_{|y| \to \infty} J_{N-1}(y) = \infty.
\]
The argument can be repeated to show that for all $k = N - 2, \ldots, 0$, if $J_{k+1}(x)$ is convex, $\lim_{|y| \to \infty} J_{k+1}(y) = \infty$, and $\lim_{|y| \to \infty} G_k(y) = \infty$, then we have
\[
J_k(x_k) = \begin{cases} 
  c(S_k - x_k) + H(S_k) + E[J_{k+1}(S_k - w_k)] & \text{if } x_k < S_k \\
  H(x_k) + E[J_{k+1}(x_k - w_k)] & \text{if } x_k \geq S_k,
\end{cases}
\]

where $S_k$ minimizes $G_k(y) = cy + H(y) + E[J_{k+1}(y - w)]$. Furthermore, $J_k(y)$ is convex, $\lim_{|y| \to \infty} J_k(y) = \infty$, $G_{k-1}(y)$ is convex, and $\lim_{|y| \to \infty} G_{k-1}(y) = \infty$.  

\footnote{Note that the function $H(y)$, on a sample path basis, is not differentiable everywhere (see Figure 3.1.2). However, the probability of the r.v. hitting the value $y$ is zero if $w$ has a continuous density, and so we can assert that $H(\cdot)$ is differentiable w.p.1.}
CHAPTER 3. APPLICATIONS

Technical note: To formally complete the proof above, when taking derivative of \( G_k(y) \), that will involve taking derivative of a expected value. Under relatively mild technical conditions, we can safely interchange differentiation and expectation. For example, it is safe to do that when the density \( f_w(w) \) of the r.v. does not depend on \( y \). More formally, if \( R_w \) is the support of the r.v. \( w \),

\[
\frac{\partial}{\partial x} E[g(x, w)] = \frac{\partial}{\partial x} \int_{w \in R_w} g(x, w) f_w(w) \, dw.
\]

Using Leibniz’s rule, if the function \( f_w(w) \) does not depend on \( x \), the set \( R_w \) does not depend on \( x \) either, and the derivative \( \frac{\partial}{\partial x} g(x, w) \) is well defined and bounded, we can interchange derivative and integral:

\[
\frac{\partial}{\partial x} \int_{w \in R_w} g(x, w) f_w(w) \, dw = \int_{w \in R_w} \left( \frac{\partial}{\partial x} g(x, w) \right) f_w(w) \, dw,
\]

and so

\[
\frac{\partial}{\partial x} E[g(x, w)] = E \left[ \frac{\partial}{\partial x} g(x, w) \right].
\]

3.1.3 Positive fixed cost and \((s, S)\) policies

Suppose that there is a fixed cost \( K > 0 \) associated with a positive inventory order, i.e., the cost of ordering \( u \geq 0 \) units is:

\[
C(u) = \begin{cases} 
K + cu & \text{if } u > 0 \\
0 & \text{if } u = 0.
\end{cases}
\]

The DP algorithm takes the form

\[
J_N(x_N) = 0
\]

\[
J_k(x_k) = \min_{u_k \geq 0} \left\{ C(u_k) + H(x_k + u_k) + E_{w_k}[J_{k+1}(x_k + u_k - w_k)] \right\},
\]

where again

\[
H(y) = pE[(w - y)^+] + hE[(y - w)^+].
\]

Consider again

\[
G_k(y) = cy + H(y) + E[J_{k+1}(y - w)].
\]

Then,

\[
J_k(x_k) = \min \left\{ \frac{G_k(x_k)}{\text{Do not order } u_k}, \min_{u_k > 0} \left\{ K + G_k(x_k + u_k) \right\} \right\} - cx_k.
\]

By changing variable \( y_k = x_k + u_k \) like in the zero fixed-cost case, we get

\[
J_k(x_k) = \min \left\{ G_k(x_k), \min_{y_k > x_k} \left\{ K + G_k(y_k) \right\} \right\} - cx_k.
\]

When \( K > 0 \), \( G_k \) is not necessarily convex\(^3\), opening the possibility of very complicated optimal policies (see Figure 3.1.6). Under this kind of function \( G_k(y) \), for the cost function (3.1.6), the optimal policy would be:

\(^3\)Note that \( G_k \) involves \( K \) through \( J_{k+1} \).
CHAPTER 3. APPLICATIONS

1. If \( x_k \in \text{Zone I} \Rightarrow G_k(x_k) > G_k(s), \forall x_k < s \Rightarrow \text{Order } u_k^* = S - x_k, \text{ such that } y_k^* = S. \) Clearly, \( G_k(S) + K < G_k(x_k), \forall x_k < s. \) So, if \( x_k \in \text{Zone I}, \) \( u_k^* = S - x_k. \)

2. If \( x_k \in \text{Zone II} \Rightarrow \)
   - If \( s < x_k < S \) and \( u > 0 \) (i.e., \( y_k > x_k \)) \( K + G_k(y_k) > G_k(x_k), \) and it is suboptimal to order.
   - If \( S < x_k < y_k' \) and \( u > 0 \) (i.e., \( y_k > x_k \)) \( K + G_k(y_k) > G_k(x_k), \) and it is also suboptimal to order.

So, if \( x_k \in \text{Zone II}, \) \( u_k^* = 0. \)

3. If \( x_k \in \text{Zone III} \Rightarrow \) Order \( u_k^* = \tilde{S} - x_k, \) so that \( y_k^* = \tilde{S}, \) and \( G_k(x_k) > K + G(\tilde{S}), \) for all \( y_k' < x_k < \tilde{s}. \)

4. If \( x_k \in \text{Zone IV} \Rightarrow \) Do not order (i.e., \( u_k^* = 0), \) since otherwise \( K + G_k(y_k) > G_k(x_k), \forall y_k > x_k. \)

In summary, the optimal policy would be to order \( u_k^* = (S - x) \) in zone I, \( u_k^* = 0 \) in zones II and IV, and \( u_k^* = (\tilde{S} - x) \) in zone III.

We will show below that even though the functions \( G_k \) may not be convex, they do have some structure: they are \( K \)-convex.

**Definition 3.1.1** A real function \( g(y) \) is \( K \)-convex if and only if it verifies the property:

\[
K + g(z + y) \geq g(y) + z \left( \frac{g(y) - g(y - b)}{b} \right),
\]

for all \( z \geq 0, b > 0, y \in \mathbb{R}. \)

The definition is illustrated in Figure 3.1.7. **Observation:** Note that the situation described in Figure 3.1.6 is impossible under \( K \)-convexity: Since \( y_0 \) is a local maximum in zone III, we must have for \( b > 0 \) small enough,

\[
G_k(y_0) - G_k(y_0 - b) \geq 0 \Rightarrow \frac{G_k(y_0) - G_k(y_0 - b)}{b} \geq 0,
\]
and from the definition of $K$-convexity, we should have for $\bar{S} = y_0 + z$, and $y = y_0$,

$$K + G_k(\bar{S}) \geq G_k(y_0) + z \left( \frac{G_k(y_0) - G_k(y_0 - b)}{b} \right) \geq 0,$$

which does not hold in our case.

**Intuition:** A $K$-convex function is a function that is “almost convex”, and for which $K$ represents the size of the “almost”. Scarf(1960) invented the notion of $K$-convex functions for the explicit purpose of analyzing this inventory model.

For a function $f$ to be $K$-convex, it must lie below the line segment connecting $(x, f(x))$ and $(y, K + f(y))$, for all real numbers $x$ and $y$ such that $x \leq y$. Figure 3.1.8 below shows that a $K$-convex function, namely $f_1$, need not be continuous. However, it can be shown that a $K$-convex function cannot have a positive jump at a discontinuity, as illustrated by $f_2$. Moreover, a negative jump cannot be too large, as illustrated by $f_3$.

Next, we compile some results on $K$-convex functions:

**Lemma 3.1.1** Properties of $K$-convex functions:

(a) A real-valued convex function $g$ is 0-convex and hence also $K$-convex for all $K > 0$.

(b) If $g_1(y)$ and $g_2(y)$ are $K$-convex and $L$-convex respectively, then $\alpha g_1(y) + \beta g_2(y)$ is $(\alpha K + \beta L)$-convex, for all $\alpha, \beta > 0$.

(c) If $g(y)$ is $K$-convex and $w$ is a random variable, then $E_w[g(y - w)]$ is also $K$-convex, provided $E_w[|g(y - w)|] < \infty$, for all $y$.

(d) If $g$ is a continuous $K$-convex function and $g(y) \to \infty$ as $|y| \to \infty$, then there exist scalars $s$ and $S$, with $s \leq S$, such that

(i) $g(S) \leq g(y), \forall y$ (i.e., $S$ is a global minimum).
(ii) $g(S) + K = g(s) < g(y), \forall y < s$.

(iii) $g(y)$ is decreasing on $(-\infty, s)$.

(iv) $g(y) \leq g(z) + K, \forall y, z, with s \leq y \leq z$.

Using part (d) of Lemma 3.1.1, we will show that the optimal policy is of the form

$$
\mu^*_k(x_k) = \begin{cases} 
S_k - x_k & \text{if } x_k < s_k \\
0 & \text{if } x_k \geq s_k
\end{cases}
$$

where $S_k$ is the value of $y$ that minimizes $G_k(y)$, and $s_k$ is the smallest value of $y$ for which $G_k(y) = K + G_k(S_k)$. This control policy is called the $(s, S)$ multiperiod policy.
Proof of the optimality of the \((s, S)\) multiperiod policy

For stage \(N - 1\),

\[
G_{N-1}(y) = cy + H(y) + E_w[J_N(y - w)]
\]

Therefore, \(G_{N-1}(y)\) is clearly convex \(\Rightarrow\) it is \(K\)-convex. Then, we have

\[
J_{N-1}(x) = \min \left\{ G_{N-1}(x), \min_{y > x} \{ K + cy + G_{N-1}(y) \} \right\} - cx,
\]

where by defining \(S_{N-1}\) as the minimizer of \(G_{N-1}(y)\) and \(s_{N-1} = \min\{y : G_{N-1}(y) = K + G_{N-1}(S_{N-1})\}\) (see Figure 3.1.9), we have the optimal control

\[
\mu^*_{N-1}(x_{N-1}) = \begin{cases} 
S_{N-1} - x_{N-1} & \text{if } x_{N-1} < s_{N-1} \\
0 & \text{if } x_{N-1} \geq s_{N-1},
\end{cases}
\]

which leads to the optimal value function

\[
J_{N-1}(x) = \begin{cases} 
K + G_{N-1}(S_{N-1}) - cx & \text{for } x < s_{N-1} \\
G_{N-1}(x) - cx & \text{for } x \geq s_{N-1}.
\end{cases}
\]

(3.1.7)

Observations:

- \(s_{N-1} \neq S_{N-1}\), because \(K > 0\)
- \(\frac{\partial}{\partial y} G_{N-1}(s_{N-1}) \leq 0\)

It turns out that the left derivative of \(J_{N-1}(\cdot)\) at \(s_{N-1}\) is greater than the right derivative \(\Rightarrow\) \(J_{N-1}(\cdot)\) is not convex (again, see Figure 3.1.9). Here, as we saw for the zero fixed ordering cost, the minimum

![Figure 3.1.9: Structure of the cost-to-go function when fixed cost is nonzero.](image-url)
$H^*$ occurs to the right of $S_{N-1}$ (recall that $S_{N-1}$ is the unconstrained minimizer of $G_{N-1}(x)$). To see this, note that

$$\frac{\partial}{\partial y} G_{N-1}(y) = c + \frac{\partial}{\partial y} H(y) \Rightarrow$$

$$\frac{\partial}{\partial y} G_{N-1}(S_{N-1}) = c + \frac{\partial}{\partial y} H(S_{N-1}) = 0 \Rightarrow$$

$$\frac{\partial}{\partial y} H(S_{N-1}) = -c < 0,$$

meaning that $H$ is decreasing at $S_{N-1}$, and so its minimum $H^*$ occurs to the right of $S_{N-1}$.

**Claim:** $J_{N-1}(x)$ is $K$-convex.

**Proof:** We must verify for all $z \geq 0, b > 0$, and $y$, that

$$K + J_{N-1}(y + z) \geq J_{N-1}(y) + z \left( \frac{J_{N-1}(y) - J_{N-1}(y - b)}{b} \right) \quad (3.1.8)$$

There are three cases according to the relative position of $y, y + z,$ and $s_{N-1}$.

Case 1: $y \geq s_{N-1}$ (i.e., $y + z \geq y \geq s_{N-1}$).

- If $y - b \geq s_{N-1} \Rightarrow J_{N-1}(x) = \frac{G_{N-1}(x)}{\text{convex} \Rightarrow K\text{-convex}} - \frac{cx}{\text{linear}}$, so by part (b) of Lemma 3.1.1, it is $K$-convex.
- If $y - b < s_{N-1} \Rightarrow$ in view of equation (3.1.7) we can write (3.1.8) as

$$K + J_{N-1}(y + z) = K + G_{N-1}(y + z) - c(y + z)$$

$$\geq G_{N-1}(y) - cy + z \left( \frac{J_{N-1}(y) - J_{N-1}(y - b)}{b} \right) + c(y - b) - \frac{G_{N-1}(s_{N-1})}{b} \cdot \frac{G_{N-1}(y) - G_{N-1}(s_{N-1})}{b},$$

or equivalently,

$$K + G_{N-1}(y + z) \geq G_{N-1}(y) + z \left( \frac{G_{N-1}(y) - G_{N-1}(s_{N-1})}{b} \right) \quad (3.1.9)$$

There are three subcases:

(i) If $y$ is such that $G_{N-1}(y) \geq G_{N-1}(s_{N-1}), y \neq s_{N-1}$ \Rightarrow by the $K$-convexity of $G_{N-1}$, and taking $y - s_{N-1}$ as the constant $b > 0$,

$$K + G_{N-1}(y + z) \geq G_{N-1}(y) + z \left( \frac{y - (y - s_{N-1})}{y - s_{N-1}} \right).$$

Thus, $K$-convexity hold.
(ii) If $y$ is such that $G_{N-1}(y) < G_{N-1}(s_{N-1})$ ⇒ From part (d-i) in Lemma 3.1.1, for a scalar $y + z$,

$$K + G_{N-1}(y + z) \geq K + G_{N-1}(s_{N-1}) = G_{N-1}(s_{N-1}) \quad \text{(by definition of } s_{N-1})$$

$$> G_{N-1}(y) \quad \text{(by hypothesis of this case).}$$

$$\geq G_{N-1}(y) + z \left( \frac{G_{N-1}(y) - G_{N-1}(s_{N-1})}{b} \right),$$

and equation (3.1.9) holds.

(iii) If $y = s_{N-1}$, then by $K$-convexity of $G_{N-1}$, note that (3.1.9) becomes

$$K + G_{N-1}(y + z) \geq G_{N-1}(y) + z \left( \frac{G_{N-1}(y) - G_{N-1}(s_{N-1})}{b} \right)$$

$$= G_{N-1}(y).$$

From Lemma 3.1.1, part (d-iv), taking $y = s_{N-1}$ there,

$$K + G_{N-1}(s_{N-1} + z) \geq G_{N-1}(s_{N-1})$$

is verified, for all $z \geq 0$.

Case 2: $y \leq y + z \leq s_{N-1}$.

By equation (3.1.7), the function $J_{N-1}(y)$ is linear ⇒ It is $K$-convex.

Case 3: $y < s_{N-1} < y + z$.

Here, we can write (3.1.8) as

$$K + J_{N-1}(y + z) = K + G_{N-1}(y + z) - c(y + z)$$

$$\geq J_{N-1}(y) + z \left( \frac{J_{N-1}(y) - J_{N-1}(y - b)}{b} \right)$$

$$= K + G_{N-1}(s_{N-1}) - cy + z \left( \frac{K + G_{N-1}(s_{N-1}) - cy - (K + G_{N-1}(s_{N-1}) - c(y - b))}{b} \right)$$

$$= K + G_{N-1}(s_{N-1}) - cy - czb/b$$

$$= K + G_{N-1}(s_{N-1}) - c(y + z)$$

Thus, the previous sequence of relations holds if and only if

$$K + G_{N-1}(y + z) - c(y + z) \geq K + G_{N-1}(s_{N-1}) - c(y + z),$$

or equivalently, if and only if $G_{N-1}(y + z) \geq G_{N-1}(s_{N-1})$, which holds from Lemma 3.1.1, part (d-i), since $G_{N-1}(\cdot)$ is $K$-convex.
This completes the proof of the claim.

We have thus proved that $K$-convexity and continuity of $G_{N-1}$, together with the fact that $G_{N-1}(y) \to \infty$ as $|y| \to \infty$, imply $K$-convexity of $J_{N-1}$. In addition, $J_{N-1}(x)$ can be seen to be continuous in $x$. Using the following facts:

- From the definition of $G_k(y)$:
  \[
  G_{N-2}(y) = cy + H(y) + E_w[ J_{N-1}(y - w) ]
  \]
  $K$-convex from Lemma 3.1.1-(c)
  $K$-convex from Lemma 3.1.1-(b)

- $G_{N-2}(y)$ is continuous (because of boundedness of $w_{N-2}$).
- $G_{N-2}(y) \to \infty$ as $|y| \to \infty$.

and repeating the preceding argument, we obtain that $J_{N-2}$ is $K$-convex, and proceeding similarly, we prove $K$-convexity and continuity of the functions $G_k$ for all $k$, as well as that $G_k(y) \to \infty$ as $|y| \to \infty$. At the same time, by using Lemma 3.1.1-(d), we prove optimality of the multiperiod $(s, S)$ policy.

Finally, it is worth noting that it is not necessary that $G_k(\cdot)$ be $K$-convex for an $(s, S)$ policy to be optimal; it is just a sufficient condition.

### 3.1.4 Exercises

**Exercise 3.1.1** Consider an inventory problem similar to the one discussed in class, with zero fixed cost. The only difference is that at the beginning of each period $k$ the decision maker, in addition to knowing the current inventory level $x_k$, receives an accurate forecast that the demand $w_k$ will be selected in accordance with one out of two possible probability distributions $P_l, P_s$ (large demand, small demand). The a priori probability of a large demand forecast is known.

(a) Obtain the optimal ordering policy for the case of a single-period problem

(b) Extend the result to the $N$-period case

**Exercise 3.1.2** Consider the inventory problem with nonzero fixed cost, but with the difference that demand is deterministic and must be met at each time period (i.e., the shortage cost per unit is $\infty$). Show that it is optimal to order a positive amount at period $k$ if and only if the stock $x_k$ is insufficient to meet the demand $w_k$. Furthermore, when a positive amount is ordered, it should bring up stock to a level that will satisfy demand for an integral number of periods.

**Exercise 3.1.3** Consider a problem of expanding over $N$ time periods the capacity of a production facility. Let us denote by $x_k$ the production capacity at the beginning of period $k$, and by $u_k \geq 0$ the addition to capacity during the $k$th period. Thus, capacity evolves according to

\[
  x_{k+1} = x_k + u_k, \quad k = 0, 1, \ldots, N - 1.
\]
The demand at the \( k \)th period is denoted \( w_k \) and has a known probability distribution that does not depend on either \( x_k \) or \( u_k \). Also, successive demands are assumed to be independent and bounded. We denote:

- \( C_k(u_k) \): Expansion cost associated with adding capacity \( u_k \).
- \( P_k(x_k + u_k - w_k) \): Penalty associated with capacity \( x_k + u_k \) and demand \( w_k \).
- \( S(x_N) \): Salvage value of final capacity \( x_N \).

Thus, the cost function has the form

\[
E_{w_0, \ldots, w_{N-1}} \left[ -S(x_N) + \sum_{k=0}^{N-1} (C_k(u_k) + P_k(x_k + u_k - w_k)) \right].
\]

(a) Derive the DP algorithm for this problem.

(b) Assume that \( S \) is a concave function with \( \lim_{x \to \infty} \frac{dS(x)}{dx} = 0 \), \( P_k \) are convex functions, and the expansion cost \( C_k \) is of the form

\[
C_k(u) = \begin{cases} 
K + c_k u & \text{if } u > 0, \\
0 & \text{if } u = 0,
\end{cases}
\]

where \( K \geq 0, c_k > 0 \) for all \( k \). Show that the optimal policy is of the \( (s, S) \) type assuming

\[
c_k y + E[P_k(y - w_k)] \to \infty \quad \text{as } |y| \to \infty.
\]

### 3.2 Single-Leg Revenue Management

Revenue Management (RM) is an OR subfield that deals with business related problems where there are finite, perishable capacities that must be depleted by a due time. Applications span from airlines, hospitality industry, and car rental, to more recent practices in retailing and media advertising. The problem sketched below is the basic RM problem that consists of rationing the capacity of a single resource through imposing limits on the quantities to be sold at different prices, for a given set of prices.

Setting:

- Initial capacity \( C \); remaining capacity denoted by \( x \).
- There are \( n \) customers classes labeled such that \( p_1 > p_2 > \cdots > p_n \).
- Time indices run backwards in time.
- Class \( n \) arrives first, followed by classes \( n-1, n-2, \ldots, 1 \).
- Demands are r.v: \( D_n, D_{n-1}, \ldots, D_1 \).
- At the beginning of stage \( j \), demands \( D_j, D_{j-1}, \ldots, D_1 \).
- Within stage \( j \) the model assumes the following sequence of events:
1. The realization of the demand $D_j$ occurs, and we observe the value.$^4$

2. We decide on a quantity $u$ to accept: $u \leq \min\{D_j, x\}$. The optimal control then is a function of both current demand and remaining capacity: $u^*(D_j, x)$. This is done for analytical convenience. In practice, the control decision has to be made before observing $D_j$. We will see that the calculation of the optimal control does not use information about $D_j$, so this assumption vanishes ex-post.

3. Revenue $p_j u$ is collected, and we proceed to stage $j - 1$ (since indices run backwards).

For the single-leg RM problem, the DP formulation becomes

$$V_j(x) = E_{D_j} \left[ \max_{0 \leq u \leq \min\{D_j, x\}} \{p_j u + V_{j-1}(x - u)\} \right], \quad (3.2.1)$$

with boundary conditions: $V_0(x) = 0, \ x = 0, 1, \ldots, C$. Note that in this formulation we have inverted the usual order between $\max\{\cdot\}$ and $E[\cdot]$. We prove below that this is w.l.o.g. for the kind of setting that we are dealing with here.

### 3.2.1 System with observable disturbances

Departure from standard DP: We can base our control $u_t$ on perfect knowledge of the random noise of the current period, $w_t$. For this section, assume as in the basic DP setting that indices run forward.

**Claim:** Assume a discrete finite horizon $t = 1, 2, \ldots, T$. The formulation

$$V_t(x) = \max_{u(x, w_t) \in U_t(x, w_t)} E_{w_t} [g_t(x, u(x, w_t), w_t) + V_{t+1}(f_t(x, u(x, w_t), w_t))]$$

is equivalent to

$$V_t(x) = E_{w_t} \left[ \max_{u(x) \in U_t(x)} g_t(x, u, w_t) + V_{t+1}(f_t(x, u, w_t)) \right], \quad (3.2.2)$$

which is more convenient to handle.

**Proof:** State space augmentation argument:

1. Reindex disturbances by defining $\tilde{w}_t = w_{t+1}, \ t = 1, \ldots, T - 1$.

2. Augment state to include the new disturbance, and define the system equation:

   $$\begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = \begin{pmatrix} f_t(x_t, u_t, w_t) \\ \tilde{w}_t \end{pmatrix}. $$

3. Starting from $(x_0, y_0) = (x, w_1)$, the standard DP recursion is:

   $$V_t(x, y) = \max_{u(x) \in U_t(x)} E_{\tilde{w}_t} [g_t(x, u, y) + V_{t+1}(f_t(x, u, y, \tilde{w}_t))].$$

$^4$Note that this is a departure from the basic DP formulation where typically we make a decision in period $k$ before the random noise $w_k$ is realized.
4. Define $G_t(x) = \mathbb{E}_{\tilde{w}_t}[V_t(x, \tilde{w}_t)]$

5. Note that:

$$V_t(x, y) = \max_{u(x) \in U_t(x)} \mathbb{E}_{\tilde{w}_t} [g_t(x, u, y) + V_{t+1}(f_t(x, u, y), \tilde{w}_t)]$$

$$= \max_{u(x) \in U_t(x)} \{g_t(x, u, y) + \mathbb{E}_{\tilde{w}_t} [V_{t+1}(f_t(x, u, y), \tilde{w}_t)]\} \quad \text{(because } g_t(\cdot) \text{ does not depend on } \tilde{w}_t)$$

$$= \max_{u(x) \in U_t(x)} \{g_t(x, u, y) + G_{t+1}(f_t(x, u, y))\} \quad \text{(by definition of } G_{t+1}(\cdot))$$

6. Replace $y$ by $w_t$ and take expectation with respect to $w_t$ on both sides above to obtain:

$$\mathbb{E}_{w_t}[V_t(x, w_t)] = \max_{u(x) \in U_t(x)} \{g_t(x, u, w_t) + G_{t+1}(f_t(x, u, w_t))\}$$

Observe that the LHS is indeed $G_t(x)$ modulus a small issue with the name of the random variable, which is justified by noting that $G_t(x) \triangleq \mathbb{E}_{\tilde{w}_t}[V_t(x, \tilde{w}_t)] = \mathbb{E}_w[V_t(x, w)]$. Finally, there is another minor “name issue”, because the final DP is expressed in terms of the value function $G$. It remains to replace $G$ by $V$ to recover formulation (3.2.2).

In words, what we are doing is anticipating and solving today the problem that we will face tomorrow, given the disturbances of today. The implicit sequence of actions of this alternative formulation is the following:

1. We observe current state $x$.
2. The value of the disturbance $w_t$ is realized.
3. We make the optimal decision $u^*(x, w_t)$.
4. We collect the current period reward $g_t(x, u(x, w_t), w_t)$.
5. We move to the next state $t + 1$.

### 3.2.2 Structure of the value function

Now, we turn to our original RM problem. First, we define the marginal value of capacity,

$$\Delta V_j(x) = V_j(x) - V_j(x-1), \quad \text{(3.2.3)}$$

and proceed to characterize the structure of the value function.

**Proposition 3.2.1** The marginal value of capacity $\Delta V_j(x)$ satisfies:

(i) For a fixed $j$, $\Delta V_j(x + 1) \leq \Delta V_j(x)$, $x = 0, \ldots, C$.

(ii) For a fixed $x$, $\Delta V_{j+1}(x) \geq \Delta V_j(x)$, $j = 1, \ldots, n$.

The proposition states two intuitive economic properties:
(i) For a given period, the marginal value of capacity is decreasing in the number of units left.

(ii) The marginal value of capacity \( x \) at stage \( j \) is smaller than its value at stage \( j + 1 \) (recall that indices are running backwards). Intuitively, this is because there are less periods remaining, and hence less opportunities to sell the \( x \)th unit.

Before going over the proof of this proposition, we need the following auxiliary lemma:

**Lemma 3.2.1** Suppose \( g : \mathbb{Z}_+ \to \mathbb{R} \) is concave. Let \( f : \mathbb{Z}_+ \to \mathbb{R} \) be defined by:

\[
f(x) = \max_{a=1, \ldots, m} \{ ap + g(x - a) \}
\]

for any given \( p \geq 0 \); and nonnegative integer \( m \leq x \). Then \( f(x) \) is concave in \( x \) as well.

**Proof:** We proceed in three steps:

1. Change of variable: Define \( y = x - a \), so that we can write:

\[
f(x) = \hat{f}(x) + px; \quad \text{where} \quad \hat{f}(x) = \max_{x-m \leq y \leq x} \{-yp + g(y)\}
\]

With this change of variable, we have that \( a = x - y \) and hence the inner part can be written as: \( (x - y)p + g(y) \), where the range for the argument is such that \( 0 \leq x - y \leq m \), or \( x - m \leq y \leq x \). The new function is

\[
f(x) = \max_{x-m \leq y \leq x} \{(x - y)p + g(y)\}.
\]

Thus, \( f(x) = \hat{f}(x) + px \), where

\[
\hat{f}(x) = \max_{x-m \leq y \leq x} \{-yp + g(y)\}
\]

Note that since \( x \geq m \), then \( y \geq 0 \).

2. Closed-form for \( \hat{f}(x) \): Let \( h(y) = -yp + g(y) \), for \( y \geq 0 \). Let \( y^* \) be the unconstrained maximizer of \( h(y) \), i.e. \( y^* = \arg\max_{y \geq 0} h(y) \). Because of the shape of \( h(y) \), this maximizer is always well defined. Moreover, since \( g(y) \) is concave, \( h(y) \) is also concave, nondecreasing for \( y \leq y^* \), and nonincreasing for \( y > y^* \). Therefore, for given \( m \) and \( p \):

\[
\hat{f}(x) = \begin{cases} 
  -xp + g(x) & \text{if } x \leq y^* \\
  y^*p + g(y^*) & \text{if } y^* \leq x \leq y^* + m \\
  -(x - m)p + g(x - m) & \text{if } x \geq y^* + m
\end{cases}
\]

The first part holds because \( h(y) \) is nondecreasing for \( 0 \leq y \leq y^* \). The second part holds because \( \hat{f}(x) = -y^*p + g(y^*) = h(y^*) \), for \( y^* \) in the range \( \{x-m, \ldots, x\} \), or equivalently, for \( y^* \leq x \leq y^* + m \). Finally, since \( h(y) \) is nonincreasing for \( y > y^* \), the maximum is attained in the border of the range, i.e., in \( x - m \).

3. Concavity of \( \hat{f}(x) \):
• Take $x < y^*$. We have
\[
\hat{f}(x + 1) - \hat{f}(x) = [-g(x + 1) + p(x + 1)] - [-p + g(x)]
\leq -p + g(x + 1) - g(x) \quad \text{(because $g(x)$ is concave)}
\]
\[
= \hat{f}(x) - \hat{f}(x - 1).
\]

So, for $x < y^*$, $\hat{f}(x)$ is concave.

• Take $y^* \leq x < y^* + m$. Here, $\hat{f}(x + 1) - \hat{f}(x) = 0$, and so $\hat{f}(x)$ is trivially concave.

• Take $x \geq y^* + m$. We have
\[
\hat{f}(x + 1) - \hat{f}(x) = [-g(x + 1 - m) + p(x + 1 - m)] - [-g(x - m) + p(x - m)]
\leq -p + g(x - m) - g(x - m - 1) \quad \text{(because $g(x)$ is concave)}
\]
\[
= \hat{f}(x) - \hat{f}(x - 1).
\]

So, for $x \geq y^* + m$, $\hat{f}(x)$ is concave.

Therefore $\hat{f}(x)$ is concave for all $x \geq 0$, and since $f(x) = \hat{f}(x) + px$, $f(x)$ is concave in $x \geq 0$ as well. □

**Proof of Proposition 3.2.1**

**Part (i):** $\Delta V_j(x + 1) \leq \Delta V_j(x)$, $\forall x$.

By induction:

• In terminal stage: $V_0(x) = 0$, $\forall x$, so it holds.

• IH: Assume that $V_{j-1}(x)$ is concave in $x$.

• Consider $V_j(x)$. Note that
\[
V_j(x) = \mathbb{E}_{D_j} \left[ \max_{0 \leq u \leq \min\{D_j, x\}} \{p_j u + V_{j-1}(x - u)\} \right].
\]

For any realization of $D_j$, the function
\[
H(x, D_j) = \max_{0 \leq u \leq \min\{D_j, x\}} \{p_j u + V_{j-1}(x - u)\}
\]
has exactly the same structure as the function of the Lemma above, with $m = \min\{D_j, x\}$, and therefore it is concave in $x$. Since $\mathbb{E}_{D_j}[H(x, D_j)]$ is a weighted average of concave functions, it is also concave. □

Going back to the original formulation for the single-leg RM problem in (3.2.1), we can express it as follows:
\[
V_j(x) = \mathbb{E}_{D_j} \left[ \max_{0 \leq u \leq \min\{D_j, x\}} \{p_j u + V_{j-1}(x - u)\} \right]
\]
\[
= V_{j-1}(x) + \mathbb{E}_{D_j} \left[ \max_{0 \leq u \leq \min\{D_j, x\}} \left\{ \sum_{z=1}^{u} (p_j - \Delta V_{j-1}(x + 1 - z)) \right\} \right], \quad (3.2.4)
\]
where we are using (3.2.3) to write $V_{j-1}(x - u)$ as a sum of increments:

$$V_{j-1}(x - u) = V_{j-1}(x) - \sum_{z=1}^{u} V_{j-1}(x + 1 - z)$$

$$= V_{j-1}(x) - [\Delta V_{j-1}(x) + \Delta V_{j-1}(x - 1) + \cdots +$$

$$+ \Delta V_{j-1}(x + 1 - (u - 1)) + \Delta V_{j-1}(x + 1 - u)]$$

$$= V_{j-1}(x) - [V_{j-1}(x) - V_{j-1}(x - 1) + V_{j-1}(x - 1) - V_{j-1}(x - 2) + \cdots +$$

$$+ V_{j-1}(x + 2 - u) - V_{j-1}(x + 1 - u) + V_{j-1}(x + 1 - u) - V_{j-1}(x - u)]$$

Note that all terms in the RHS except for the last one cancel out. The inner sum in (3.2.4) is defined to be zero when $u = 0$.

**Part (ii):** $\Delta V_{j+1}(x) \geq \Delta V_{j}(x)$, $\forall j$.

From (3.2.4) we can write:

$$V_{j+1}(x) = V_{j}(x) + E_{D_{j+1}} \left[ \max_{0 \leq u \leq \min\{D_{j+1}, x\}} \left\{ \sum_{z=1}^{u} (p_{j+1} - \Delta V_{j}(x + 1 - z)) \right\} \right].$$

Similarly, we can write:

$$V_{j+1}(x - 1) = V_{j}(x - 1) + E_{D_{j+1}} \left[ \max_{0 \leq u \leq \min\{D_{j+1}, x - 1\}} \left\{ \sum_{z=1}^{u} (p_{j+1} - \Delta V_{j}(x - z)) \right\} \right].$$

Subtracting both equalities, we get:

$$\Delta V_{j+1}(x) = \Delta V_{j}(x) + E_{D_{j+1}} \left[ \max_{0 \leq u \leq \min\{D_{j+1}, x\}} \left\{ \sum_{z=1}^{u} (p_{j+1} - \Delta V_{j}(x + 1 - z)) \right\} \right]$$

$$- E_{D_{j+1}} \left[ \max_{0 \leq u \leq \min\{D_{j+1}, x - 1\}} \left\{ \sum_{z=1}^{u} (p_{j+1} - \Delta V_{j}(x - z)) \right\} \right]$$

$$\geq \Delta V_{j}(x) + E_{D_{j+1}} \left[ \max_{0 \leq u \leq \min\{D_{j+1}, x\}} \left\{ \sum_{z=1}^{u} (p_{j+1} - \Delta V_{j}(x - z)) \right\} \right]$$

$$- E_{D_{j+1}} \left[ \max_{0 \leq u \leq \min\{D_{j+1}, x - 1\}} \left\{ \sum_{z=1}^{u} (p_{j+1} - \Delta V_{j}(x - z)) \right\} \right]$$

$$\geq \Delta V_{j}(x) + E_{D_{j+1}} \left[ \max_{0 \leq u \leq \min\{D_{j+1}, x - 1\}} \left\{ \sum_{z=1}^{u} (p_{j+1} - \Delta V_{j}(x - z)) \right\} \right]$$

$$- E_{D_{j+1}} \left[ \max_{0 \leq u \leq \min\{D_{j+1}, x - 1\}} \left\{ \sum_{z=1}^{u} (p_{j+1} - \Delta V_{j}(x - z)) \right\} \right]$$

$$= \Delta V_{j}(x),$$

where the first inequality holds from part (i) in Proposition 3.2.1, and the second one holds because the domain of $u$ in the maximization problem of the first expectation (in the second to last line) is smaller, and hence it is a more constrained optimization problem. 

---

119
3.2.3 Structure of the optimal policy

The good feature of formulation (3.2.4) is that it is very insightful about the structure of the optimal policy. In particular, from part (i) of Proposition 3.2.1, since $\Delta V_j(x)$ is decreasing in $x$, $p_j - \Delta V_{j-1}(x + 1 - z)$ is decreasing in $z$. So, it is optimal to keep adding terms to the sum (i.e., increase $u$) as long as

$$p_j - \Delta V_{j-1}(x + 1 - u) \geq 0,$$

or the upper bound $\min\{D_j, x\}$ is reached, whichever comes first. In words, we compare the instantaneous revenue $p_j$ with the marginal value of capacity (i.e., the value of a unit if we keep it for the next period). If the former dominates, then we accept the price $p_j$ for the unit.

The resulting optimal controls can be expressed in terms of optimal protection levels $y_j^*$, for classes $j, j-1, \ldots, 1$ (i.e., class $j$ and higher in the revenue order). Specifically, we define

$$y_j^* = \max\{x : 0 \leq x \leq C, \ p_{j+1} < \Delta V_j(x)\}, \quad j = 1, 2, \ldots, n - 1,$$

and we assume $y_0^* = 0$ and $y_n^* = C$. Figure 3.2.1 illustrates the determination of $y_j^*$. For $x \leq y_j^*$, $p_{j+1} \geq \Delta V_j(x)$, and therefore it is worth waiting for the demand to come rather than selling now.

![Figure 3.2.1: Calculation of the optimal protection level $y_j^*$](image)

The optimal control at stage $j + 1$ is then

$$\mu_j^*(x, D_{j+1}) = \min\{(x - y_j^*)^+, D_{j+1}\}$$

The key observation here is that the computation of $y_j^*$ does not depend on $D_{j+1}$, because the knowledge of $D_{j+1}$ does not affect the future value of capacity. Therefore, going back to the assumption we made at the beginning, assuming that we know demand $D_{j+1}$ to compute $y_j^*$ does not really matter, because we do not make real use of that information.

Part (ii) in Proposition 3.2.1 implies the nested protection structure

$$y_1^* \leq y_2^* \leq \cdots \leq y_{n-1}^* \leq y_n^* = C.$$

This is illustrated in Figure 3.2.2. The reason is that since the curve $\Delta V_{j-1}(x)$ is below the curve $\Delta V_j(x)$ pointwise, and since by definition, $p_j > p_{j+1}$, then $y_{j-1}^* \leq y_j^*$. 

---

120
3.2.4 Computational complexity

Using the optimal control, the single-leg RM problem (3.2.1) could be reformulated as

\[ V_j(x) = E_{D_j} \left[ p_j \min\{(x - y^{*}_{j-1})^+, D_j\} + V_{j-1}(x - \min\{(x - y^{*}_{j-1})^+, D_j\}) \right] \quad (3.2.6) \]

This procedure is repeated starting from \( j = 1 \) and working backward to \( j = n \).

- For discrete-demand distributions, computing the expectation in (3.2.6) for each state \( x \) requires evaluating at most \( O(C) \) terms since \( \min\{(x - y^{*}_{j-1})^+, D_j\} \leq C \). Since there are \( C \) states (capacity levels), the complexity at each stage is \( O(C^2) \).

- The critical values \( y^{*}_j \) can then be identified from (3.2.5) in \( \log(C) \) time by binary search as \( \Delta V_j(x) \) is nonincreasing. In fact, since we know \( y^{*}_j \geq y^{*}_{j-1} \), the binary search can be further constrained to values in the interval \([y^{*}_{j-1}, C]\). Therefore, computing \( y^{*}_j \) does not add to the complexity at stage \( j \).

- These steps must be repeated for each of the \( n - 1 \) stages, giving a total complexity of \( O(nC^2) \).

3.2.5 Airlines: Practical implementation

Airlines that use capacity control as their RM strategy (as opposed to dynamic pricing) post protection levels \( y^{*}_j \) in their own reservation systems, and accept requests for product \( j + 1 \) until \( y^{*}_j \) is reached or stage \( j + 1 \) ends (whichever comes first). Figure 3.2.3 is a snapshot from Expedia.com showing this practice from American Airlines.

3.2.6 Exercises

Exercise 3.2.1 Single-leg Revenue Management problem: For a single leg RM problem assume that:

- There are \( n = 10 \) classes.
Demand $D_j$ is calculated through discretizing a truncated normal with mean $\mu = 10$ and standard deviation $\sigma = 2$, on support $[0, 20]$. Specifically, take:

$$P(D_j = k) = \Phi\left(\frac{(k + 0.5 - 10)/2}{\Phi(20.5)/2} - \Phi((-0.5 - 10)/2)\right), \quad k = 0, \ldots, 20$$

Note that this discretization and re-scaling verifies: $\sum_{k=0}^{20} P(D_j = k) = 1$.

• Total capacity available is $C = 100$.

• Prices are $p_1 = 500, p_2 = 480, p_3 = 465, p_4 = 420, p_5 = 400, p_6 = 350, p_7 = 320, p_8 = 270, p_9 = 250$, and $p_{10} = 200$.

Write a MATLAB or C code to compute optimal protection levels $y_1^*, \ldots, y_9^*$; and find the total expected revenue $V_{10}(100)$. Note that you can take advantage of the structure of the optimal policy to simplify its computation. Submit your results, and a copy of the code.

**Exercise 3.2.2 Heuristic for the single-leg RM problem:** In the airline industry, the single-leg RM problem is typically solved using a heuristic; the so-called EMSR-b (expected marginal seat revenue - version b). There is no much reason for this other than the tradition of its usage, and the fact that it provides consistently good results. Here is a description:

Consider stage $j + 1$ in which we want to determine protection level $y_j$. Define the aggregated future demand for classes $j, j-1, \ldots, 1$, by $S_j = \sum_{k=1}^{j} D_k$, and let the weighted-average revenue
from classes 1, ..., j, denoted $\bar{p}_j$, be defined by

$$\bar{p}_j = \frac{\sum_{k=1}^{j} p_k \mathbb{E}[D_k]}{\sum_{k=1}^{n} \mathbb{E}[D_k]}.$$ 

Then the EMSR-b protection level for class $j$ and higher, $y_j$, is chosen by

$$\mathbb{P}(S_j > y_j) = \frac{p_{j+1}}{\bar{p}_j}.$$ 

It is common when using EMSR-b to assume demand for each class $j$ is independent and normally distributed with mean $\mu_j$ and variance $\sigma^2_j$, in which case

$$y_j = \mu + z_\alpha \sigma,$$

where $\mu = \sum_{k=1}^{j} \mu_k$ is the mean and $\sigma^2 = \sum_{k=1}^{j} \sigma^2_k$ is the variance of the aggregated demand to come at stage $j + 1$, and

$$z_\alpha = \Phi^{-1}(1 - p_{j+1}/\bar{p}_j).$$

Apply this heuristic to compute protection levels $y_1, ..., y_9$ using the data of the previous exercise and assuming that demand is normal (no truncation, no discretization), and compare the outcome with the optimal protection levels computed before.

### 3.3 Optimal Stopping and Scheduling Problems

In this section, we focus on two other types of problems with perfect state information: optimal stopping problems (mainly) and discuss few ideas on scheduling problems.

#### 3.3.1 Optimal stopping problems

We assume the following:

- At each state, there is a control available that stops the system.
- At each stage, you observe the current state and decide either to stop or continue.
- Each policy consists of a partition of the set of states $x_k$ into two regions: the stop region and the continue region. Figure 3.3.1 illustrates this.
- Domain of states remains the same throughout the process.

**Application: Asset selling problem**

- Consider a person owning an asset for which she is offered an amount of money from period to period, across $N$ periods.
- Offers are random and independent, denoted $w_0, w_1, ...w_{N-1}$, with $w_i \in [0, \bar{w}]$. 

Figure 3.3.1: Each policy consists of a partition of the state space into the stop and the continue regions.

- If the seller accepts an offer, she can invest the money at a fixed rate $r > 0$.
  Otherwise, she waits until next period to consider the next offer.

- Assume that the last offer $w_{N-1}$ must be accepted if all prior offers are rejected.

- Objective: Find a policy for maximizing reward at the $N$th period.

Let’s solve this problem.

- Control:
  
  $\mu_k(x_k) = \begin{cases} 
  u_1 : \text{Sell} \\
  u_2 : \text{Wait} 
  \end{cases}$

- State: $x_k = \mathbb{R}_+ \cup \{T\}$.

- System equation:
  
  $x_{k+1} = \begin{cases} 
  T & \text{if } x_k = T, \text{ or } x_k \neq T \text{ and } \mu_k = u_1, \\
  w_k & \text{otherwise.} 
  \end{cases}$

- Reward function:
  
  $E_{w_0,...,w_{N-1}} \left[ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k, w_k) \right]$ 

  where

  $g_N(x_N) = \begin{cases} 
  x_N & \text{if } x_N \neq T, \\
  0 & \text{if } x_N = T 
  \end{cases}$ (i.e., the seller must accept the offer by time $N$),

  and for $k = 0, 1, \ldots, N-1$,

  $g_k(x_k, \mu_k, w_k) = \begin{cases} 
  (1 + r)^{N-k} x_k & \text{if } x_k \neq T \text{ and } \mu_k = u_1, \\
  0 & \text{otherwise.} 
  \end{cases}$

  Note that here, a critical issue is how to account for the reward, being careful with the double counting. In this formulation, once the seller accepts the offer, she gets the compound interest for the rest of the horizon all together, and from there onwards, she gets zero reward.
• DP formulation

\[
J_N(x_N) = \begin{cases} 
  x_N & \text{if } x_N \neq T, \\
  0 & \text{if } x_N = T. 
\end{cases} 
\]  

(3.3.1)

For \( k = 0, 1, \ldots, N - 1 \),

\[
J_k(x_k) = \begin{cases} 
  \max\{ (1 + r)^N x_k, E[J_{k+1}(w)] \} & \text{if } x_k \neq T, \\
  0 & \text{if } x_k = T. 
\end{cases} 
\]  

(3.3.2)

• Optimal policy: Accept offer only when

\[
x_k > \alpha_k \triangleq \frac{E[J_{k+1}(w)]}{(1 + r)^N}. 
\]

Note that \( \alpha_k \) represents the net present value of the expected reward. This comparison is a fair one, because it is conducted between the instantaneous payoff \( x_k \) and the expected reward discounted back to the present time \( k \). Thus, the optimal policy is of the threshold type, described by the scalar sequence \( \{\alpha_k : k = 0, \ldots, N - 1\} \). Figure 3.3.2 represents this threshold structure.

![Figure 3.3.2: Optimal policy of accepting/rejecting offers in the asset selling problem.](image)

**Proposition 3.3.1** Assume that offers \( w_k \) are i.i.d., with \( w \sim F(\cdot) \). Then, \( \alpha_k \geq \alpha_{k+1}, \ k = 1, \ldots, N - 1, \) with \( \alpha_N = 0 \).

**Proof:** For now, let’s disregard the terminal condition, and define

\[
V_k(x_k) \triangleq \frac{J_k(x_k)}{(1 + r)^N}, \quad x_k \neq T. 
\]

We can rewrite equations (3.3.1) and (3.3.2) as follows:

\[
V_N(x_N) = x_N, \\
V_k(x_k) = \max\{ x_k, (1 + r)^{-1}E_0[V_{k+1}(w)] \}, \quad k = 0, \ldots, N - 1. 
\]  

(3.3.3)
Hence, defining $\alpha_N = 0$ (since we have to accept no matter what in the last period), we get

$$\alpha_k = \frac{E_w[V_{k+1}(w)]}{1 + r}, \quad k = 0, 1, ..., N - 1.$$  

Next, we compare the value function at periods $N - 1$ and $N$: For $k = N$ and $k = N - 1$, we have

$$V_N(x) = x$$

$$V_{N-1}(x) = \max\{x, (1 + r)^{-1}E_w[V_N(w)]\} \geq V_N(x)$$

Given that we have a stationary system, from the monotonicity of DP (see Homework #2), we know that

$$V_1(x) \geq V_2(x) \geq \cdots \geq V_N(x), \quad \forall x.$$  

Since $\alpha_k = \frac{E_w[V_{k+1}(w)]}{1 + r}$ and $\alpha_{k+1} = \frac{E_w[V_{k+2}(w)]}{1 + r}$, we have $\alpha_k \geq \alpha_{k+1}$.

**Compute limiting $\alpha$**

Next, we explore the question: What if the selling horizon is very long? Note that equation (3.3.3) can be written as $V_k(x_k) = \max\{x_k, \alpha_k\}$, where

$$\alpha_k = (1 + r)^{-1}E_w[V_{k+1}(w)]$$

$$= \frac{1}{1 + r} \int_{\alpha_{k+1}}^{\alpha_{k+1}} \alpha_{k+1} dF(w) + \frac{1}{r + 1} \int_{\alpha_{k+1}}^{\infty} wdF(w)$$

$$= \frac{\alpha_{k+1}}{1 + r} F(\alpha_{k+1}) + \frac{1}{1 + r} \int_{\alpha_{k+1}}^{\infty} wdF(w)$$  \hspace{1cm} (3.3.4)

We will see that the sequence $\{\alpha_k\}$ converges as $k \to -\infty$ (i.e., as the selling horizon becomes very long).

Observations:

1. $0 \leq \frac{F(\alpha)}{1 + r} \leq \frac{1}{1 + r}$.

2. For $k = 0, 1, \ldots, N - 1$,

$$0 \leq \frac{1}{1 + r} \int_{\alpha_{k+1}}^{\infty} wdF(w) \leq \frac{1}{1 + r} \int_{0}^{\infty} wdF(w) = \frac{E[w]}{1 + r}.$$  

3. From equation (3.3.4) and Proposition 3.3.1:

$$\alpha_k \leq \frac{\alpha_{k+1}}{1 + r} + \frac{E[w]}{1 + r} \leq \alpha_k \frac{1}{1 + r} + \frac{E[w]}{1 + r} \Rightarrow \alpha_k < \frac{E[w]}{r}$$

Using $\alpha_k \geq \alpha_{k+1}$ and knowing that the sequence is bounded from above, we know that when $k \to -\infty$, $\alpha_k \to \bar{\alpha}$, where $\bar{\alpha}$ satisfies

$$(1 + r)\bar{\alpha} = F(\bar{\alpha})\bar{\alpha} + \int_{\bar{\alpha}}^{\infty} wdF(w)$$
When $N$ is "big", then an approximate method is to use the constant policy: Accept the offer $x_k$ if and only if $x_k > \bar{\alpha}$. More formally, if we define $G(\alpha)$ as

$$G(\alpha) \triangleq \frac{1}{r+1} \left( F(\alpha) \alpha + \int_{\alpha}^{\infty} w dF(w) \right),$$

then from the Contraction Mapping Theorem (due to Banach, 1922), $G(\alpha)$ is a contraction mapping, and hence the iterative procedure $\alpha_{n+1} = G(\alpha_n)$ finds the unique fixed point in $[0, E[w]/r]$, starting from any arbitrary $\alpha_0 \in [0, E[w]/r]$.

**Recall:** $G$ is a contraction mapping if for all $x, y \in \mathbb{R}^n$, $||G(x) - G(y)|| < K||x - y||$, for a constant $0 \leq K < 1$, $K$ independent of $x, y$.

**Application: Purchasing with a deadline**

- Assume that a certain quantity of raw material is needed at a certain time.
- Price of raw materials fluctuates
  Decision: Purchase or not?
  Objective: Minimum expected price of purchase
- Assume that successive prices $w_k$ are i.i.d. and have c.d.f. $F(\cdot)$.
- Purchase must be made within $N$ time periods.
- Controls:
  $$\mu_k(x_k) = \begin{cases} u_1 : \text{Purchase} \\ u_2 : \text{Wait} \end{cases}$$
- State: $x_k = \mathbb{R}_+ \cup \{T\}$.
- System equation:
  $$x_{k+1} = \begin{cases} T & \text{if } x_k = T, \text{ or } x_k \neq T \text{ and } \mu_k = u_1, \\ w_k & \text{otherwise.} \end{cases}$$
- DP formulation:
  $$J_N(x_N) = \begin{cases} x_N & \text{if } x_N \neq T, \\ 0 & \text{otherwise.} \end{cases}$$

For $k = 0, \ldots, N - 1$, 

$$J_k(x_k) = \begin{cases} \min\{ x_k, \text{E}[J_{k+1}(w_k)] \} & \text{if } x_k \neq T, \\ 0 & \text{if } x_k = T. \end{cases}$$

- Optimal policy: Purchase if and only if
  $$x_k < \alpha_k \triangleq \text{E}_w[J_{k+1}(w)],$$
where
\[
\alpha_k \overset{\triangle}{=} E_w[J_{k+1}(w)] = E_w[\min\{w, \alpha_{k+1}\}] = \int_0^{\alpha_{k+1}} wdF(w) + \int_{\alpha_{k+1}}^{\infty} \alpha_{k+1} dF(w).
\]

With terminal condition:
\[
\alpha_{N-1} = \int_0^{\infty} wdF(w) = E[w].
\]

Analogously to the asset selling problem, it must hold that
\[
\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_{N-1} = E[w].
\]

Intuitively, we are less stringent and willing to accept a higher price as time goes by.

The case of correlated prices

Suppose that prices evolve according to the system equation
\[
x_{k+1} = \lambda x_k + \xi_k, \quad \text{where } 0 < \lambda < 1,
\]
and where \(\xi_1, \xi_2, \ldots, \xi_{N-1}\) are i.i.d. with \(E[\xi] = \bar{\xi} > 0\).

DP Algorithm:
\[
J_N(x_N) = x_N
\]
\[
J_k(x_k) = \min\{x_k, E[J_{k+1}(x_k \lambda + \xi_k)]\}, \quad k = 0, \ldots, N-1.
\]

In particular, for \(k = N-1\), we have
\[
J_{N-1}(x_{N-1}) = \min\{x_{N-1}, E[J_N(x_{N-1} \lambda + \xi_{N-1})]\}
\]
\[
\overset{\triangle}{=} \min\{x_{N-1}, x_{N-1} \lambda + \bar{\xi}\}
\]

Optimal policy at time \(N-1\): Purchase only when \(x_{N-1} < \alpha_{N-1}\), where \(\alpha_{N-1}\) comes from
\[
x_{N-1} < \lambda x_{N-1} + \bar{\xi} \iff x_{N-1} < \alpha_{N-1} \overset{\triangle}{=} \frac{\bar{\xi}}{1 - \lambda}.
\]

In addition, we can see that
\[
J_{N-1}(x) = \min\{x, \lambda x + \bar{\xi}\} \leq x = J_N(x).
\]

Using the stationarity of the system and the monotonicity property of DP, we have that for any \(x\),
\(J_k(x) \leq J_{k+1}(x), k = 0, \ldots, N-1\).

Moreover, \(J_{N-1}(x)\) is concave and increasing in \(x\) (see Figure 3.3.3). By a backward induction argument, we can prove for \(k = 0, 1, \ldots, N-2\) that \(J_k(x)\) is concave and increasing in \(x\) (see Figure 3.3.4). These facts imply that the optimal policy for every period \(k\) is of the form: Purchase if and only if \(x_k < \alpha_k\), where the scalar \(\alpha_k\) is the unique positive solution of the equation
\[
x = E[J_{k+1}(\lambda x + \xi_k)].
\]

Notice that the relation \(J_k(x) \leq J_{k+1}(x)\) for all \(x\) and \(k\) implies that
\[
\alpha_k \leq \alpha_{k+1}, \quad k = 0, \ldots, N-2,
\]
and again (as one would expect) the threshold price to purchase increases as the deadline gets closer. In other words, one is more willing to accept a higher price as one approaches the end of the horizon. This is illustrated in Figure 3.3.5.
3.3.2 General stopping problems and the one-step look ahead policy

- Consider a stationary problem
- At time \( k \), we may stop at cost \( t(x_k) \) or choose a control \( \mu_k(x_k) \in U(x_k) \) and continue.
- The DP algorithm is given by:

\[
J_N(x_N) = t(x_N),
\]

and for \( k = 0, 1, \ldots, N - 1 \),

\[
J_k(x_k) = \min \left\{ t(x_k), \min_{u_k \in U(x_k)} \mathbb{E} \left[ g(x_k, u_k, w) + J_{k+1}(f(x_k, u_k, w)) \right] \right\},
\]

and it is optimal to stop at time \( k \) for states \( x \) in the set

\[
T_k = \left\{ x : t(x) \leq \min_{u \in U(x)} \mathbb{E} \left[ g(x, u) + J_{k+1}(f(x, u, w)) \right] \right\}.
\]
Figure 3.3.5: Structure of the value functions $J_k(x)$ and $J_{k+1}(x)$ when prices are correlated.

- Note that $J_{N-1}(x) \leq J_N(x), \forall x$. This holds because

\[
J_{N-1}(x) = \min \left\{ t(x), \min_{u \in U(x)} \mathbb{E} [g(x,u,u_{N-1},w) + J_N(f(x,u,u_{N-1},w))] \right\} \leq t(x) = J_N(x).
\]

Using the monotonicity of the DP, we have $J_k(x) \leq J_{k+1}(x), k = 0, 1, \ldots, N-1$. Since

\[
T_{k+1} = \left\{ x : t(x) \leq \min_{u \in U(x)} \mathbb{E} [g(x,u,w) + J_{k+2}(f(x,u,w))] \right\},
\]

and the RHS in (3.3.5) is less or equal than the RHS in (3.3.6), we have

\[
T_0 \subset T_1 \subset \cdots \subset T_k \subset T_{k+1} \subset \cdots \subset T_{N-1}.
\]

- Question: When are all stopping sets $T_k$ equal?

Answer: Suppose that the set $T_{N-1}$ is absorbing in the sense that if a state belongs to $T_{N-1}$ and termination is not selected, the next state will also be in $T_{N-1}$; that is,

\[
f(x,u,w) \in T_{N-1}, \ \forall x \in T_{N-1}, u \in U(x), \text{ and } w. \quad (3.3.8)
\]

By definition of $T_{N-1}$ we have

\[
J_{N-1}(x) = t(x), \text{ for all } x \in T_{N-1}.
\]

We obtain for $x \in T_{N-1},$

\[
\min_{u \in U(x)} \mathbb{E} [g(x,u,w) + J_{N-1}(f(x,u,w))] = \min_{u \in U(x)} \mathbb{E} [g(x,u,w) + t(f(x,u,w))]
\geq t(x) \quad \text{(because of (3.3.5) applied to } k = N - 1). \]

Since

\[
J_{N-2}(x) = \min \left\{ t(x), \min_{u \in U(x)} \mathbb{E} [g(x,u,w) + J_{N-1}(f(x,u,w))] \right\},
\]

then $x \in T_{N-2},$ or equivalently $T_{N-1} \subset T_{N-2}$. This, together with (3.3.7), implies $T_{N-1} = T_{N-2}$. Proceeding similarly, we obtain $T_k = T_{N-1}, \forall k.$
Conclusion: If condition (3.3.8) holds (i.e., the one-step stopping set $T_{N-1}$ is absorbing), then the stopping sets $T_k$ are all equal to the set of states for which it is better to stop rather than continue for one more stage and then stop. A policy of this type is known as a one-step-look-ahead policy. Such a policy turns out to be optimal in several types of applications.

Example 3.3.1 (Asset selling with past offers retained)

Take the previous asset selling problem in Section 3.3.1, and suppose now that rejected offers can be accepted at a later time. Then, if the asset is not sold at time $k$, the state evolves according to:

$$x_{k+1} = \max\{x_k, w_k\},$$

instead of just $x_{k+1} = w_k$. Note that this system equation retains the best offered got so far from period 0 to $k$.

The DP algorithm becomes:

$$V_N(x_N) = x_N,$$

and for $k = 0, 1, \ldots, N-1$,

$$V_k(x_k) = \max\{x_k, (1+r)^{-1}Ew_k[V_{k+1}(\max\{x_k, w_k\})]\}.$$

The one-step stopping set is:

$$T_{N-1} = \{x : x \geq (1+r)^{-1}Ew[\max\{x, w\}]\}.$$

Define $\bar{\alpha}$ as the $x$ that satisfies the equation

$$x = \frac{Ew[\max\{x, w\}]}{1+r};$$

so that $T_{N-1} = \{x : x \geq \bar{\alpha}\}$. Thus,

$$\bar{\alpha} = \frac{1}{1+r}Ew[\max\{\bar{\alpha}, w\}] = \frac{1}{1+r} \left( \int_0^{\bar{\alpha}} \bar{\alpha} dF(w) + \int_{\bar{\alpha}}^{\infty} wdF(w) \right) = \frac{1}{1+r} \left( \bar{\alpha} F(\bar{\alpha}) + \int_{\bar{\alpha}}^{\infty} wdF(w) \right),$$

or equivalently,

$$(1+r)\bar{\alpha} = \bar{\alpha} F(\bar{\alpha}) + \int_{\bar{\alpha}}^{\infty} wdF(w).$$

Since past offers can be accepted at a later date, the effective offer available cannot decrease with time, and it follows that the one-step stopping set

$$T_{N-1} = \{x : x \geq \bar{\alpha}\}$$

is absorbing in the sense of (3.3.8). In symbols, for $x \in T_{N-1}$, $f(x, u, w) = \max\{x, w\} \geq x \geq \bar{\alpha}$, and so $f(x, u, w) \in T_{N-1}$. Therefore, the one-step-look-ahead stopping rule that accepts the first offer that equals or exceeds $\bar{\alpha}$ is optimal. □
3.3.3 Scheduling problem

- Consider a given a set of tasks to perform, with the ordering subject to optimal choice.
- Costs depend on the order.
- There might be uncertainty, and precedence and resource availability constraints.
- Some problems can be solved efficiently by an interchange argument.

Example: Quiz problem

- Given a list of $N$ questions, if question $i$ is answered correctly (which occurs with probability $p_i$), we receive reward $R_i$; if not the quiz terminates.
- Let $i$ and $j$ be the $k$th and $(k+1)$st questions in an optimally ordered list $L = (i_0, i_1, \ldots, i_{k-1}, i, j, i_{k+2}, \ldots, i_{N-1})$.

We have

$$
E[\text{Reward}(L)] = E[\text{Reward}(i_0, \ldots, i_{k-1})] + p_{i_0} \cdots p_{i_k} (p_i R_i + p_i p_j R_j) + E[\text{Reward}(i_{k+2}, \ldots, i_{N-1})].
$$

Consider the list $L$, now with $i$ and $j$ interchanged, and let:

$$
L' = (i_0, \ldots, i_{k-1}, j, i, i_{k+2}, \ldots, i_{N-1}).
$$

Since $L$ is optimal,

$$
E[\text{Reward}(L)] \geq E[\text{Reward}(L')],
$$

and then

$$
p_i R_i + p_i p_j R_j \geq p_j R_j + p_i p_j R_i,
$$

or

$$
\frac{p_i R_i}{1 - p_i} \geq \frac{p_j R_j}{1 - p_j}.
$$

Therefore, to maximize the total expected reward, questions should be ordered in decreasing order of $p_i R_i/(1 - p_i)$.

3.3.4 Exercises

Exercise 3.3.1 Consider the optimal stopping, asset selling problem discussed in class. Suppose that the offers $w_k$ are i.i.d. random variables, Unif[500, 2000]. For $N = 10$, compute the thresholds $\alpha_k$, $k = 0, 1, \ldots, 9$, for $r = 0.05$ and $r = 0.1$. Recall that $\alpha_N = 0$. Also compute the expected value $J_0(0)$ for both interest rates.

Exercise 3.3.2 Consider again the optimal stopping, asset selling problem discussed in class.
(a) For the stationary, limiting policy defined by $\bar{\alpha}$, where $\bar{\alpha}$ is the solution to the equation
\[
(1 + r)\alpha = F(\alpha)\alpha + \int_{\alpha}^{\infty} wdF(w)
\]
Prove that $G(\alpha)$, defined as
\[
G(\alpha) = \frac{1}{r + 1} \left( F(\alpha)\alpha + \int_{\alpha}^{\infty} wdF(w) \right),
\]
is a contraction mapping, and hence the iterative procedure $\alpha_{n+1} = G(\alpha_n)$ finds the unique fixed point in $[0, E[w]/r]$, starting from any arbitrary $\alpha_0 \in [0, E[w]/r]$.

Recall: $G$ is a contraction mapping if for all $x$ and $y$, $||G(x) - G(y)|| < \theta ||x - y||$, for a constant $0 \leq \theta < 1$, $\theta$ independent of $x, y$.

(b) Apply the iterative procedure to compute $\bar{\alpha}$ over the scenarios described in Exercise 1.

(c) Compute the expected value $\tilde{J}_0(0)$ for Problem 1 when the controller applies control $\bar{\alpha}$ in every stage. Compare the results and comment on them.

**Exercise 3.3.3 (The job/secretary/partner selection problem)** A collection of $N \geq 2$ objects is observed randomly and sequentially one at a time. The observer may either select the current object observed, in which case the selection process is terminated, or reject the object and proceed to observe the next. The observer can rank each object relative to those already observed, and the objective is to maximize the probability of selecting the “best” object according to some criterion. It is assumed that no two objects can be judged to be equal. Let $r^*$ be the smallest positive integer $r$ such that
\[
\frac{1}{N - 1} + \frac{1}{N - 2} + \cdots + \frac{1}{r} \leq 1
\]
Show that an optimal policy requires that the first $r^*$ objects be observed. If the $r^*$th object has rank 1 relative to the others already observed, it should be selected; otherwise, the observation process should be continued until an object of rank 1 relative to those already observed is found.

**Hint:** Assume uniform distribution of the objects, i.e., if the $r$th object has rank 1 relative to the previous $(r - 1)$ objects, then the probability that it is the best is $r/N$. Define the state of the system as $x_k = \begin{cases} T & \text{if the selection has already terminated}, \\ 1 & \text{if the } k\text{th object observed has rank 1 among the first } k\text{ objects}, \\ 0 & \text{if the } k\text{th object observed has rank } > 1 \text{ among the first } k\text{ objects}. \end{cases}$

For $k \geq r^*$, let $J_k(0)$ be the maximal probability of finding the best object assuming $k$ objects have been observed and the $k$th object is not best relative to the previous $(k - 1)$ objects. Show that
\[
J_k(0) = \frac{k}{N} \left( \frac{1}{N - 1} + \cdots + \frac{1}{k} \right).
\]
Analogously, let $J_k(1)$ be the maximal probability of finding the best object assuming $k$ objects have been observed and the $k$th object is indeed the best relative to the previous $(k - 1)$ objects. Show that
\[
J_k(1) = \frac{k}{N}.
\]
Then, analyze the case $k < r^*$.
Exercise 3.3.4 A driver is looking for parking on the way to his destination. Each parking place is free with probability $p$ independently of whether other parking places are free or not. The driver cannot observe whether a parking place is free until he reaches it. If he parks $k$ places from his destination, he incurs a cost $k$. If he reaches the destination without having parked, the cost is $C$.

(a) Let $F_k$ be the minimal expected cost if he is $k$ parking places from his destination, where $F_0 = C$. Show that

$$F_k = p \min\{k, F_{k-1}\} + qF_{k-1}, \quad k = 1, 2, \ldots,$$

where $q = 1 - p$.

(b) Show that an optimal policy is of the form: “Never park if $k \geq k^*$, but take the first free place if $k < k^*$”, where $k$ is the number of parking places from the destination, and

$$k^* = \min \{ i : i \text{ integer}, q^{i-1} < (pC + q)^{-1} \}$$

Exercise 3.3.5 (Hardy’s Theorem) Let $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_n\}$ be monotonically nondecreasing sequences of numbers. Let us associate with each $i = 1, \ldots, n$, a distinct index $j_i$, and consider the expression $\sum_{i=1}^{n} a_i b_{j_i}$. Use an interchange argument to show that this expression is maximized when $j_i = i$ for all $i$, and is minimized when $j_i = n - i + 1$ for all $i$. 
Chapter 4

DP with Imperfect State Information.

So far we have studied the problem that the controller has access to the exact value of the current state, but this assumption is sometimes unrealistic. In this chapter, we will study the problems with imperfect state information. In this setting, we suppose that the controller receives some noisy observations about the value of the current state instead of the actual underlying states.

4.1 Reduction to the perfect information case

Basic problem with imperfect state information

- Suppose that the controller has access to observations $z_k$ of the form

$$z_0 = h_0(x_0, v_0), \quad z_k = h_k(x_k, u_{k-1}, v_k), \quad k = 1, 2, ..., N - 1,$$

where

$$z_k \in Z_k \quad \text{(observation space)}$$

$$v_k \in V_k \quad \text{(random observation disturbances)}$$

The random observation disturbance $v_k$ is characterized by a probability distribution

$$P_{v_k}(|x_k, ..., x_0, u_{k-1}, ..., u_0, w_{k-1}, ..., w_0, v_{k-1}, ..., v_0)$$

- Initial state $x_0$
- Control $\mu_k \in U_k \subseteq C_k$
- Define $I_k$ the information available to the controller at time $k$ and call it the information vector

$$I_k = (z_0, z_1, ..., z_k, u_0, u_1, ..., u_{k-1})$$

Consider a class of policies consisting of a sequence of functions $\pi = \{\mu_0, \mu_1, ..., \mu_{N-1}\}$ where $\mu_k(I_k) \in U_k$ for all $I_k, \quad k = 0, 1, ..., N - 1$
Objective: Find an admissible policy \( \pi = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\} \) that minimizes the cost function

\[
J_\pi = \mathbb{E}_{x_0, w_k, v_k} \left[ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(I_k), w_k) \right]
\]

subject to the system equation

\[
x_{k+1} = f_k(x_k, \mu_k(I_k), w_k), \quad k = 0, 1, \ldots, N - 1,
\]

and the measurement equation

\[
\begin{align*}
z_0 &= h_0(x_0, v_0) \\
z_k &= h_k(x_k, \mu_{k-1}(I_{k-1}), v_k), \quad k = 1, 2, \ldots, N - 1
\end{align*}
\]

Note the difference from the perfect state information case. In perfect state information case, we tried to find a rule that would specify the control \( u_k \) to be applied for each state \( x_k \) at time \( k \). However, now we are looking for a rule that gives the control to be applied for every possible information vector \( I_k \), for every sequence of observations received and controls applied up to time \( k \).

Example: Multiaccess Communication

- Consider a group of transmitting stations sharing a common channel

![Diagram of multiaccess communication](image)

- Stations are synchronized to transmit packets of data at integer times
- Each packet requires one slot (time unit) for transmission

Let

\[
\begin{aligned}
a_k &= \text{Number of packet arrivals during slot } k \text{ (with a given probability distribution)} \\
x_k &= \text{Number of packet waiting to be transmitted at the beginning of slot } k \text{ (backlog)}
\end{aligned}
\]

- Packet transmissions are scheduled using a strategy called *slotted Aloha protocol*:
  - Each packet in the system at the beginning of slot \( k \) is transmitted during that slot with probability \( u_k \) (common for all packets)
  - If two or more packets are transmitted simultaneously, they collide and have to rejoin the backlog for retransmission at a later slot
  - Stations can observe the channel and determine whether in any one slot:
    1. there was a collision
2. a success in the slot 
3. nothing happened (i.e., idle slot)

- **Control**: transmission probability $u_k$
- **Objective**: keep backlog small, so we assume a cost per stage $g_k(x_k)$, with $g_k(.)$ a monotonically increasing function of $x_k$
- **State of system**: size of the backlog $x_k$ (unobservable)
- **System equation**: 
  \[ x_{k+1} = x_k + a_k - t_k \]
  where $a_k$ is the number of new arrivals, and $t_k$ is the number of packets successfully transmitted during slot $k$. The distribution of $t_k$ is given by
  \[
  t_k = \begin{cases} 
  1 \text{ (success) w.p. } x_k u_k (1 - u_k)^{x_k-1} \quad \text{(i.e., } \mathbb{P}(\text{one Tx, } x_k - 1 \text{ do not Tx})) \\
  0 \text{ (failure) w.p. } 1 - x_k u_k (1 - u_k)^{x_k-1} 
  \end{cases}
  \]
- **Measurement equation**: 
  \[
  z_{k+1} = v_{k+1} = \begin{cases} 
  \text{“idle” w.p. } (1 - u_k)^{x_k} \\
  \text{“success” w.p. } x_k u_k (1 - u_k)^{x_k-1} \\
  \text{“collision” w.p. } 1 - (1 - u_k)^{x_k} - x_k u_k (1 - u_k)^{x_k-1} 
  \end{cases}
  \]
  where $z_{k+1}$ is the observation obtained at the end of the $k$th slot

**Reformulated as a perfect information problem**

Candidate for state is the information vector $I_k$

\[
I_{k+1} = (I_k, z_{k+1}, u_k), \quad k = 0, 1, \ldots, N - 2, \quad I_0 = z_0
\]

The state of the system is $I_k$, the control is $u_k$ and $z_{k+1}$ can be viewed as a random disturbance. Furthermore, we have

\[
\mathbb{P}(z_{k+1}|I_k, u_k) = \mathbb{P}(z_{k+1}|I_k, u_k, z_0, z_1, \ldots, z_k)
\]

Note that the prior disturbances $z_0, z_1, \ldots, z_k$ are already included in the information vector $I_k$. So, in the LHS we now have the system in the framework of basic DP where the probability distribution of $z_{k+1}$ depends explicitly only on the state $I_k$ and control $u_k$ of the new system and not on the prior disturbances (although implicitly it does through $I_k$).

- **The cost per stage**: 
  \[
  \tilde{g}_k(I_k, u_k) = \mathbb{E}_{x_k, w_k}[g_k(x_k, u_k, w_k)|I_k, u_k]
  \]
  Note that the new formulation is focused on past info and controls rather than on original system disturbances $w_k$.
- **DP algorithm**: 
  \[
  J_{N-1}(I_{N-1}) = \min_{u_{N-1} \in U_{N-1}} \{ \mathbb{E}_{x_{N-1}, w_{N-1}}[g_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1})) + g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1})|I_{N-1}, u_{N-1}] \}
  \]
and for $k = 0, 1, ..., N - 2$,

$$J_k(I_k) = \min_{u_k \in U_k} \{ E_{x_k, w_k, x_{k+1}} [g_k(x_k, u_k, w_k) + \underbrace{J_{k+1}(I_{k+1}, z_{k+1}, u_k)}_{I_{k+1}}] \}$$

We minimize the RHS for every possible value of the vector $I_{k+1}$ to obtain $\mu^*_k(I_{k+1})$. The optimal cost is given by $J_0^* = E_0 [J_0(z_0)]$.

**Example 4.1.1 (Machine repair)**

- A machine can be in one of two unobservable states: $P$ (good state) and $P$ (bad state)
- State space: $\{P, \bar{P}\}$
- Number of periods: $N = 2$
- At the end of each period, the machine is inspected with two possible inspection outcomes: $G$ (probably good state), $B$ (probably bad state)
- Control space: actions after each inspection, which could be either
  - $C$: continue operation of the machine; or
  - $S$: stop, diagnose its state and if it is in bad state $\bar{P}$, repair.
- Cost per stage: $g(P, C) = 0; \ g(P, S) = 1; \ g(\bar{P}, C) = 2; \ g(\bar{P}, S) = 1$
- Total cost: $g(x_0, u_0) + g(x_1, u_1)$ (assume zero terminal cost)
- Let $x_0, x_1$ be the state of the machine at the end of each period
- Distribution of initial state: $P(x_0 = P) = \frac{2}{3}, \ P(x_0 = \bar{P}) = \frac{1}{3}$
- Assume that we start with a machine in good state, i.e., $x_{-1} = P$
- System equation:

\[
x_{k+1} = w_k, \quad k = 0, 1
\]

where the transition probabilities are given by
Note that we do not have perfect state information, since the inspection does not reveal the state of the machine with certainty. Rather, the result of each inspection may be viewed as a noisy measure of the system state.

Result of inspections: \( z_k = v_k, \quad k = 0, 1; \quad v_k \in \{B, G\} \)

**Information vector:**

\[
I_0 = z_0, \quad I_1 = (z_0, z_1, u_0)
\]

and we seek functions \( \mu_0(I_0), \mu_1(I_1) \) that minimize

\[
E_{x_0, w_0, w_1} \left[ g(x_0, \mu_0(I_0)) + g(x_1, \mu_1(I_1)) \right]
\]

**DP algorithm.** Terminal condition: \( J_2(I_2) = 0 \) for all \( I_2 \)

For \( k = 0, 1 \):

\[
J_k(I_k) = \min \left\{ \begin{array}{ll}
\mathbb{P}(x_k = P|I_k, C) g(P, C) + \mathbb{P}(x_k = P|I_k, C) g(P, C) + E_{z_{k+1}}[J_{k+1}(I_k, z_{k+1}, C)|I_k, C], \\
\mathbb{P}(x_k = P|I_k, S) g(P, S) + \mathbb{P}(x_k = P|I_k, S) g(P, S) + E_{z_{k+1}}[J_{k+1}(I_k, z_{k+1}, S)|I_k, S]\end{array} \right. 
\]

**Last stage** (\( k = 1 \)): compute \( J_1(I_1) \) for each possible \( I_1 = (z_0, z_1, u_0) \). Recalling that \( J_2(I) = 0, \forall I \), we have

\[
\text{cost of } C = 2\mathbb{P}(x_1 = \bar{P}|I_1), \quad \text{cost of } S = 1,
\]

and therefore \( J_1(I_1) = \min\{2\mathbb{P}(x_1 = \bar{P}|I_1), 1\} \). Compute probability \( \mathbb{P}(x_1 = \bar{P}|I_1) \) for all possible realizations of \( I_1 = (z_0, z_1, u_0) \) by using the conditional probability formula:

\[
\mathbb{P}(X|A, B) = \frac{\mathbb{P}(X, A|B)}{\mathbb{P}(A|B)}.
\]

There are 8 cases to consider. We describe here 3 of them.

(1) For \( I_1 = (G, G, S) \)

\[
\mathbb{P}(x_1 = \bar{P}|G, G, S) = \frac{\mathbb{P}(x_1 = \bar{P}, G, G|S)}{\mathbb{P}(G, G|S)} = \frac{1}{7}
\]
Numerator:

\[ P(x_1 = \bar{P}, G, G | S) = \left( \frac{2}{3} \times \frac{3}{4} \times \frac{1}{3} \times \frac{1}{4} \right) + \left( \frac{1}{3} \times \frac{1}{4} \times \frac{1}{3} \times \frac{1}{4} \right) = \frac{7}{144} \]

Denominator:

\[ P(G, G | S) = \left( \frac{2}{3} \times \frac{3}{4} \times \frac{2}{3} \times \frac{3}{4} \right) + \left( \frac{2}{3} \times \frac{3}{4} \times \frac{1}{3} \times \frac{1}{4} \right) + \left( \frac{1}{3} \times \frac{1}{4} \times \frac{2}{3} \times \frac{3}{4} \right) + \left( \frac{1}{3} \times \frac{1}{4} \times \frac{1}{3} \times \frac{1}{4} \right) = \frac{49}{144} \]

Hence,

\[ J_1(G, G, S) = 2P(x_1 = \bar{P} | G, G, S) = \frac{2}{7} < 1, \quad \mu^*_1(G, G, S) = C \]

(2) For \( I_1 = (B, G, S) \)

\[ P(x_1 = \bar{P} | B, G, S) = \frac{1}{7} \]

Numerator:

\[ P(x_1 = \bar{P}, B, G | S) = \frac{1}{4} \times \frac{1}{3} \times \left( \frac{1}{3} \times \frac{2}{3} + \frac{3}{4} \times \frac{1}{3} \right) = \frac{5}{144} \]
Denominator:
\[ P(B, G | S) = \frac{2}{3} \times \frac{1}{3} \times \left( \frac{2}{3} \times \frac{2}{4} + \frac{1}{3} \times \frac{1}{4} \right) + \frac{1}{3} \times \frac{3}{4} \times \left( \frac{2}{3} \times \frac{3}{4} + \frac{1}{3} \times \frac{1}{4} \right) \]

Hence,
\[ J_1(B, G, S) = \frac{2}{7}, \quad \mu_1(B, G, S) = C \]

(3) For \( I_1 = (G, B, S) \)
\[ P(x_1 = \bar{P} | G, B, S) = \frac{P(x_1 = \bar{P}, G, B | S)}{P(G, B | S)} = \frac{3}{5} \]

Numerator:
\[ P(x_1 = \bar{P}, G, B | S) = \frac{3}{4} \times \frac{1}{3} \times \left( \frac{3}{4} \times \frac{2}{3} + \frac{1}{4} \times \frac{1}{3} \right) = \frac{7}{48} \]
Denominator:

\[ P(G, B | S) = \frac{1}{4} \times \frac{2}{3} \times \left( \frac{3}{4} \times \frac{2}{3} + \frac{1}{3} \times \frac{1}{3} \right) + \frac{3}{4} \times \frac{1}{3} \times \left( \frac{3}{4} \times \frac{2}{3} + \frac{1}{2} \times \frac{1}{3} \right) = \frac{35}{144} \]

Hence,

\[ J_1(G, B, S) = 1, \quad \mu^*_1(G, B, S) = S \]

Summary: For all other 5 cases of \( I_1 \), we compute \( J_1(I_1) \) and \( \mu^*_1(I_1) \). The optimal policy is to continue \( (u_1 = C) \) if the result of last inspection was \( G \) and to stop \( (u_1 = S) \) if the result of the last inspection was \( B \).

First stage \( (k = 0) \): Compute \( J_0(I_0) \) for each of the two possible information vectors \( I_0 = (G) \), \( I_0 = (B) \). We have

\[
\text{cost of } C = 2P(x_0 = \bar{P} | I_0, C) + E_{z_1} \{ J_1(I_0, z_1, C) | I_0, C \} = 2P(x_0 = \bar{P} | I_0, C) + P(z_1 = G | I_0, C)J_1(I_0, G, C) + P(z_1 = B | I_0, C)J_1(I_0, B, C)
\]

\[
\text{cost of } S = 1 + E_{z_1} \{ J_1(I_0, z_1, S) | I_0, S \} = 1 + P(z_1 = G | I_0, S)J_1(I_0, G, S) + P(z_1 = B | I_0, S)J_1(I_0, B, S),
\]

using the values of \( J_1 \) from previous stage. Thus, we have

\[ J_0(I_0) = \min \{ \text{cost of } C, \text{cost of } S \} \]

The optimal cost is

\[ J^* = P(G)J_0(G) + P(B)J_0(B) \]

For illustration, we compute one of the values. For example, for \( I_0 = G \)

\[
P(z_1 = G | G, C) = \frac{P(z_1 = G, u_0 = C) \bigg| \bar{P}(G | C) \bigg| z_0}{P(G | C)} = \frac{P(z_1 = G, G | C)}{P(G)} = \frac{15}{72} = \frac{15}{28}
\]

Note that the \( P(G | C) = P(G) \) follows since \( z_0 = G \) is independent of the control \( u_0 = C \) or \( u_0 = S \).
CHAPTER 4. DP WITH IMPERFECT STATE INFORMATION.

Prof. R. Caldentey

Numerator:

\[ P(z_1 = G, G|C) = \frac{2}{3} \times \frac{3}{4} \times \left( \frac{2}{3} \times \frac{3}{4} + \frac{1}{3} \times \frac{1}{4} \right) + \frac{1}{3} \times \frac{1}{4} \times 1 \times \frac{1}{4} = \frac{15}{48} \]

Similarly, we can compute \( P(B) = \frac{2}{3} \times \frac{1}{4} + \frac{1}{3} \times \frac{3}{4} = \frac{5}{12} \)

Denominator:

\[ P(G) = \frac{2}{3} \times \frac{3}{4} + \frac{1}{3} \times \frac{1}{4} = \frac{7}{12} \]

Similarly, we can compute \( P(B) = \frac{2}{3} \times \frac{1}{4} + \frac{1}{3} \times \frac{3}{4} = \frac{5}{12} \)
Note: The optimal policy for both stages is to continue (C) if the result of latest inspection is G and to stop and repair (S) otherwise. The optimal cost can be proved to be $J^* = \frac{176}{144}$.

Problem: The DP can be computationally prohibitive if the number of information vectors $I_k$ is large or infinite. □

4.2 Linear-Quadratic Systems and Sufficient Statistics

In this section, we consider again the problem studied in Section 2.5, but now under the assumption that the controller does not observe the real state of the system $x_k$, but just a noisy representation of it, $z_k$. Then, we investigate how we can reduce the quantity of information needed to solve problems under imperfect state information.

4.2.1 Linear-Quadratic systems

Problem setup

System equation: $x_{k+1} = A_k x_k + B_k u_k + w_k$ [Linear in both state and control.]

Quadratic cost:

$$E_{x_0, w_0, \ldots, w_{N-1}} \left\{ x_N' Q_N x_N + \sum_{k=0}^{N-1} (x_k' Q_k x_k + u_k' R_k u_k) \right\},$$

where:

- $Q_k$ are square, symmetric, positive semidefinite matrices with appropriate dimension,
- $R_k$ are square, symmetric, positive definite matrices with appropriate dimension,
- Disturbances $w_k$ are independent with $E[w_k] = 0$, finite variance, and independent of $x_k$ and $u_k$.
- Controls $u_k$ are unconstrained, i.e., $u_k \in \mathbb{R}^n$.

Observations: Driven by a linear measurement equation:

$$z_k = C_k x_k + v_k, \quad k = 0, 1, \ldots, N - 1,$$

where $v_k$s are mutually independent, and also independent from $w_k$ and $x_0$.

Key fact to show: Given an information vector $I_k = (z_0, \ldots, z_k, u_0, \ldots, u_{k-1})$, the optimal policy $\{\mu^*_0, \ldots, \mu^*_{N-1}\}$ is of the form

$$\mu^*_k(I_k) = L_k E[x_k|I_k],$$

where

- $L_k$ is the same as for the perfect state info case, and solves the “control problem”.
- $E[x_k|I_k]$ solves the “estimation problem”.

This means that the control and estimation problems can be solved separately.
DP algorithm

The DP algorithm becomes:

- At stage \( N - 1 \),
  \[
  J_{N-1}(I_{N-1}) = \min_{u_{N-1}} E_{x_{N-1}, w_{N-1}} \left[ x'_{N-1}Q_{N-1}x_{N-1} + u'_{N-1}R_{N-1}u_{N-1} \right. \\
  \left. + (A_{N-1}x_{N-1} + B_{N-1}u_{N-1} + w_{N-1})'Q_N(A_{N-1}x_{N-1} + B_{N-1}u_{N-1} + w_{N-1}) | I_{N-1}, u_{N-1} \right]
  \]
  \[
  \tag{4.2.1}
  \]

  - Recall that \( w_{N-1} \) is independent of \( x_{N-1} \), and that both are random at stage \( N - 1 \); that’s why we take expected value over both of them.
  - Since the \( w_k \) are mutually independent and do not depend on \( x_k \) and \( u_k \) either, we have
    \[
    E[w_{N-1}|I_{N-1}, u_{N-1}] = E[w_{N-1}|I_{N-1}] = E[w_{N-1}] = 0,
    \]
  then the minimization just involves
    \[
    \min_{u_{N-1}} \left\{ u'_{N-1}(B'_{N-1}QNB_{N-1} + R_{N-1})u_{N-1} + 2E[x_{N-1}|I_{N-1}]'A'_{N-1}Q_NB_{N-1}u_{N-1} \right\}_{>0} \]

  Taking derivative of the argument with respect to \( u_{N-1} \), we have the first order condition:
  \[
  2(B'_{N-1}Q_NB_{N-1} + R_{N-1})u_{N-1} + 2E[x_{N-1}|I_{N-1}]'A'_{N-1}Q_NB_{N-1} = 0.
  \]

  This yields the optimal \( u^*_N \):
  \[
  u^*_N = \mu^*_N(I_{N-1}) = L_{N-1}E[x_{N-1}|I_{N-1}],
  \]
  where
  \[
  L_{N-1} = -(B'_{N-1}Q_NB_{N-1} + R_{N-1})^{-1}B'_{N-1}Q_NA_{N-1}.
  \]
  Note that this is very similar to the perfect state info counterpart, except that now \( x_{N-1} \) is replaced by \( E[x_{N-1}|I_{N-1}] \).

  - Substituting back in \((4.2.1)\), we get:
    \[
    J_{N-1}(I_{N-1}) = E_{x_{N-1}} \left[ x'_{N-1}K_{N-1}x_{N-1}|I_{N-1} \right] \quad \text{(quadratic in } x_{N-1}) \]
    \[
    + E_{x_{N-1}} \left[ (x_{N-1} - E[x_{N-1}|I_{N-1}])'P_{N-1}(x_{N-1} - E[x_{N-1}|I_{N-1}]) | I_{N-1} \right] \quad \text{(quadratic in estimation error } x_{N-1} - E[x_{N-1}|I_{N-1}]) \]
    \[
    + E_{w_{N-1}} \left[ w'_{N-1}Q_Nw_{N-1} \right] \quad \text{(constant term)},
    \]
    where the matrices \( K_{N-1} \) and \( P_{N-1} \) are given by
    \[
    P_{N-1} = A'_{N-1}Q_NB_{N-1}(R_{N-1} + B'_{N-1}Q_NB_{N-1})^{-1}B'_{N-1}Q_NA_{N-1},
    \]
    and
    \[
    K_{N-1} = A'_{N-1}Q_NA_{N-1} - P_{N-1} + Q_{N-1}.
    \]
– Note the structure of $J_{N-1}$: In addition to the quadratic and constant terms (which are identical to the perfect state info case for a given state $x_{N-1}$), it involves a quadratic term in the estimation error

$$ x_{N-1} - E[x_{N-1}|I_{N-1}]. $$

In words, the estimation error is penalized quadratically in the value function.

- At stage $N-2$,

$$ J_{N-2}(I_{N-2}) = \min_{u_{N-2}} E_{x_{N-2}, w_{N-2}, z_{N-1}} \left[ x_{N-2}' Q N_{-2} x_{N-2} + u_{N-2}' R N_{-2} u_{N-2} + J_{N-1}(I_{N-1}) | I_{N-2}, u_{N-2} \right] $$

$$ = E[x_{N-2}' Q N_{-2} x_{N-2} | I_{N-2}] + \min_{u_{N-2}} \left\{ u_{N-2}' R N_{-2} u_{N-2} + E[x_{N-1}' K N_{-1} x_{N-1} | I_{N-2}, u_{N-2}] \right\} $$

$$ + E \left[ \left( x_{N-1} - E[x_{N-1}|I_{N-1}] \right)' P_{N-1} \left( x_{N-1} - E[x_{N-1}|I_{N-1}] \right) | I_{N-2}, u_{N-2} \right] + E_{w_{N-1}} \left[ w_{N-1}' Q N w_{N-1} \right]. $$

\textbf{Key point (to be proved):} The term (4.2.2) turns out to be independent of $u_{N-2}$, and so we can exclude it from the minimization with respect to $u_{N-2}$.

This says that the quality of estimation as expressed by the statistics of the error $x_k - E[x_k|I_k]$ cannot be influenced by the choice of control, which is not very intuitive!

For the next result, we need the linearity of both system and measurement equations.

**Lemma 4.2.1 (Quality of Estimation)** For every stage $k$, there is a function $M_k(\cdot)$ such that

$$ M_k(x_0, w_0, \ldots, w_{k-1}, v_0, \ldots, v_k) = x_k - E[x_k|I_k], $$

independently of the policy being used.

\textbf{PROOF:} Fix a policy, and consider the following two systems:

1. There is a control $u_k$ being implemented, and the system evolves according to

$$ x_{k+1} = A_k x_k + B_k u_k + w_k, \quad z_k = C_k x_k + v_k. $$

2. There is no control being applied, and the system evolves according to

$$ \bar{x}_{k+1} = A_k \bar{x}_k + \bar{w}_k, \quad \bar{z}_k = C_k \bar{x}_k + \bar{v}_k. \quad (4.2.3) $$

Consider the evolution of the two systems from identical initial conditions: $x_0 = \bar{x}_0$; and when system disturbances and observation noise vectors are also identical:

$$ w_k = \bar{w}_k, \quad v_k = \bar{v}_k, \quad k = 0, 1, \ldots, N - 1. $$

Consider the vectors:

$$ Z^k = (z_0, \ldots, z_k)', \quad \bar{Z}^k = (\bar{z}_0, \ldots, \bar{z}_k)', \quad W^k = (w_0, \ldots, w_k)', $$

$$ V^k = (v_0, \ldots, v_k)', \quad \text{and} \quad U^k = (u_0, \ldots, u_k)'. $$
CHAPTER 4. DP WITH IMPERFECT STATE INFORMATION.

Prof. R. Caldentey

Applying the system equations above for stages 0, 1, \ldots, k, their linearity implies the existence of matrices $F_k, G_k$ and $H_k$ such that:

\[
x_k = F_k x_0 + G_k U_k^{k-1} + H_k W^{k-1},
\]

\[
\bar{x}_k = F_k x_0 + H_k W^{k-1}.
\]

Note that the vector $U^{k-1} = (u_0, \ldots, u_k)\prime$ is part of the information vector $I_k$, as verified below:

\[
I_k = (z_0, \ldots, z_k, u_0, \ldots, u_k)_{U^{-1}}, \quad k = 1, \ldots, N - 1,
\]

\[
I_0 = z_0.
\]

Then, $U^{k-1} = E[U^{k-1}|I_k]$, and conditioning with respect to $I_k$ in (5.3.2) and (4.2.5):

\[
E[x_k|I_k] = F_k E[x_0|I_k] + G_k U^{k-1} + H_k E[W^{k-1}|I_k]
\]

(4.2.6)

\[
E[\bar{x}_k|I_k] = F_k E[x_0|I_k] + H_k E[W^{k-1}|I_k].
\]

(4.2.7)

Then,

\[
\frac{x_k - E[x_k|I_k]}{\bar{x}_k - E[\bar{x}_k|I_k]} = \text{from (5.3.2)} \quad \frac{x_k - E[x_k|I_k]}{\bar{x}_k - E[\bar{x}_k|I_k]} = \text{from (4.2.6)} \quad \frac{x_k - E[x_k|I_k]}{\bar{x}_k - E[\bar{x}_k|I_k]} = \text{from (4.2.7)}
\]

where the term $G_k U^{k-1}$ gets canceled. The intuition for this is that the linearity of the system equation affects "equally" the true state $x_k$ and our estimation of it, $E[x_k|I_k]$.

Applying now the measurement equations above for 0, 1, \ldots, k, their linearity implies the existence of a matrix $R_k$ such that:

\[
Z^k - \bar{Z}^k = R_k U^{k-1}
\]

Note that $Z^k$ involves the term $B_{k-1} u_{k-1}$ from the system equation for $x_k$, and recursively we can build such a matrix $R_k$. In addition, from (4.2.3) above and the sample path identity for the disturbances, $\bar{Z}^k$ depends on the original $w_k, v_k$ and $x_0$:

\[
Z^k - \bar{Z}^k = R_k U^{k-1} \Rightarrow \bar{Z}^k = Z^k - R_k U^{k-1} = S_k W^{k-1} + T_k V^k + D_k x_0,
\]

where $S_k, T_k, D_k$ are matrices of appropriate dimension. Thus, the information provided by $I_k = (Z^k, U^{k-1})$ regarding $\bar{x}_k$ is summarized in $\bar{Z}^k$, and we have

\[
E[\bar{x}_k|I_k] = E[\bar{x}_k|Z^k],
\]

so that

\[
x_k - E[x_k|I_k] = \bar{x}_k - E[\bar{x}_k|I_k] = \bar{x}_k - E[\bar{x}_k|\bar{Z}^k].
\]

Therefore, the function $M_k$ to use is

\[
M_k(x_0, w_0, \ldots, w_{k-1}, v_0, \ldots, v_k) = \bar{x}_k - E[\bar{x}_k|\bar{Z}_k],
\]

which does not depend on the controls $u_0, \ldots, u_{k-1}$.

Going back to the DP equation $J_{N-2}(I_{N-2})$, and using the Quality of Estimation Lemma, we get

\[
\xi_{N-1} \triangleq M_{N-1}(x_0, w_0, \ldots, w_{N-2}, v_0, \ldots, v_{N-1}) = x_{N-1} - E[x_{N-1}|I_{N-1}].
\]

147
Since \( \xi_{N-1} \) is independent of \( u_{N-2} \), we have
\[
E[\xi_{N-1}P_{N-1}\xi_{N-1}|I_{N-2}, u_{N-2}] = E[\xi_{N-1}P_{N-1}\xi_{N-1}|I_{N-2}].
\]

So, going back to the DP equation for \( J_{N-2}(I_{N-2}) \), we can drop the term (4.2.2) to minimize over \( u_{N-2} \), and similarly to stage \( N-1 \), the minimization yields
\[
u^*_{N-2} = \mu^*_{N-2}(I_{N-2}) = L_{N-2}E[x_{N-2}|I_{N-2}].
\]
Continuing similarly (using also the Quality of Estimation Lemma), we have
\[
\mu^*_k(I_k) = L_kE[x_k|I_k],
\]
where \( L_k \) is the same as for perfect state info:
\[
L_k = -(R_k + B_k'K_{k+1}B_k)^{-1}B_k'K_{k+1}A_k,
\]
with \( K_k \) generated from \( K_N = Q_N \), using
\[
K_k = A_k'K_{k+1}A_k - P_k + Q_k,
\]
\[
P_k = A_k'K_{k+1}B_k(R_k + B_k'K_{k+1}B_k)^{-1}B_k'K_{k+1}A_k.
\]
The optimal controller is represented in Figure 4.2.1.

**Figure 4.2.1:** Structure of the optimal controller for the L-Q problem.

**Separation interpretation:**

1. The optimal controller can be decomposed into:
   - An *estimator*, which uses the data to generate the conditional expectation \( E[x_k|I_k] \).
   - An *actuator*, which multiplies \( E[x_k|I_k] \) by the gain matrix \( L_k \) and applies the control \( u_k = L_kE[x_k|I_k] \).
2. Observation: Consider the problem of finding the estimate $\hat{x}$ of a random vector $x$ given some information (random vector) $I$, which minimizes the mean squared error

$$E_x[||x - \hat{x}||^2|I] = E[||x||^2] - 2E[x|I]\hat{x} + ||\hat{x}||^2.$$ 

When we take derivative with respect to $\hat{x}$ and set it equal to zero:

$$2\hat{x} - 2E[x|I] = 0 \Rightarrow \hat{x} = E[x|I],$$

which is exactly our estimator.

3. The estimator portion of the optimal controller is optimal for the problem of estimating the state $x_k$ assuming the control is not subject to choice.

4. The actuator portion is optimal for the control problem assuming perfect state information.

### 4.2.2 Implementation aspects – Steady-state controller

- In the imperfect info case, we need to compute an estimator $\hat{x}_k = E[x_k|I_k]$, which is indeed the one that minimizes the mean squared error $E_x[||x - \hat{x}||^2|I]$.

- However, this is computationally hard in general.

- Fortunately, if the disturbances $w_k$ and $v_k$, and the initial state $x_0$ are Gaussian random vectors, a convenient implementation of the estimator is possible by means of the Kalman filter algorithm.

This algorithm produces $\hat{x}_{k+1}$ at time $k + 1$ just depending on $z_{k+1}$, $u_k$ and $\dot{x}_k$.

**Kalman filter recursion:** For all $k = 0, 1, \ldots, N - 1$, compute

$$\hat{x}_{k+1} = A_k\hat{x}_k + B_ku_k + \Sigma_{k+1|k+1}C_0^{-1}(z_{k+1} - C_k\hat{x}_k + B_ku_k),$$

and

$$\hat{x}_0 = E[x_0] + \Sigma_{0|0}C_0^{-1}(x_0 - C_0E[x_0]),$$

where the matrices $\Sigma_{k|k}$ are precomputable and are given recursively by

$$\Sigma_{k+1|k+1} = \Sigma_{k+1|k} - \Sigma_{k+1|k}C_kC_k^{-1}(C_k\Sigma_{k+1|k}C_k^{-1} + N_{k+1})^{-1}C_k\Sigma_{k+1|k},$$

$$\Sigma_{k+1|k} = A_k\Sigma_{k|k}A_k^T + M_k, \quad k = 0, 1, \ldots, N - 1,$$

with

$$\Sigma_{0|0} = S - SC_0(C_0S + N_0)^{-1}C_0S.$$

In these equations, $M_k$, $N_k$, and $S$ are the covariance matrices of $w_k$, $v_k$ and $x_0$, respectively, and we assume that $w_k$ and $v_k$ have zero mean; that is

$$E[w_k] = E[v_k] = 0,$$

$$M_k = E[w_kw_k^T], \quad N_k = E[v_kv_k^T],$$

$$S = E[(x_0 - E[x_0])(x_0 - E[x_0])^T].$$

Moreover, we are assuming that matrices $N_k$ are positive definite (and hence, invertible).

---

1Recall that for a random vector $X$, its covariance matrix $\Sigma$ is given by $E[(X - E[X])(X - E[X])^T]$. Its entry $(i, j)$ is given by $\Sigma_{ij} = \text{Cov}(X_i, X_j) = E[(X_i - E[X_i])(X_j - E[X_j])]$. The covariance matrix $\Sigma$ is always positive semi-definite.
Stationary case

- Assume that the system and measurement equations are stationary (i.e., same distribution across time; $N_k = N$, $M_k = M$).

- Suppose that $(A, B)$ is controllable and that matrix $Q$ can be written as $Q = F'F$, where $F$ is a matrix such that $(A, F)$ is observable.\(^2\)

- By the theory of LQ-systems under perfect info, when $N \to \infty$ (i.e., the horizon length becomes large), the optimal controller tends to the steady-state policy

$$\mu_k^*(I_k) = L\hat{x}_k,$$

where

$$L = -(R + B'KB)^{-1}B'KA,$$

and where $K$ is the unique $\geq 0$ symmetric solution of the algebraic Riccati equation

$$K = A'(K - KB(R + B'KB)^{-1}B'K)A + Q.$$

- It can also be shown in the limit as $N \to \infty$, that

$$\hat{x}_{k+1} = (A + BL)\hat{x}_k + \hat{\Sigma}C'N^{-1}(z_{k+1} - C(A + BL)\hat{x}_k),$$

where $\hat{\Sigma}$ is given by

$$\hat{\Sigma} = \Sigma - \Sigma C'(C\Sigma C' + N)^{-1}C\Sigma,$$

and $\Sigma$ is the unique $\geq 0$ symmetric solution of the Riccati equation

$$\Sigma = A(\Sigma - \Sigma C'(C\Sigma C' + N)^{-1}C\Sigma)A' + M.$$

The assumptions required for this are:

1. $(A, C)$ is observable.
2. The matrix $M$ can be written as $M = DD'$, where $D$ is a matrix such that the pair $(A, D)$ is controllable.

Non-Gaussian uncertainty

When the uncertainty of the system is non-Gaussian, computing $E[x_k|I_k]$ may be very difficult from a computational viewpoint. So, a suboptimal solution is typically used.

A common suboptimal controller is to replace $E[x_k|I_k]$ by the estimate produced by the Kalman filter (i.e., act as if $x_0, w_k$ and $v_k$ are Gaussian).

A nice property of this approximation is that it can be proved to be optimal within the class of controllers that are linear functions of $I_k$.

\(^2\)Recall the definitions: A pair of matrices $(A, B)$, where $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, is said to be controllable if the $n \times (n, m)$ matrix: $[B, AB, A^2B, \ldots, A^{n-1}B]$ has full rank. A pair $(A, C), A \in \mathbb{R}^{n \times n}, C \in \mathbb{R}^{m \times n}$ is said to be observable if the pair $(A', C')$ is controllable.
4.2.3 Sufficient statistics

- Problem of DP algorithm under imperfect state info: Growing dimension of the reformulated state space $I_k$.

- Objective: Find sufficient statistics (ideally, of smaller dimension) for $I_k$ that summarize all the essential contents of $I_k$ as far as control is concerned.

- Recall the DP formulation for the imperfect state info case:

$$J_{N-1}(I_{N-1}) = \min_{u_{N-1} \in U_{N-1}} \mathbb{E}_{x_{N-1}, w_{N-1}} \left[ g_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1})) ight.$$

$$\left. + g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1})|I_{N-1}, u_{N-1}) \right]$$

$$J_k(I_k) = \min_{u_k \in U_k} \mathbb{E}_{x_k, w_k, z_{k+1}} \left[ g_k(x_k, u_k, w_k) + J_{k+1}(I_{k+1}, z_{k+1}, u_k)|I_k, u_k \right].$$

- Suppose that we can find a function $S_k(I_k)$ such that the RHS of (4.2.8) and (4.2.9) can be written in terms of some function $H_k$ as

$$\min_{u_k \in U_k} H_k(S_k(I_k), u_k),$$

such that

$$J_k(I_k) = \min_{u_k \in U_k} H_k(S_k(I_k), u_k).$$

- Such a function $S_k$ is called a sufficient statistic.

- An optimal policy obtained by the preceding minimization can be written as

$$\mu^*_k(I_k) = \bar{\mu}^*_k(S_k(I_k)),$$

where $\bar{u}^*_k$ is an appropriate function.

- Example of a sufficient statistic: $S_k(I_k) = I_k$.

- Another important sufficient statistic is the conditional probability distribution of the state $x_k$ given the information vector $I_k$, i.e.,

$$S_k(I_k) = P_{x_k|I_k}.$$

For this case, we need an extra assumption: The probability distribution of the observation disturbance $v_{k+1}$ depends explicitly only on the immediate preceding $x_k, u_k$ and $w_k$, and not on earlier ones.

- It turns out that $P_{x_k|I_k}$ is generated recursively by a dynamic system (estimator) of the form

$$P_{x_{k+1}|I_{k+1}} = \Phi_k(P_{x_k|I_k}, u_k, z_{k+1}),$$

for a suitable function $\Phi_k$ determined from the data of the problem. (We will verify this later)
• **Claim:** Suppose for now that function $\Phi_k$ in equation (4.3.1) exists. We will argue now that this is enough to solve the DP algorithm.

**Proof:** By induction. For $k = N - 1$ (i.e., to solve (4.2.8)), given the Markovian nature of the system, it is sufficient to know the distribution $P_{x_{N-1}I_{N-1}}$ together with the distribution $P_{w_{N-1}|x_{N-1},u_{N-1}}$, so that

$$J_{N-1}(I_{N-1}) = \min_{u_{N-1} \in U_{N-1}} H_{N-1}(P_{x_{N-1}|I_{N-1}}, u_{N-1}) = \bar{J}_{N-1}(P_{x_{N-1}|I_{N-1}})$$

for appropriate functions $H_{N-1}$ and $\bar{J}_{N-1}$.

III: Assume

$$J_{k+1}(I_{k+1}) = \min_{u_{k+1} \in U_{k+1}} H_{k+1}(P_{x_{k+1}|I_{k+1}}, u_{k+1}) = \bar{J}_{k+1}(P_{x_{k+1}|I_{k+1}}), \quad (4.2.11)$$

for appropriate functions $H_{k+1}$ and $\bar{J}_{k+1}$.

We want to show that there exist functions $H_k$ and $\bar{J}_k$ such that

$$J_k(I_k) = \min_{u_k \in U_k} H_k(P_{x_k|I_k}, u_k) = \bar{J}_k(P_{x_k|I_k}).$$

Using equations (4.3.1) and (4.2.11), the DP in (4.2.9) can be written as

$$J_k(I_k) = \min_{u_k \in U_k} E_{x_k, w_k, z_{k+1}} \left[ g_k(x_k, u_k, w_k) + \bar{J}_{k+1}(\Phi_k(P_{x_k|I_k}, u_k, z_{k+1}))|I_k, u_k \right]. \quad (4.2.12)$$

To solve this problem, we also need the joint distribution $P(x_k, w_k, z_{k+1}|I_k, u_k)$, or equivalently, given that from the primitives of the system,

$$z_{k+1} = h_{k+1}(x_{k+1}, u_k, v_{k+1}), \quad \text{and} \quad x_{k+1} = f_k(x_k, u_k, w_k),$$

we need

$$P(x_k, w_k, h_{k+1}(f_k(x_k, u_k, w_k), u_k, v_{k+1})|I_k, u_k).$$

This distribution can be expressed in terms of $P_{x_k|I_k}$, the given distributions

$$P(w_k|x_k, u_k), \quad P(v_{k+1}|f_k(x_k, u_k, w_k), u_k, w_k),$$

and the system equation $x_{k+1} = f_k(x_k, u_k, w_k)$.

Therefore, the expression minimized over $u_k$ in (4.2.12) can be written as a function of $P_{x_k|I_k}$ and $u_k$, and the DP equation (4.2.12) can be written as

$$J_k(I_k) = \min_{u_k \in U_k} H_k(P_{x_k|I_k}, u_k)$$

for a suitable function $H_k$. Thus, $P_{x_k|I_k}$ is a sufficient statistic.

• If the conditional distribution $P_{x_k|I_k}$ is uniquely determined by another expression $S_k(I_k)$, i.e., there exist a function $G_k$ such that

$$P_{x_k|I_k} = G_k(S_k(I_k)),$$

then $S_k(I_k)$ is also a sufficient statistic.

For example, if we can show that $P_{x_k|I_k}$ is a Gaussian distribution, then the mean and the covariance matrix corresponding to $P_{x_k|I_k}$ form a sufficient statistic.
• The representation of the optimal policy as a sequence of functions of $P_{x_k|I_k}$, i.e.,

$$\mu_k(I_k) = \bar{\mu}_k(P_{x_k|I_k}), \quad k = 0, 1, \ldots, N - 1,$$

is conceptually very useful. It provides a decomposition of the optimal controller in two parts:

1. An estimator, which uses at time $k$ the measurement $z_k$ and the control $u_{k-1}$ to generate the probability distribution $P_{x_k|I_k}$.
2. An actuator, which generates a control input to the system as a function of the probability distribution $P_{x_k|I_k}$.

This is illustrated in Figure 4.2.2.

![Figure 4.2.2: Conceptual separation of the optimal controller into an estimator and an actuator.](image)

- This separation is the basis for various suboptimal control schemes that split the controller a priori into an estimator and an actuator.
- The controller $\bar{\mu}_k(P_{x_k|I_k})$ can be viewed as controlling the “probabilistic state” $P_{x_k|I_k}$, so as to minimize the expected cost-to-go conditioned on the information $I_k$ available.

### 4.2.4 The conditional state distribution recursion

We still need to justify the recursion

$$P_{x_{k+1}|I_{k+1}} = \Phi_k(P_{x_k|I_k}, u_k, z_{k+1}) \quad (4.2.13)$$

For the case where the state, control, observation, and disturbance spaces are the real line, and all r.v. involved posses p.d.f., the conditional density $p(x_{k+1}|I_{k+1})$ is generated from $p(x_k|I_k), u_k,$ and $z_{k+1}$ by means of the equation:

\[
p(x_{k+1}|I_{k+1}) = p(x_{k+1}|I_k, u_k, z_{k+1}) = 
\begin{cases} 
p(x_{k+1}|I_k, u_k, z_{k+1}) \\
p(z_{k+1}|I_k, u_k) \\
p(x_{k+1}|I_k, u_k)p(z_{k+1}|I_k, u_k, x_{k+1}) \\
\int_{-\infty}^{\infty} p(x_{k+1}|I_k, u_k)p(z_{k+1}|I_k, u_k, x_{k+1})dx_{k+1}
\end{cases}
\]
In this expression, all the probability densities appearing in the RHS may be expressed in terms of 
\( p(x_k | I_k), u_k, \) and \( z_{k+1} \).

In particular:

- The density \( p(x_{k+1} | I_k, u_k) \) may be expressed through \( p(x_k | I_k), u_k, \) and the system equation \( x_{k+1} = f_k(x_k, u_k, w_k) \) using the given density \( p(w_k | x_k, u_k) \) and the relation
  \[
  p(w_k | x_k, u_k) = \int_{-\infty}^{\infty} p(x_k | I_k) p(w_k | x_k, u_k) \, dx_k.
  \]
- The density \( p(z_{k+1} | I_k, u_k, x_{k+1}) \) is expressed through the measurement equation \( z_{k+1} = h_{k+1}(x_{k+1}, u_k, v_{k+1}) \) using the densities \( p(x_k | I_k), p(w_k | x_k, u_k), p(v_{k+1} | x_k, u_k, w_k) \).

Now, we give an example for the finite space set case.

**Example 4.2.1 (A search problem)**

- At each period, decide to search or not search a site that may contain a treasure.
- If we search and treasure is present, we find it w.p. \( \beta \) and remove it from the site.
- State \( x_k \) (unobservable at the beginning of period \( k \)): Treasure is present or not.
- Control \( u_k \): search or not search.
- If the site is searched in period \( k \), the observation \( z_{k+1} \) takes two values: treasure found or not.
  If site is not searched, the value of \( z_{k+1} \) is irrelevant.
- Denote \( p_k \): probability a treasure is present at the beginning of period \( k \).

The probability evolves according to the recursion:

\[
p_{k+1} = \begin{cases} 
p_k & \text{if site is not searched at time } k \\
0 & \text{if the site is searched and a treasure is found (and removed)} \\
\frac{p_k (1-\beta)}{p_k (1-\beta) + 1 - p_k} & \text{if the site is searched but no treasure is found}
\end{cases}
\]

For the third case:

- Numerator \( p_k (1-\beta) \): It is the \( k \)th period probability that the treasure is present and the search is unsuccessful.
- Denominator \( p_k (1-\beta) + 1 - p_k \): Probability of an unsuccessful search, when the treasure is either there or not.

- The recursion for \( p_{k+1} \) is a special case of (4.3.4).

**4.3 Sufficient Statistics**

In this section, we continue investigating the conditional state distribution as a sufficient statistic for problems with imperfect state information.
4.3.1 Conditional state distribution: Review of basics

- Recall the important sufficient statistic conditional probability distribution of the state $x_k$ given the information vector $I_k$, i.e.,

$$S_k(I_k) = P_{x_k | I_k}$$

For this case, we need an extra assumption: The probability distribution of the observation disturbance $v_{k+1}$ depends explicitly only on the immediate preceding $x_k, u_k$ and $w_k$ and not on earlier ones, which gives the system a Markovian flavor.

- It turns out that $P_{x_k | I_k}$ is generated recursively by a dynamic system (estimator) of the form

$$P_{x_{k+1} | I_{k+1}} = \Phi_k(P_{x_k | I_k}, u_k, z_{k+1}), \quad (4.3.1)$$

for a suitable function $\Phi_k$ determined from the data of the problem.

- We have already proven that if function $\Phi_k$ in equation (4.3.1) exists, then we can solve the DP algorithm.

- The representation of the optimal policy as a sequence of functions of $P_{x_k | I_k}$, i.e.,

$$\mu_k(I_k) = \tilde{\mu}_k(P_{x_k | I_k}), \quad k = 0, 1, \ldots, N - 1,$$

is conceptually very useful. It provides a decomposition of the optimal controller in two parts:

1. An estimator, which uses at time $k$ the measurement $z_k$ and the control $u_{k-1}$ to generate the probability distribution $P_{x_k | I_k}$.
2. An actuator, which generates a control input to the system as a function of the probability distribution $P_{x_k | I_k}$.

- The DP algorithm can be written as:

$$\tilde{J}_{N-1}(P_{x_{N-1} | I_{N-1}}) = \min_{u_{N-1} \in U_{N-1}} \mathbb{E}_{x_{N-1}, w_{N-1}} \left[ g_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1})) \right. \left. + g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) | I_{N-1}, u_{N-1} \right] \quad (4.3.2)$$

and for $k = 0, 1, \ldots, N - 2$,

$$\tilde{J}_k(P_{x_k | I_k}) = \min_{u_k \in U_k} \mathbb{E}_{x_k, u_k, z_{k+1}} \left[ g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(\Phi_k(P_{x_k | I_k}, u_k, z_{k+1})) | I_k, u_k \right], \quad (4.3.3)$$

where $P_{x_k | I_k}$ plays the role of the state, and

$$P_{x_{k+1} | I_{k+1}} = \Phi_k(P_{x_k | I_k}, u_k, z_{k+1}) \quad (4.3.4)$$

is the system equation. Here, the role of control is played by $u_k$, and the role of the disturbance is played by $z_{k+1}$.

Example 4.3.1 (A search problem Revisited)

- At each period, decide to search or not search a site that may contain a treasure.
• If we search and the treasure is present, we find it w.p. \( \beta \) and remove it from the site.

• State \( x_k \) (unobservable at the beginning of period \( k \)): Treasure is present or not.

• Control \( u_k \): search or not search.

• Basic costs: Treasure’s worth is \( V \), and search cost is \( C \).

• If the site is searched in period \( k \), the observation \( z_{k+1} \) takes one of two values: treasure found or not.

  If site is not searched, the value of \( z_{k+1} \) is irrelevant.

• Denote \( p_k \): probability that the treasure is present at the beginning of period \( k \).

  The probability evolves according to the recursion:

  \[
p_{k+1} = \begin{cases} 
  p_k & \text{if site is not searched at time } k \\
  0 & \text{if the site is searched and a treasure is found (and removed)} \\
  \frac{p_k(1-\beta)}{p_k(1-\beta) + 1 - p_k} & \text{if the site is searched but no treasure is found}
  \end{cases}
  \]

  For the third case:

  – Numerator \( p_k(1 - \beta) \): It is the \( k \)th period probability that the treasure is present and the search is unsuccessful.

  – Denominator \( p_k(1 - \beta) + 1 - p_k \): Probability of an unsuccessful search, when the treasure is either there or not.

• The recursion for \( p_{k+1} \) is a special case of (4.3.4).

• Assume that once we decide not to search in a period, we cannot search at future times.

• The DP algorithm is

  \[
  \bar{J}_N(p_N) = 0,
  \]

  and for \( k = 0, 1, \ldots, N - 1 \),

  \[
  \bar{J}_k(p_k) = \max \left\{ 0, -C + \frac{p_k \beta V}{\text{reward for search & find}} + \frac{(1 - p_k \beta)}{\text{prob. of search & not find}} \bar{J}_{k+1} \left( \frac{p_k(1 - \beta)}{p_k(1 - \beta) + 1 - p_k} \right) \right\}
  \]

• It can be shown by induction that the functions \( \bar{J}_k(p_k) \) satisfy

  \[
  \bar{J}_k(p_k) = 0, \quad \forall p_k \leq \frac{C}{\beta V}
  \]

  Furthermore, it is optimal to search at period \( k \) if and only if

  \[
  \frac{p_k \beta V}{\text{expected reward from search}} \geq \frac{C}{\text{cost of search}} \quad \Box
  \]
4.3.2 Finite-state systems

- Suppose the system is a finite-state Markov chain with states 1, \ldots, \(n\).
- Then, the conditional probability distribution \(P_{x_k|I_k}\) is an \(n\)-dimensional vector
  \[
  (\mathbb{P}(x_k = 1|I_k), \mathbb{P}(x_k = 2|I_k), \ldots, \mathbb{P}(x_k = n|I_k)).
  \]
- When a control \(u \in U\) is applied (\(U\) finite), the system moves from state \(i\) to state \(j\) w.p. \(p_{ij}(u)\). Note that the real system state transition is only driven by the control \(u\) applied at each stage.
- There is a finite number of possible observation outcomes \(z^1, z^2, \ldots, z^q\). The probability of occurrence of \(z^\theta\), given that the current state is \(x_k = j\) and the preceding control was \(u_{k-1}\), is denoted by \(\mathbb{P}(z_k = z^\theta|u_{k-1}, x_k = j) \triangleq r_j(u_{k-1}, z^\theta), \theta = 1, \ldots, q\).
- The information available to the controller at stage \(k\) is
  \[I_k(z_1, \ldots, z_k, u_0, \ldots, u_{k-1}).\]
- Following the observation \(z_k\), a control \(u_k\) is applied, and a cost \(g(x_k, u_k)\) is incurred.
- The terminal cost at stage \(N\) for being in state \(x\) is \(G(x)\).
- Objective: Minimize the expected cumulative cost incurred over \(N\) stages.

We can reformulate the problem as one with imperfect state information. The objective is to control the column vector of conditional probabilities

\[p_k = (p^0_k, \ldots, p^n_k)^\prime,\]

where

\[p^i_k = \mathbb{P}(x_k = i|I_k), \quad i = 1, 2, \ldots, n.\]

We refer to \(p_k\) as the belief state. It evolves according to

\[p_{k+1} = \Phi_k(p_k, u_k, z_{k+1}),\]

where the function \(\Phi_k\) is an estimator that given the sufficient statistic \(p_k\) provides the new sufficient statistic \(p_{k+1}\). The initial belief \(p_0\) is given.

The conditional probabilities can be updated according to the Bayesian updating rule

\[
p^j_{k+1} = \frac{\mathbb{P}(x_{k+1} = j|I_{k+1})}{\mathbb{P}(z_{k+1}|I_k, u_k)}
= \frac{\mathbb{P}(x_{k+1} = j|z_0, \ldots, z_{k+1}, u_0, \ldots, u_k)}{\mathbb{P}(z_{k+1}|I_k, u_k)}
= \frac{\mathbb{P}(x_{k+1} = j|z, z_{k+1}|I_k, u_k)}{\mathbb{P}(z_{k+1}|I_k, u_k)}
= \frac{\sum_{i=1}^n \mathbb{P}(x_{k+1} = j|I_k) \mathbb{P}(x_k = i|I_k) \mathbb{P}(x_{k+1} = j|u_k, x_k = i)}{\sum_{j=1}^n \sum_{i=1}^n \mathbb{P}(x_k = i|I_k) \mathbb{P}(x_{k+1} = j|u_k, x_k = i)}
= \frac{\sum_{i=1}^n \sum_{j=1}^n p^i_k p_{ij}(u_k) r_j(u_k, z_{k+1})}{\sum_{i=1}^n \sum_{j=1}^n p^i_k p_{is}(u_k) r_s(u_k, z_{k+1})},
\]
In vector form, we have
\[ p^j_{k+1} = \frac{r_j(u_k, z_{k+1})[P(u_k)'p_k]^j}{\sum_{s=1}^{n} r_s(u_k, z_{k+1})[P(u_k)'p_k]^s}, \quad j = 1, \ldots, n, \]  
(4.3.5)
where \( P(u_k) \) is the \( n \times n \) transition probability matrix formed by \( p_{ij}(u_k) \), and \( [P(u_k)'p_k]^j \) is the \( j \)th component of vector \( [P(u_k)'p_k] \).

The corresponding DP algorithm (4.3.2)-(4.3.3) has the specific form
\[ J_k(p_k) = \min_{u_k \in U} \left\{ p'_k g(u_k) + E_{z_{k+1}} \left[ J_{k+1}(\Phi(p_k, u_k, z_{k+1}))|p_k, u_k \right] \right\}, \quad k = 0, \ldots, N - 1, \]  
(4.3.6)
where \( g(u_k) \) is the column vector with components \( g(1, u_k), \ldots, g(n, u_k) \), and \( p'_k g(u_k) \) is the expected stage cost.

The algorithm starts at stage \( N \) with
\[ J_N(p_N) = p'_N G, \]  
where \( G \) is the column vector with components the terminal costs \( G(i), i = 1, \ldots, n \), and proceeds backwards.

It turns out that the cost-to-go functions \( J_k \) in the DP algorithm are piecewise linear and concave. A consequence of this fact is that \( J_k \) can be characterized by a finite set of scalars. Still, however, for a fixed \( k \), the number of these scalars can increase fast with \( N \), and there may be no computationally efficient way to solve the problem.

**Example 4.3.2 (Machine repair revisited)**
Consider again the machine repair problem, whose setting is included below:

- A machine can be in one of two unobservable states (i.e., \( n = 2 \)): \( \bar{P} \) (bad state) and \( P \) (good state).
- State space: \( \{\bar{P}, P\} \), where for the indexing: State 1 is \( \bar{P} \), and state 2 is \( P \).
- Number of periods: \( N = 2 \)
- At the end of each period, the machine is inspected with two possible inspection outcomes: \( G \) (probably good state), \( B \) (probably bad state)
- Control space: actions after each inspection, which could be either
  - \( C \) : continue operation of the machine; or
  - \( S \) : stop, diagnose its state and if it is in bad state \( \bar{P} \), repair.
- Cost per stage: \( g(\bar{P}, C) = 2; \quad g(P, C) = 0; \quad g(\bar{P}, S) = 1; \quad g(P, S) = 1 \), or in vector form:
\[ g(C) = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad g(S) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \]
• Total cost: \( g(x_0, u_0) + g(x_1, u_1) \) (assume zero terminal cost)

• Let \( x_0, x_1 \) be the state of the machine at the end of each period

• Distribution of initial state: \( P(x_0 = P) = \frac{1}{3}, \ P(x_0 = P) = \frac{2}{3} \)

• Assume that we start with a machine in good state, i.e., \( x_{-1} = P \)

• System equation:

\[
x_{k+1} = w_k, \quad k = 0, 1
\]

where the transition probabilities are given by

In matrix form, following the aforementioned indexing of the states, transition probabilities can be expressed as

\[
P(C) = \begin{pmatrix} 1 & 0 \\ 1/3 & 2/3 \end{pmatrix}; \quad P(S) = \begin{pmatrix} 1/3 & 2/3 \\ 1/3 & 2/3 \end{pmatrix}.
\]

• Note that we do not have perfect state information, since the inspection does not reveal the state of the machine with certainty. Rather, the result of each inspection may be viewed as a noisy assessment of the system state.

Result of inspections: \( z_k = v_k, \quad k = 0, 1; \quad v_k \in \{B, G\} \)
The inspection results can be described by the following definitions:

\[ r_1(S, G) = \mathbb{P}(z_{k+1} = G|u_k = S, x_{k+1} = \bar{P}) = \frac{1}{4} = r_1(C, G), \]

\[ r_1(S, B) = \mathbb{P}(z_{k+1} = B|u_k = S, x_{k+1} = \bar{P}) = \frac{3}{4} = r_1(C, B), \]

\[ r_2(S, G) = \mathbb{P}(z_{k+1} = G|u_k = S, x_{k+1} = P) = \frac{3}{4} = r_2(C, G), \]

\[ r_2(S, B) = \mathbb{P}(z_{k+1} = B|u_k = S, x_{k+1} = P) = \frac{1}{4} = r_2(C, B). \]

Note that in this case, the observation \( z_{k+1} \) does not depend on the control \( u_k \), but just on the state \( x_{k+1} \).

Define the belief state \( p_0 \) as the 2-dimensional vector with components:

\[ p_0^1 \triangleq \mathbb{P}(x_0 = \bar{P}|I_0), \quad p_0^2 \triangleq \mathbb{P}(x_0 = P|I_0) = 1 - p_0^1. \]

Similarly, define the belief state \( p_1 \) with coordinates

\[ p_1^1 \triangleq \mathbb{P}(x_1 = \bar{P}|I_1), \quad p_1^2 \triangleq \mathbb{P}(x_1 = P|I_1) = 1 - p_1^1, \]

where the evolution of the beliefs is driven by the estimator

\[ p_1 = \Phi_0(p_0, u_0, z_1). \]

We will use equation (4.3.5) to compute \( p_1 \) given \( p_0 \), but first we calculate the matrix products \( P(u_0)'p_0 \), for \( u_0 \in \{S, C\} \):

\[ P(S)'p_0 = \begin{pmatrix} 1/3 & 1/3 \\ 2/3 & 2/3 \end{pmatrix} \begin{pmatrix} p_0^1 \\ p_0^2 \end{pmatrix} = \begin{pmatrix} 1/3 + 1/3 \\ 2/3 + 2/3 \end{pmatrix} = \begin{pmatrix} 2/3 \\ 4/3 \end{pmatrix}, \] (4.3.7)

and

\[ P(C)'p_0 = \begin{pmatrix} 1 & 1/3 \\ 0 & 2/3 \end{pmatrix} \begin{pmatrix} p_0^1 \\ p_0^2 \end{pmatrix} = \begin{pmatrix} p_0^1 + 1/3p_0^2 \\ 2/3p_0^2 \end{pmatrix} = \begin{pmatrix} p_0^1 + 2/3p_0^1 \\ 2/3 - 2/3p_0^1 \end{pmatrix}. \] (4.3.8)

Now, using equation (4.3.5) for state \( j = 1 \) (i.e., for state \( \bar{P} \)), we get

- For \( u_0 = S, z_1 = G \):

\[ p_1^1 = \frac{r_1(S, G)[P(S)'p_0]_1}{r_1(S, G)[P(S)'p_0]_1 + r_2(S, G)[P(S)'p_0]_2} = \frac{1/4 \times 1}{3/4 \times 1/3 + 2/3 \times 2/3} = \frac{1}{7}. \]

- For \( u_0 = S, z_1 = B \):

\[ p_1^1 = \frac{r_1(S, B)[P(S)'p_0]_1}{r_1(S, B)[P(S)'p_0]_1 + r_2(S, B)[P(S)'p_0]_2} = \frac{3/4 \times 1}{4/3 \times 1/3 + 4/3 \times 2/3} = \frac{3}{5}. \]

- For \( u_0 = C, z_1 = G \):

\[ p_1^1 = \frac{r_1(C, G)[P(C)'p_0]_1}{r_1(C, G)[P(C)'p_0]_1 + r_2(C, G)[P(C)'p_0]_2} = \frac{1/4 \times (1/3 + 2/3p_0^1)}{3/4 \times (2/3p_0^1) + 2/3 \times (2/3 - 2/3p_0^1)} = \frac{1 + 2p_0^1}{7 - 4p_0^1}. \]
For \( u_0 = C, z_1 = B:\)

\[
p_1^1 = \frac{r_1(C, B)[P(C)p_0]}{r_1(C, B)[P(C)p_0]_1 + r_2(C, B)[P(C)p_0]_2} = \frac{\frac{3}{4} \times \left( \frac{1}{3} + \frac{2}{3}p_0 \right)}{\frac{5}{4} \times \left( \frac{1}{3} + \frac{2}{3}p_0 \right) + \frac{1}{4} \times \left( \frac{2}{3} - \frac{2}{3}p_0 \right)} = \frac{3 + 6p_0^1}{5 + 4p_0^1}.
\]

In summary, we get

\[
p_1^1 = [\Phi_0(p_0, u_0, z_1)]_1 = \begin{cases} 
\frac{1}{5} & \text{if } u_0 = S, \ z_1 = G, \\
\frac{1 + 2p_0^1}{7 - 4p_0^1} & \text{if } u_0 = S, \ z_1 = B, \\
\frac{1}{3 + 6p_0^1} & \text{if } u_0 = C, \ z_1 = G, \\
\frac{3 + 6p_0^1}{5 + 4p_0^1} & \text{if } u_0 = C, \ z_1 = B,
\end{cases}
\]

where \( p_0^2 = 1 - p_0^1 \) and \( p_0^1 = 1 - p_1^1 \).

The DP algorithm (4.3.6) may be written as:

\[
\bar{J}_2(p_2) = 0 \quad \text{(i.e., zero terminal cost)},
\]

and

\[
\bar{J}_1(p_1) = \min_{u_1 \in \{S, C\}} \{ p_1^1 g(u_1) \} = \min \left\{ \begin{pmatrix} p_1^1 + p_0^2 \\ p_1^1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} p_1^1 + p_0^2 \\ p_1^1 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \end{pmatrix} \right\}_{u_1 = S}^{u_1 = C}.
\]

This minimization yields

\[
\bar{p}_1^*(p_1) = \begin{cases} 
C & \text{if } p_1^1 \leq \frac{1}{2}, \\
S & \text{if } p_1^1 > \frac{1}{2}.
\end{cases}
\]

For \( \bar{J}_0(p_0) \) we have

\[
\bar{J}_0(p_0) = \min_{u_0 \in \{C, S\}} \left\{ p_1^1 g(u_0) + E_{z_1} \left[ \bar{J}_1(\Phi_0(p_0, u_0, z_1)) | p_0, u_0 \right] \right\}
\]

\[
= \min \left\{ 2p_0^1 + \mathbb{P}(z_1 = G | I_0, C)\bar{J}_1(\Phi_0(p_0, C, G)) + \mathbb{P}(z_1 = B | I_0, C)\bar{J}_1(\Phi_0(p_0, C, B)) \right\}_{u_0 = C}
\]

\[
\left\{ \left( p_0^1 + p_0^2 \right) + \mathbb{P}(z_1 = G | I_0, S)\bar{J}_1(\Phi_0(p_0, S, G)) + \mathbb{P}(z_1 = B | I_0, S)\bar{J}_1(\Phi_0(p_0, S, B)) \right\}_{u_0 = S}
\]

The probabilities here may be expressed in terms of \( p_0 \) by using the expression in the denominator of (4.3.5); that is:

\[
\mathbb{P}(z_{k+1}|I_k, u_k) = \sum_{s=1}^{n} \sum_{i=1}^{n} \mathbb{P}(x_k = i|I_k)\mathbb{P}(x_{k+1} = s|x_k = i, u_k)\mathbb{P}(z_{k+1}|u_k, x_{k+1} = s)
\]

\[
= \sum_{s=1}^{n} \sum_{i=1}^{n} p_i^k p_{is}(u_k)r_s(u_k, z_{k+1})
\]

\[
= \sum_{s=1}^{n} r_s(u_k, z_{k+1})[P(u_k)'p_k]_s.
\]
In our case:

\[
P(z_1 = G|I_0, u_0 = C) = r_1(C, G)[P(C)'p_0]_1 + r_2(C, G)[P(C)'p_0]_2
\]

\[
= \frac{1}{4} \times \left( \frac{1}{3} + \frac{2}{3}p_0^1 \right) + \frac{3}{4} \times \left( \frac{2}{3} - \frac{2}{3}p_0^1 \right)
\]

\[
= \frac{7 - 4p_0^1}{12}.
\]

Similarly, we obtain:

\[
P(z_1 = B|I_0, C) = \frac{5 + 4p_0^1}{12}, \quad P(z_1 = G|I_0, S) = \frac{7}{12}, \quad P(z_1 = B|I_0, S) = \frac{5}{12}.
\]

Using these values we have

\[
\bar{J}_0(p_0) = \min \left\{ 2p_0^1 + \frac{7 - 4p_0^1}{12} \hat{J}_1 \left( \frac{1 + 2p_0^1}{7 - 4p_0^1}, 1 - p_0^1 \right), \frac{5 + 4p_0^1}{12} \hat{J}_1 \left( \frac{3 + 6p_0^1}{5 + 4p_0^1}, 1 - p_0^1 \right), \right. \right.
\]

\[
\left. \left. \left. \left. 1 + \frac{7 - 4p_0^1}{12} \hat{J}_1 \left( \frac{1}{7 - 4p_0^1}, \frac{6}{7} \right) + \frac{5}{12} \hat{J}_1 \left( \frac{3}{5}, \frac{2}{5} \right) \right\} \right. \right. \right. \right.
\]

By substitution of \( \hat{J}_1(p_1) \) and after some algebra we obtain

\[
\bar{J}_0(p_0) = \left\{ \begin{array}{ll}
\frac{19}{12} & \text{if } \frac{3}{8} \leq p_0^1 \leq 1, \\
\frac{7 + 32p_0^1}{12} & \text{if } 0 \leq p_0^1 \leq \frac{3}{8},
\end{array} \right.
\]

and an optimal control for the first stage

\[
\hat{\mu}_0^*(p_0) = \left\{ \begin{array}{ll}
C & \text{if } p_0^1 \leq \frac{3}{8}, \\
S & \text{if } p_0^1 > \frac{3}{8}.
\end{array} \right.
\]

Also, we know that \( P(z_0 = G) = \frac{7}{12}, \) and \( P(z_0 = B) = \frac{5}{12}. \) In addition, we can establish the initial value for \( p_0^1 \) according to the value of \( I_0 \) (i.e., \( z_0 \)):

\[
P(x_0 = P|z_0 = G) = \frac{P(x_0 = P, z_0 = G)}{P(z_0 = G)} = \frac{\frac{1}{3} \times \frac{4}{7}}{\frac{7}{12}} = \frac{1}{7},
\]

and

\[
P(x_0 = P|z_0 = B) = \frac{P(x_0 = P, z_0 = B)}{P(z_0 = B)} = \frac{\frac{1}{3} \times \frac{3}{4}}{\frac{5}{12}} = \frac{3}{5},
\]

so that the formula

\[
J^* = E_{z_0} [\bar{J}_0(P_x|z_0)] = \frac{7}{12} \bar{J}_0 \left( \frac{1}{7}, \frac{6}{7} \right) + \frac{5}{12} \bar{J}_0 \left( \frac{3}{5}, \frac{2}{5} \right) = \frac{176}{144}
\]

yields the same optimal cost as the one obtained above by means of the general DP algorithm for problems with imperfect state information.

Observe also that the functions \( \bar{J}_k \) are linear in this case; recall that we had said that in general they are piecewise linear. □
4.4 Exercises

Exercise 4.4.1 Take the linear system and measurement equation for the LQ-system with imperfect state information. Consider the problem of finding a policy \( \{\mu^*_0(I_0), \ldots, \mu^*_{N-1}(I_{N-1})\} \) that minimizes the quadratic cost

\[
E \left[ x_N' Q x_N + \sum_{k=0}^{N-1} u_k' R_k u_k \right]
\]

Assume, however, that the random vectors \( x_0, w_0, \ldots, w_{N-1}, v_0, \ldots, v_{N-1} \) are correlated and have a given joint probability distribution, and finite first and second moments. Show that the optimal policy is given by

\[
\mu^*_k(I_k) = L_k E[y_k | I_k],
\]

where the gain matrices \( L_k \) are obtained from the algorithm

\[
L_k = -(B_k' K_{k+1} B_k + R_k)^{-1} B_k' K_{k+1} A_k,
\]

\[
K_N = Q,
\]

\[
K_k = A_k' (K_{k+1} - K_{k+1} B_k (B_k' K_{k+1} B_k + R_k)^{-1} B_k' K_{k+1}) A_k,
\]

and the vectors \( y_k \) are given by \( y_N = x_N \), and

\[
y_k = x_k + A_k^{-1} w_k + A_k^{-1} A_{k+1}^{-1} w_{k+1} + \cdots + A_{N-1}^{-1} w_{N-1}, \quad k = 0, \ldots, N - 1.
\]

(assuming the matrices \( A_0, A_1, \ldots, A_{N-1} \) are invertible).

Hint: Show that the cost can be written as

\[
E \left[ y_0' K_0 y_0 + \sum_{k=0}^{N-1} (u_k - L_k y_k)' P_k (u_k - L_k y_k) \right],
\]

where \( P_k = B_k' K_{k+1} B_k + R_k \).

Exercise 4.4.2 Consider the scalar, imperfect state information system

\[
x_{k+1} = x_k + u_k + w_k,
\]

\[
z_k = x_k + v_k,
\]

where we assume that the initial state \( x_0 \), and the disturbances \( w_k \) and \( v_k \) are all independent. Let the cost be

\[
E \left[ x_N^2 + \sum_{k=0}^{N-1} (x_k^2 + u_k^2) \right],
\]

and let the given probability distributions be

\[
\mathbb{P}(x_0 = 2) = 1/2, \quad \mathbb{P}(w_k = 1) = 1/2, \quad \mathbb{P}(v_k = 1/4) = 1/2,
\]

\[
\mathbb{P}(x_0 = -2) = 1/2, \quad \mathbb{P}(w_k = -1) = 1/2, \quad \mathbb{P}(v_k = -1/4) = 1/2.
\]

(a) Show that this problem could be transformed in a perfect information problem, where first we infer the value of \( x_0 \), and then we sequentially compute the values \( x_1, \ldots, x_N \). Determine the optimal policy. Hint: For this problem, \( x_k \) can be determined from \( x_{k-1}, u_{k-1}, \) and \( z_k \).
(b) Determine the policy that is identical to the optimal except that it uses a linear least square estimator of $x_k$ given $I_k$ in place of $\mathbb{E}[x_k|I_k]$

**Exercise 4.4.3** A linear system with Gaussian disturbances and Gaussian initial state

$$x_{k+1} = Ax_k + Bx_k + w_k,$$

is to be controlled so as to minimize a quadratic cost similar to that discussed above. The difference is that the controller has the option of choosing at each time $k$ one of two types of measurement equations for the next stage $k + 1$:

- First type: $z_{k+1} = C^1 x_{k+1} + v^1_{k+1}$,
- Second type: $z_{k+1} = C^2 x_{k+1} + v^2_{k+1}$.

Here, $C^1$ and $C^2$ are given matrices of appropriate dimension, and $\{v^1_k\}$ and $\{v^2_k\}$ are zero-mean, independent, random sequences with given finite covariances that do not depend on $x_0$ and $\{w_k\}$. There is a cost $g_1$ (or $g_2$) each time a measurement of type 1 (or type 2) is taken. The problem is to find the optimal control and measurement selection policy that minimizes the expected value of the sum of the quadratic cost

$$x_N'Qx_N + \sum_{k=0}^{N-1} (x_k'Qx_k + u_k'R u_k)$$

and the total measurement cost. Assume for convenience that $N = 2$ and that the first measurement $z_0$ is of type 1. Show that the optimal measurement selection at time $k = 0$ and $k = 1$ does not depend on the value of the information vectors $I_0$ and $I_1$, and can be determined a priori. Describe the nature of the optimal policy.

**Exercise 4.4.4** Consider a machine that can be in one of two states, good or bad. Suppose that the machine produces an item at the end of each period. The item produced is either good or bad depending on whether the machine is in good or bad state at the beginning of the corresponding period, respectively. We suppose that once the machine is in a bad state it remains in that state until it is replaced. If the machine is in a good state at the beginning of a certain period, then with probability $t$ it will be in the bad state at the end of the period. Once an item is produced, we may inspect the item at a cost $I$, or not inspect. If an inspected item is found to be bad, the machine is replaced with a machine in good state at a cost $R$. The cost for producing a bad item is $C > 0$. Write a DP algorithm for obtaining an optimal inspection policy assuming a machine is initially in good state and a horizon of $N$ periods. Then, solve the problem for $t = 0.2$, $I = 1$, $R = 3$, $C = 2$, and $N = 8$.

**Hint:** Define

$$x_k = \text{State at the beginning of the } k\text{th stage} \in \{\text{Good, Bad}\}$$

$$w_k = \text{State at the end of the } k\text{th stage before an action is taken}$$

$$u_k \in \{\text{Inspect, No inspect}\}$$

Take as information vector the stage at which the last inspection was made.
Exercise 4.4.5 A person is offered 2 to 1 odds in a coin-tossing game where he wins whenever a tail occurs. However, he suspects that the coin is biased and has an a priori probability distribution $F(p)$ for the probability $p$ that a head occurs at each toss. The problem is to find an optimal policy of deciding whether to continue or stop participating in the game given the outcomes of the game so far. A maximum of $N$ tossing is allowed. Indicate how such a policy can be found by means of DP. Specify the update rule for the belief about $p$.

*Hint:* Define the state as $n_k$, the number of heads observed in the first $k$ flips.
Chapter 5

Infinite Horizon Problems

5.1 Types of infinite horizon problems

- Setting similar to the basic finite horizon problem, but:
  - The number of stages is infinite.
  - The system is stationary.

- Simpler version: Assume finite number of states. (We will keep this assumption)

- Total cost problems: Minimize over all admissible policies $\pi$,

$$J_\pi(x_0) = \lim_{N \to \infty} \mathbb{E}_{w_0, w_1, \ldots} \left[ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right]$$

The value function $J_\pi(x_0)$ should be finite for at least some admissible policies $\pi$ and some initial states $x_0$.

Variants of total cost problems:

(a) Stochastic shortest path problems ($\alpha = 1$): It requires a cost free terminal state $t$ that is reached in finite time w.p.1.

(b) Discounted problems ($\alpha < 1$) with bounded cost per stage, i.e., $|g(x, u, w)| < M$.

Here, $J_\pi(x_0) < \text{decreasing geometric progression } \{\alpha^k M\}$.

(c) Discounted and non-discounted problems with unbounded cost per stage.

Here, $\alpha \leq 1$, but $|g(x, u, w)|$ could be $\infty$. Technically more challenging!

- Average cost problems (type (d)): Minimize over all admissible policies $\pi$,

$$J_\pi(x_0) = \lim_{N \to \infty} \frac{1}{N} \mathbb{E}_{w_0, w_1, \ldots} \left[ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k), w_k) \right]$$

The approach works even if $J_\pi(x_0)$ is infinite for every policy $\pi$ and initial state $x_0$. 

167
5.1.1 Preview of infinite horizon results

- Key issue: The relation between the infinite and finite horizon optimal cost-to-go functions.

- Illustration: Let $\alpha = 1$ and $J_N(x)$ denote the optimal cost of the $N$-stage problem, generated after $N$ iterations of the DP algorithm, starting from $J_0(x) = 0$, and proceeding with

  $$J_{k+1} = \min_{u \in U(x)} E_w [g(x, u, w) + J_k(f(x, u, w))]. \quad \forall x. \quad (5.1.1)$$

Typical results for total cost problems:

- Relation valuable from a computational viewpoint:

  $$J^*(x) = \lim_{N \to \infty} J_N(x), \quad \forall x. \quad (5.1.2)$$

  It holds for problems (a) and (b); some unusual exceptions for problems (c).

- The limiting form of the DP algorithm should hold for all states $x$,

  $$J^*(x) = \min_{u \in U(x)} E_w [g(x, u, w) + J^*(f(x, u, w))], \quad \forall x. \quad \text{(Bellman’s equation)}$$

- If $\mu(x)$ minimizes RHS in Bellman’s equation for each $x$, the policy $\pi = \{\mu, \mu, \ldots\}$ is optimal. This is true for most infinite horizon problems of interest (and in particular, for problems (a) and (b)).

5.1.2 Total cost problem formulation

- We assume an underlying system equation

  $$x_{k+1} = w_k.$$  

- At state $i$, the use of a control $u$ specifies the transition probability $p_{ij}(u)$ to the next state $j$.

- The control $u$ is constrained to take values in a given finite constraint set $U(i)$, where $i$ is the current state.

- We will assume a $k$th stage cost $g(x_k, u_k)$ for using control $u_k$ at stage $x_k$. If $\tilde{g}(i, u, j)$ is the cost of using $u$ at state $i$ and moving to state $j$, we use as cost-per-stage the expected cost $g(i, u)$ given by

  $$g(i, u) = \sum_j p_{ij}(u) \tilde{g}(i, u, j).$$

- The total expected cost associated with an initial state $i$ and a policy $\pi = \{\mu_0, \mu_1, \ldots\}$ is

  $$J_\pi(i) = \lim_{N \to \infty} E \left[ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k)) \bigg| x_0 = i \right],$$

  where $\alpha$ is a discount factor, with $0 < \alpha \leq 1$.

- Optimal cost from state $i$ is $J^*(i) = \min_{\pi} J_\pi(i)$. 

168
• Stationary policy: Admissible policy (i.e., \( \mu_k(x_k) \in U(x_k) \)) of the form

\[
\pi = \{ \mu, \mu, \ldots \},
\]

with corresponding cost function \( J_\mu(i) \).

The stationary policy \( \mu \) is optimal if

\[
J_\mu(i) = J^*_\mu(i) = \min_{\pi} J_\pi(i), \quad \forall i.
\]

5.2 Stochastic shortest path problems

• Assume there is no discounting (i.e., \( \alpha = 1 \)).

• Set of “normal states” \( \{1, 2, \ldots, n\} \).

• There is a special, cost-free, absorbing, terminal state \( t \). That is, \( p_{tt} = 1 \), and \( g(t, u) = 0 \) for all \( u \in U(t) \).

• Objective: Reach the terminal state with minimum expected cost.

• Assumption 5.2.1 There exists an integer \( m \) such that for every policy and initial state, there is a positive probability that the termination state \( t \) will be reached in at most \( m \) stages. Then for all \( \pi \), we have

\[
\rho_\pi = \mathbb{P}\{x_m \neq t|x_0 \neq t, \pi\} < 1
\]

That is, \( \mathbb{P}\{x_m = t|x_0 \neq t, \pi\} > 0 \).

• In terms of discrete-time Markov chains, Assumption 5.2.1 is claiming that \( t \) is accessible from any state \( i \).

• Remark: Assumption 5.2.1 is requiring that all policies are proper. A stationary policy is proper if when using it, there is a positive probability that the destination will be reached after at most \( n \) stages. Otherwise, it is improper.

However, the results to be presented can be proved under the following weaker conditions:

1. There exists at least one proper policy.
2. For every improper policy \( \pi \), the corresponding cost \( J_\pi(i) \) is \( \infty \) for at least one state \( i \).

• Note that the assumption implies that

\[
\mathbb{P}\{x_m \neq t|x_0 = i, \pi\} \leq \mathbb{P}\{x_m \neq t|x_0 \neq t, \pi\} = \rho_\pi < 1, \quad \forall i = 1, \ldots, n.
\]

• Let

\[
\rho = \max_\pi \rho_\pi.
\]

Since the number of controls available at each state is finite, the number of distinct \( m \)-stage policies is also finite. So, there must be only a finite number of values of \( \rho_\pi \), so that the max above is well defined (we do not need sup). Then,

\[
\mathbb{P}\{x_m \neq t|x_0 \neq t, \pi\} \leq \rho < 1.
\]
• For any $\pi$ and any initial state $i$, 
\[
P\{x_{2m} \neq t|x_0 = i, \pi\} = P\{x_{2m} \neq t|x_m \neq t, x_0 = i, \pi\} \times P\{x_m \neq t|x_0 = i, \pi\} \\
\leq P\{x_{2m} \neq t|x_m \neq t, \pi\} \times P\{x_m \neq t|x_0 \neq t, \pi\} \leq \rho^2,
\]
and similarly,
\[
P\{x_{km} \neq t|x_0 = i, \pi\} \leq \rho^k, \quad i = 1, \ldots, n.
\]

• So,
\[
|E [\text{cost between times } km \text{ and } (k+1)m - 1]| \leq m \rho^k \times \max_{u \in U(i)} |g(u, i)|, \quad \text{bound for each stage}
\]
and hence,
\[
|J_\pi(i)| \leq \sum_{k=0}^{\infty} m \rho^k \max_{u \in U(i)} |g(u, i)| = \frac{m}{1 - \rho} \max_{u \in U(i)} |g(u, i)|.
\]

• Key idea for the main result (to be presented below) is that the tail of the cost series vanishes, i.e.,
\[
\lim_{K \to \infty} \sum_{k=mK}^{\infty} E [g(x_k, \mu_k(x_k))] = 0.
\]
The reason is that $\lim_{K \to \infty} P\{x_{mK} \neq t|x_0 = i, \pi\} = 0$.

**Proposition 5.2.1** Under Assumption 5.2.1, the following hold for the stochastic shortest path problem:

(a) Given any initial conditions $J_0(1), \ldots, J_0(n)$, the sequence $J_k(i)$ generated by the DP iteration
\[
J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)J_k(j) \right\}, \quad \forall i,
\]
converges to the optimal cost $J^*(i)$.

(b) The optimal costs $J^*(1), \ldots, J^*(n)$ satisfy Bellman’s equation,
\[
J^*(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)J^*(j) \right\}, \quad i = 1, \ldots, n,
\]
and in fact they are the unique solution of this equation.

(c) For any stationary policy $\mu$, the costs $J_\mu(1), \ldots, J_\mu(n)$ are the unique solution of the equation
\[
J_\mu(i) = g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i))J_\mu(j), \quad i = 1, \ldots, n.
\]
Furthermore, given any initial conditions \( J_0(1), \ldots, J_0(n) \), the sequence \( J_k(i) \) generated by the DP iteration

\[
J_{k+1}(i) = g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i)) J_k(j), \quad i = 1, \ldots, n,
\]

converges to the cost \( J_\mu(i) \) for each \( i \).

(d) A stationary policy \( \mu \) is optimal if and only if for every state \( i \), \( \mu(i) \) attains the minimum in Bellman’s equation (5.2.3).

PROOF: Following the labeling of the proposition:

(a) For every possible integer \( K \), initial state \( x_0 \), and policy \( \pi = \{ \mu_0, \mu_1, \ldots \} \), we break down the cost \( J_\pi(x_0) \) as follows:

\[
J_\pi(x_0) = \lim_{N \to \infty} E \left[ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k)) \right] = E \left[ \sum_{k=0}^{mK-1} g(x_k, \mu_k(x_k)) \right] + \lim_{N \to \infty} E \left[ \sum_{k=mK}^{N-1} g(x_k, \mu_k(x_k)) \right] \quad (5.2.4)
\]

Let \( M \) be an upper bound on the cost of an \( m \)-stage cycle, assuming \( t \) is not reached during the cycle, i.e.,

\[
M = m \max_{i=1, \ldots, n} \max_{u \in U(i)} |g(i, u)|.
\]

Recall from (5.2.1) that

\[
\left| E[\text{cost during } K \text{th cycle, between stages } Km \text{ and } (K+1)m-1] \right| \leq M \rho^K, \quad (5.2.5)
\]

so that

\[
\left| \lim_{N \to \infty} E \left[ \sum_{k=mK}^{N-1} g(x_k, \mu_k(x_k)) \right] \right| \leq M \sum_{k=K}^{\infty} \rho^k = \frac{\rho^K M}{1 - \rho}. \quad (5.2.6)
\]

Also, denoting \( J_0(t) = 0 \), let us view \( J_0 \) as a terminal cost function. We will provide a bound for its expected value based on the current policy \( \pi \) applied over \( mK \) stages. Starting from \( x_0 \neq t \), \( J_0(x_{mK}) \) is the cost of reaching state \( x_{mK} \) in \( mK \) steps. So,

\[
\left| E[J_0(x_{mK})] \right| = \left| \sum_{i=1}^{n} \mathbb{P}\{x_{mK} = i|x_0 \neq t, \pi\} J_0(i) + \mathbb{P}\{x_{mK} = t|x_0 \neq t, \pi\} J_0(t) \right| \leq \rho^K \max_{i=1, \ldots, n} \left| J_0(i) \right| \quad (5.2.7)
\]
Now, we set the following bound (following equations (5.2.6) and (5.2.7)):

\[
\left| \mathbb{E}[J_0(x_{mK})] - \lim_{N \to \infty} \mathbb{E} \left[ \sum_{k=mK}^{N-1} g(x_k, \mu_k(x_k)) \right] \right| \leq \left| \mathbb{E}[J_0(x_{mK})] + \lim_{N \to \infty} \mathbb{E} \left[ \sum_{k=mK}^{N-1} g(x_k, \mu_k(x_k)) \right] \right|
\]

\[
\leq \rho^K \max_{i=1,\ldots,n} |J_0(i)| + M \sum_{k=K}^{\infty} \rho^k
\]

Taking the LHS above, and using (5.2.4), we have

\[
\left| \mathbb{E}[J_0(x_{mK})] - \lim_{N \to \infty} \mathbb{E} \left[ \sum_{k=mK}^{N-1} g(x_k, \mu_k(x_k)) \right] \right| = \left| \mathbb{E}[J_0(x_{mK})] + \mathbb{E} \left[ \sum_{k=0}^{mK-1} g(x_k, \mu_k(x_k)) \right] - J_\pi(x_0) \right|
\]

Then, we get the bounds

\[-\rho^K \max_{i=1,\ldots,n} |J_0(i)| + J^*(x_0) - \frac{\rho^K M}{1-\rho} \leq \mathbb{E}[J_0(x_{mK})] + \mathbb{E} \left[ \sum_{k=0}^{mK-1} g(x_k, \mu_k(x_k)) \right] \leq \rho^K \max_{i=1,\ldots,n} |J_0(i)| + J^*(x_0) + \frac{\rho^K M}{1-\rho}.
\]

(5.2.8)

Note that

- The expected value of the middle term above is the \(mK\)-stage cost of policy \(\pi\), starting from state \(x_0\), with terminal cost \(J_0(x_{mK})\).
- The min of this \(mK\)-stage cost over all \(\pi\) is equal to the value \(J_{mK}(x_0)\), which is generated by the DP recursion (5.2.2) after \(mK\) iterations.

Thus, taking the min over \(\pi\) in equation (5.2.8), we obtain for all \(x_0\) and \(K\),

\[-\rho^K \max_{i=1,\ldots,n} |J_0(i)| + J^*(x_0) - \frac{\rho^K M}{1-\rho} \leq J_{mK}(x_0) \leq \rho^K \max_{i=1,\ldots,n} |J_0(i)| + J^*(x_0) + \frac{\rho^K M}{1-\rho}.
\]

(5.2.9)

When taking limit as \(K \to \infty\), the terms in LHS and RHS involving \(\rho^K \to 0\), leading to

\[\lim_{K \to \infty} J_{mK}(x_0) = J^*(x_0), \quad \forall x_0.\]

Since from (5.2.5)

\[|J_{mK+q}(x_0) - J_{mK}(x_0)| \leq \rho^K M, \quad q = 0, \ldots, m-1,\]

we see that for \(q = 0, \ldots, m-1,\)

\[-\rho^K M + J_{mK}(x_0) \leq J_{mK+q}(x_0) \leq J_{mK}(x_0) + \rho^K M.\]

Taking limit as \(K \to \infty\), we get

\[\lim_{K \to \infty} (\rho^K M + J_{mK}(x_0)) = J^*(x_0).\]

Thus, for any \(q = 0, \ldots, m-1,\)

\[\lim_{K \to \infty} J_{mK+q}(x_0) = J^*(x_0),\]

and hence,

\[\lim_{k \to \infty} J_k(x_0) = J^*(x_0).\]
(b) Existence: By taking limit as \( k \to \infty \) in the DP iteration (5.2.2), and using the convergence result of part (a) \( \Rightarrow J^*(1), \ldots, J^*(n) \) satisfy Bellman’s equation.

Uniqueness: If \( J(1), \ldots, J(n) \) satisfy Bellman’s equation, then the DP iteration (5.2.2) starting from \( J(1), \ldots, J(n) \) just replicates \( J(1), \ldots, J(n) \). Then, from the convergence result of part (a), \( J(i) = J^*(i), \ i = 1, \ldots, n \).

(c) Given stationary policy \( \mu \), redefine the control constraint sets to be \( \tilde{U}(i) = \{ \mu(i) \} \) instead of \( U(i) \). From part (b), we then obtain that \( J_{\mu}(1), \ldots, J_{\mu}(n) \) solve uniquely Bellman’s equation for this redefined problem; i.e.,

\[
J_{\mu}(i) = g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i))J_{\mu}(j), \quad i = 1, \ldots, n,
\]

and from part (a) it follows that the corresponding DP iteration converges to \( J_{\mu}(i) \).

(d) We have that \( \mu(i) \) attains the minimum in equation (5.2.3) if and only if we have

\[
J^*(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)J^*(j) \right\} = g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i))J^*(j), \quad i = 1, \ldots, n.
\]

This equation and part (c) imply that \( J_{\mu}(i) = J^*(i) \) for all \( i \). Conversely, if \( J_{\mu}(i) = J^*(i) \) for all \( i \), parts (b) and (c) imply the above equation.

This completes the proof of the four parts of the proposition.

\[ \square \]

Observation: Part (c) provides a a way to compute \( J_{\mu}(i), i = 1, \ldots, n \), for a given stationary policy \( \mu \), but the computation is substantial for large \( n \) (of order \( O(n^3) \)).

Example 5.2.1 (Minimizing Expected Time to Termination)

- Let \( g(i, u) = 1, \ \forall i = 1, \ldots, n, \ u \in U(i) \).

- Under our assumptions, the costs \( J^*(i) \) uniquely solve Bellman’s equation, which has the form

\[
J^*(i) = \min_{u \in U(i)} \left\{ 1 + \sum_{j=1}^{n} p_{ij}(u)J^*(j) \right\}, \quad i = 1, \ldots, n.
\]

- In the special case where there is only one control at each state, \( J^*(i) \) is the mean first passage time from \( i \) to \( t \). These times, denoted \( m_i \), are the unique solution of the equations

\[
m_i = 1 + \sum_{j=1}^{n} p_{ij}m_j, \quad i = 1, \ldots, n.
\]

Recall that in a discrete-time Markov chain, if there is only one recurrent class and \( t \) is a state of that class (in our case, the only recurrent class is given by \{\( t \}\}), the mean first passage times from \( i \) to \( t \) are the unique solution to the previous system of linear equations. \( \square \)
Example 5.2.2 (Spider and a fly)

• A spider and a fly move along a straight line.

• At the beginning of each period, the slider knows the position of the fly.

• The fly moves one unit to the left w.p. $p$, one unit to the right w.p. $p$, and stays where it is w.p. $1 - 2p$.

• The spider moves one unit towards the fly if its distance from the fly is more than one unit.

• If the spider is one unit away from the fly, it will either move one unit towards the fly or stay where it is.

• If the spider and the fly land in the same position, the spider captures the fly.

• The spider’s objective is to capture the fly in minimum expected time.

• The initial distance between the spider and the fly is $n$.

• This is a stochastic shortest path problem with state $i =$ distance between spider and fly, with $i = 1, \ldots, n$, and $t = 0$ the termination state.

• There is control choice only at state 1. Otherwise, the spider simply moves towards the fly.

• Assume that the controls (in state 1) are $M =$ move, and $\bar{M} =$ don’t move.

• The transition probabilities from state 1 when using control $M$ are described in Figure 5.2.1.

\[ P_{11}(M) = 2p, \text{ described by the two possible situations:} \]

\[ \begin{array}{c}
\text{F} \\
\text{S}
\end{array} \quad \text{or} \quad \begin{array}{c}
\text{F} \\
\text{S}
\end{array} \]

\[ P_{10}(M) = 1 - 2p, \text{ when fly did not move} \]

Figure 5.2.1: Transition probabilities for control $M$ from state 1.

Other probabilities are:

\[ p_{12}(\bar{M}) = p, \quad p_{11}(\bar{M}) = 1 - 2p, \quad p_{10}(\bar{M}) = p, \]

and for $i \geq 2$,

\[ p_{ii} = p, \quad p_{i(i-1)} = 1 - 2p, \quad p_{i(i-2)} = p. \]

All other transition probabilities are zero.
Bellman’s equation:

\[ J^*(i) = 1 + pJ^*(i) + (1 - 2p)J^*(i - 1) + pJ^*(i - 2), \quad i \geq 2. \]

\[ J^*(1) = 1 + \min \left\{ \frac{2pJ^*(1)}{M}, \frac{pJ^*(2) + (1 - 2p)J^*(1)}{M} \right\}, \]

with \( J^*(0) = 0 \).

In order to solve the Bellman’s equation, we proceed as follows: First, note that

\[ J^*(2) = 1 + pJ^*(2) + (1 - 2p)J^*(1). \]

Then, substitute \( J^*(2) \) in the equation for \( J^*(1) \), getting:

\[ J^*(1) = 1 + \min \left\{ \frac{2pJ^*(1)}{1 - p} + \frac{(1 - 2p)J^*(1)}{1 - p} \right\}. \]

Next, we work from here to find that when one unit away from the fly, it is optimal to use \( \bar{M} \) if and only if \( p \geq 1/3 \). Moreover, it can be verified that

\[ J^*(1) = \begin{cases} 1/(1 - 2p) & \text{if } p \leq 1/3, \\ 1/p & \text{if } p \geq 1/3. \end{cases} \]

Given \( J^*(1) \), we can compute \( J^*(2) \), and then \( J^*(i) \), for all \( i \geq 3 \).

5.2.1 Computational approaches

There are three main computational approaches used in practice for calculating the optimal cost function \( J^* \). From

**Value iteration**

The DP iteration

\[ J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)J_k(j) \right\}, \quad i = 1, \ldots, n, \]

is called value iteration.

From equation (5.2.9), we know that the error

\[ |J_{mK}(i) - J^*(i)| \leq Dp^K, \quad \text{for some constant } D. \]

The value iteration algorithm can sometimes be strengthened with the use of error bounds (i.e., they provide a useful guideline for stopping the value iteration algorithm while being assured that \( J_k \) approximates \( J^* \) with sufficient accuracy). In particular, it can be shown that for all \( k \) and \( j \), we have

\[ J_{k+1}(j) + (N^*(j) - 1)c_k \leq J^*(j) \leq J_{k+1}(j) + (N^k(j) - 1)c_k, \]

where
• $\mu^k$ is such that $\mu^k(i)$ attains the minimum in the $k$th iteration for all $i$,

• $N^*(j) =$ average number of stages to reach $t$ starting from $j$ and using some optimal stationary policy,

• $N^k(j) =$ average number of stages to reach $t$ starting from $j$ and using some stationary policy $\mu^k$,

• $c_k = \min_{i=1,\ldots,n} \{ J_{k+1}(i) - J_k(i) \}$,

• $\bar{c}_k = \max_{i=1,\ldots,n} \{ J_{k+1}(i) - J_k(i) \}$,

Unfortunately, the values $N^*(j)$ and $N^k(j)$ are easily computed or approximated only in some cases.

**Policy iteration**

• It generates a sequence $\mu^1, \mu^2, \ldots$ of stationary policies, starting with any stationary policy $\mu^0$.

• At a typical iteration, given a policy $\mu^k$, we perform two steps:

  (i) *Policy evaluation step:* Computes $J_{\mu^k}(i)$ as the solution of the linear system of equations

  $$J(i) = g(i, \mu^k(i)) + \sum_{j=1}^n p_{ij}(\mu^k(i))J(j), \quad i = 1, \ldots, n,$$

  in the unknowns $J(1), \ldots, J(n)$.

  (ii) *Policy improvement step:* Computes a new policy $\mu^{k+1}$ as

  $$\mu^{k+1}(i) = \arg \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^n p_{ij}(u)J_{\mu^k}(j) \right\}, \quad i = 1, \ldots, n.$$  

• The algorithm stops when $J_{\mu^k}(i) = J_{\mu^{k+1}}(i)$ for all $i$.

**Proposition 5.2.2** Under Assumption 5.2.1, the policy iteration algorithm for the stochastic shortest path problem generates an improving sequence of policies (i.e., $J_{\mu^{k+1}}(i) \leq J_{\mu^k}(i)$, $\forall i, k$) and terminates with an optimal policy.

**Proof:** For any $k$, consider the sequence generated by the recursion

$$J_{N+1}(i) = g(i, \mu^{k+1}(i)) + \sum_{j=1}^n p_{ij}(\mu^{k+1}(i))J_N(j), \quad i = 1, \ldots, n,$$

where $N = 0, 1, \ldots$, and the solution to equation (5.2.10):

$$J_0(i) = J_{\mu^k}(i), \quad i = 1, \ldots, n.$$
From equation (5.2.10), we have

\[ J_0(i) = g(i, \mu_k(i)) + \sum_{j=1}^{n} p_{ij}(\mu_k(i))J_0(j) \]

\[ \geq g(i, \mu_k+1(i)) + \sum_{j=1}^{n} p_{ij}(\mu_k+1(i))J_0(j) \quad \text{(from (5.2.11))} \]

\[ = J_1(i), \quad \forall i \quad \text{(from iteration (5.2.12))} \]

By using the above inequality we obtain

\[ J_1(i) = g(i, \mu_k+1(i)) + \sum_{j=1}^{n} p_{ij}(\mu_k+1(i))J_0(j) \]

\[ \geq g(i, \mu_k+1(i)) + \sum_{j=1}^{n} p_{ij}(\mu_k+1(i))J_1(j) \quad \text{(because } J_0(i) \geq J_1(i)) \]

\[ = J_2(i), \quad \forall i \quad \text{(from iteration (5.2.12)).} \tag{5.2.13} \]

Continuing similarly we get

\[ J_0(i) \geq J_1(i) \geq \cdots \geq J_N(i) \geq J_{N+1}(i) \geq \cdots, \quad i = 1, \ldots, n. \tag{5.2.14} \]

Since by Proposition 5.2.1(c), \( J_N(i) \to J_{\mu_k+1}(i) \), we obtain

\[ J_0(i) \geq J_{\mu_k+1}(i) \quad \Rightarrow \quad J_{\mu_k}(i) \geq J_{\mu_k+1}(i), \quad i = 1, \ldots, n, \quad k = 0, 1, \ldots \tag{5.2.15} \]

Thus, the sequence of generated policies is improving, and since the number of stationary policies is finite, we must after a finite number of iterations—say, \( k + 1 \)—obtain \( J_{\mu_k}(i) = J_{\mu_k+1}(i) \), for all \( i \).

Then, we will have equality holding throughout equation (5.2.15), which in particular means from (5.2.12),

\[ J_0(i) = J_{\mu_k}(i) = J_1(i) = g(i, \mu_k+1(i)) + \sum_{j=1}^{n} p_{ij}(\mu_k+1(i))J_{\mu_k}(j), \]

and in particular,

\[ J_{\mu_k}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)J_{\mu_k}(j) \right\}, \quad i = 1, \ldots, n. \]

Thus, the costs \( J_{\mu_k}(1), \ldots, J_{\mu_k}(n) \) solve Bellman’s equation and by Proposition 5.2.1(b), it follows that \( J_{\mu_k}(i) = J^*(i) \), and that \( \mu_k(i) \) is optimal.

**Linear programming**

**Claim:** \( J^* \) is the “largest” \( J \) that satisfies the constraints

\[ J(i) \leq g(i, u) + \sum_{j=1}^{n} p_{ij}(u)J(j), \tag{5.2.16} \]

for all \( i = 1, \ldots, n \), and \( u \in U(i) \).
CHAPTER 5. INFINITE HORIZON PROBLEMS

PROOF: Assume that \( J_0(i) \leq J_1(i) \), where \( J_1(i) \) is generated through value iteration; i.e.,

\[
J_0(i) \leq \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)J_0(j) \right\}, \quad i = 1, \ldots, n.
\]

Then because of the stationarity of the problem and the monotonicity property of DP, we will have \( J_k(i) \leq J_{k+1}(i) \), for all \( k \) and \( i \). From Proposition 5.2.1(a), the value iteration sequence converges to \( J^*(i) \), so that \( J_0(i) \leq J^*(i) \), for all \( i \).

Hence, \( J^* = (J^*(1), \ldots, J^*(n)) \) is the solution of the linear program

\[
\max \sum_{i=1}^{n} J(i),
\]

subject to the constraint (5.2.16).

Figure 5.2.2 illustrates a linear program associated with a two-state stochastic shortest path problem. The decision variables in this case are \( J(1) \) and \( J(2) \).

\[
\begin{align*}
J(1) &= g(1, u^1) + p_{11}(u^1)J(1) + p_{12}(u^1)J(2) \\
J(2) &= g(1, u^2) + p_{11}(u^2)J(1) + p_{12}(u^2)J(2) \\
J(1) &= g(2, u^1) + p_{21}(u^1)J(1) + p_{22}(u^1)J(2) \\
J(2) &= g(2, u^2) + p_{21}(u^2)J(1) + p_{22}(u^2)J(2)
\end{align*}
\]

Figure 5.2.2: Illustration of the LP solution method for infinite horizon DP.

Drawback: For large \( n \), the dimension of this program is very large. Furthermore, the number of constraints is equal to the number of state-control pairs.

5.3 Discounted problems

- Go back to the total cost problem, but now assume a discount factor \( \alpha < 1 \) (i.e., future costs matter less than current cost).

- Can be converted to a stochastic shortest path (SSP) problem, for which the analysis of the preceding section holds.

- The transformation mechanism relies on adjusting the probabilities using the discount factor \( \alpha \). The instantaneous costs \( g(i, u) \) are preserved. Figure 5.3.1 illustrates this transformation.
• Justification: Take a policy \( \mu \), and apply it over both formulations. Note that:

– Given that the terminal state has not been reached in SSP, the state evolution in the two problems is governed by the same transition probabilities.

– The expected cost of the \( k \)th stage of the associated SSP is \( g(x_k, \mu_k(x_k)) \), multiplied by the probability that state \( t \) has not been reached, which is \( \alpha^k \). This is also the expected cost of the \( k \)th stage of the discounted problem.

– Note that value iteration produces identical iterates for the two problems:

\[
J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J_k(j) \right\}, \quad i = 1, \ldots, n.
\]

Corresponding SSP:

\[
J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J_k(j) \right\}, \quad i = 1, \ldots, n.
\]

Figure 5.3.1: Illustration of the transformation from \( \alpha \)-discounted to stochastic shortest path.

• The results of SPP, summarized in Proposition 5.2.1, extend to this case. In particular:

(i) Value iteration converges to \( J^* \) for all initial \( J_0 \):

\[
J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J_k(j) \right\}, \quad i = 1, \ldots, n.
\]

(ii) \( J^* \) is the unique solution of Bellman’s equation:

\[
J^*(i) = \min_{u \in U(i)} \left\{ g(i, u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J^*(j) \right\}, \quad i = 1, \ldots, n.
\]

(iii) Policy iteration converges finitely to an optimal.

(iv) Linear programming also works.

For completeness, we compile these results in the following proposition.

**Proposition 5.3.1** The following hold for the discounted problem:
(a) Given any initial conditions $J_0(1), \ldots, J_0(n)$, the value iteration algorithm

$$J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \alpha \sum_{j=1}^{n} p_{ij}(u)J_k(j) \right\}, \quad \forall i,$$

converges to the optimal cost $J^*(i)$.

(b) The optimal costs $J^*(1), \ldots, J^*(n)$ satisfy Bellman’s equation,

$$J^*(i) = \min_{u \in U(i)} \left\{ g(i, u) + \alpha \sum_{j=1}^{n} p_{ij}(u)J^*(j) \right\}, \quad i = 1, \ldots, n,$$

and in fact they are the unique solution of this equation.

(c) For any stationary policy $\mu$, the costs $J_\mu(1), \ldots, J_\mu(n)$ are the unique solution of the equation

$$J_\mu(i) = g(i, \mu(i)) + \alpha \sum_{j=1}^{n} p_{ij}(\mu(i))J_\mu(j), \quad i = 1, \ldots, n.$$

Furthermore, given any initial conditions $J_0(1), \ldots, J_0(n)$, the sequence $J_k(i)$ generated by the DP iteration

$$J_{k+1}(i) = g(i, \mu(i)) + \alpha \sum_{j=1}^{n} p_{ij}(\mu(i))J_k(j), \quad i = 1, \ldots, n,$$

converges to the cost $J_\mu(i)$ for each $i$.

(d) A stationary policy $\mu$ is optimal if and only if for every state $i$, $\mu(i)$ attains the minimum in Bellman’s equation of part (b).

(e) The policy iteration algorithm given by

$$\mu^{k+1}(i) = \arg \min_{u \in U(i)} \left\{ g(i, u) + \alpha \sum_{j=1}^{n} p_{ij}(u)J_\mu^k(j) \right\}, \quad i = 1, \ldots, n,$$

generates an improving sequence of policies and terminates with an optimal policy.

As in the case of stochastic shortest path problems (see equation (5.2.9)), we can show that

- $|J_k(i) - J^*(i)| \leq D\alpha^k$, for some constant $D$.
- The error bounds become

$$J_{k+1}(j) + \frac{\alpha}{1 - \alpha} \xi_k \leq J^*(j) \leq J^{\mu_k}(j) \leq J_{k+1}(j) + \frac{\alpha}{1 - \alpha} \bar{\xi}_k,$$

where $\mu^k(j)$ attains the minimum in the $k$th value iteration (5.3.1) for all $i$, and

$$\xi_k = \min_{i=1, \ldots, n} \left| J_{k+1}(i) - J_k(i) \right|, \quad \text{and} \quad \bar{\xi}_k = \max_{i=1, \ldots, n} \left| J_{k+1}(i) - J_k(i) \right|.$$

Example 5.3.1 (Asset selling problem)
• Assume system evolves according to $x_{k+1} = w_k$.

• If the offer $x_k$ of period $k$ is accepted, it is invested at an interest rate $r$.

• By depreciating the sale amount to period 0 dollars, we view $(1 + r)^{-k}x_k$ as the reward for selling the asset in period $k$ at a price $x_k$, where $r > 0$ is the interest rate.

  Idea: We discount the reward by the interest we did not make for the first $k$ periods.

• The discount factor is therefore: $\alpha = 1/(1 + r)$.

• $J^*$ is the unique solution of Bellman’s equation

$$J^*(x) = \max \left\{ x, \frac{E[J^*(w)]}{1 + r} \right\}.$$ 

• An optimal policy is to sell of and only if the current offer $x_k$ is greater than or equal to $\bar{\alpha}$, where

$$\bar{\alpha} = \frac{E[J^*(w)]}{1 + r}.$$ 

\[\square\]

Example 5.3.2 (Manufacturer’s production plan)

• A manufacturer at each time period receives an order for her product with probability $p$ and receives no order with probability $1 - p$.

• At any period she has a choice of processing all unfilled orders in a batch, or process no order at all.

• The cost per unfilled order at each time period is $c > 0$, and the setup cost to process the unfilled orders is $K > 0$. The manufacturer wants to find a processing policy that minimizes the total expected cost, assuming the discount factor is $\alpha < 1$ and the maximum number of orders that can remain unfilled is $n$. When the maximum $n$ of unfilled orders is reached, the orders must necessarily be processed.

• Define the state as the number of unfilled orders at the beginning of each period. The Bellman’s equation for this problem is

$$J^*(i) = \min \left\{ K + \alpha(1 - p)J^*(0) + \alpha pJ^*(1), \quad ci + \alpha(1 - p)J^*(i) + \alpha pJ^*(i + 1) \right\}$$

for the states $i = 0, 1, \ldots, n - 1$, and takes the form

$$J^*(n) = K + \alpha(1 - p)J^*(0) + \alpha pJ^*(1)$$

for state $n$.

• Consider the value iteration method applied over this problem. We prove now by using the (finite horizon) DP algorithm that the $k$-stage optimal cost functions $J_k(i)$ are monotonically nondecreasing in $i$ for all $k$, and therefore argue that the optimal infinite horizon cost function $J^*(i)$ is also monotonically nondecreasing in $i$ since

$$J^*(i) = \lim_{k \to \infty} J_k(i).$$
Given that \( J^*(i) \) is monotonically nondecreasing in \( i \), we have that if processing a batch of \( m \) orders is optimal, that is,

\[
K + \alpha(1 - p)J^*(0) + \alpha p J^*(1) \leq cm + \alpha(1 - p)J^*(m) + \alpha p J^*(m + 1),
\]

then processing a batch of \( m + 1 \) orders is also optimal. Therefore, a threshold policy (i.e., a policy that processes the orders if their number exceeds some threshold integer \( m^* \)) is optimal.

**Claim:** The \( k \)-stage optimal cost functions \( J_k(i) \) are monotonically nondecreasing in \( i \) for all \( k \).

**Proof:** We proceed by induction. Start from \( J_0(i) = 0 \), for all \( i \), and suppose that \( J_k(i + 1) \geq J_k(i) \) for all \( i \). We will see that \( J_{k+1}(i + 1) \geq J_{k+1}(i) \) for all \( i \). Consider first the case \( i + 1 < n \). Then, by induction hypothesis, we have

\[
c(i + 1) + \alpha(1 - p)J_k(i + 1) + \alpha p J_k(i + 2) \geq ci + \alpha(1 - p)J_k(i) + \alpha p J_k(i + 1). \tag{5.3.2}
\]

Define for any scalar \( \gamma \),

\[
F_k(\gamma) = \min\{K + \alpha(1 - p)J_k(0) + \alpha p J_k(1), \gamma\}.
\]

Since \( F_k(\gamma) \) is monotonically increasing in \( \gamma \), we have from equation (5.3.2),

\[
J_{k+1}(i + 1) = F_k(c(i + 1) + \alpha(1 - p)J_k(i + 1) + \alpha p J_k(i + 2)) \geq F_k(ci + \alpha(1 - p)J_k(i) + \alpha p J_k(i + 1)) = J_{k+1}(i).
\]

Finally, consider the case \( i + 1 = n \). Then, we have

\[
J_{k+1}(n) = K + \alpha(1 - p)J_k(0) + \alpha p J_k(1) \geq F_k(ci + \alpha(1 - p)J_k(i) + \alpha p J_k(i + 1)) = J_{k+1}(n - 1).
\]

The induction is complete.

### 5.4 Average cost-per-stage problems

#### 5.4.1 General setting

- Stationary system with finite number of states and controls
- Minimize over admissible policies \( \pi = \{\mu_0, \mu_1, \ldots\} \),
  
  \[
  J_\pi(i) = \lim_{N \to \infty} \frac{1}{N} E \left[ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k)) \mid x_0 = i \right]
  \]
  - Assume \( 0 \leq g(x_k, \mu_k(x_k)) < \infty \).
• Fact: For most problems of interest, the average cost per stage of a policy and the optimal average cost per stage are independent of the initial state.

Intuition: Costs incurred in the early stages do not matter in the long run. More formally, suppose that all state communicate under a given stationary policy \( \mu \). Let

\[
K_{ij}(\mu) = \text{first passage time from } i \text{ to } j \text{ under } \mu,
\]

i.e., \( K_{ij}(\mu) \) is the first index \( k \) such that \( x_k = j \) starting from \( x_0 = i \). Then,

\[
J_\mu(i) = \lim_{N \to \infty} \frac{1}{N} E \left[ \sum_{k=0}^{K_{ij}(\mu)-1} g(x_k, \mu_k(x_k)) | x_0 = i \right] + \lim_{N \to \infty} \frac{1}{N} E \left[ \sum_{k=K_{ij}(\mu)}^{N-1} g(x_k, \mu_k(x_k)) | x_0 = i \right] = 0.
\]

Therefore, \( J_\mu(i) = J_\mu(j) \), for all \( i, j \) with \( E[K_{ij}(\mu)] < \infty \) (or equivalently, with \( P(K_{ij}(\mu) = \infty) = 0 \).

• Because communication issues are so important, the methodology relies heavily on Markov chain theory.

5.4.2 Associated stochastic shortest path (SSP) problem

Assumption 5.4.1 State \( n \) is such that for some integer \( m > 0 \), and for all initial states and all policies, \( n \) is visited with positive probability at least once within the first \( m \) stages.

In other words, state \( n \) is recurrent in the Markov chain corresponding to each stationary policy.

Consider a sequence of generated states, and divide it into cycles that go through \( n \), as shown in Figure 5.4.1.

Figure 5.4.1: Each cycle can be viewed as a state trajectory of a corresponding SSP problem with termination state being \( n \).

The SSP is obtained via the transformation described in Figure 5.4.2.

Let the cost at \( i \) of the SSP be \( g(i, u) - \lambda^* \). We will show that

\[
\text{Average cost problem} \equiv \text{Min cost cycle problem} \equiv \text{SSP problem}
\]

\(^1\)We are assuming that there is a single recurrent class. Recall that a state is recurrent if the probability of reentering it is one. Positive recurrent means that the expected time of returning to it is finite. Also, recall that in a finite state Markov chain, all recurrent states are positive recurrent.
5.4.3 Heuristic argument

- Under all stationary policies in the original average cost problem, there will be an infinite number of cycles marked by successive visits to state $n$ ⇒ We want to find a stationary policy $\mu$ that minimizes the average cycle stage cost.

- Consider a minimum cycle cost problem: Find a stationary policy $\mu$ that minimizes:

$$\text{Expected cost per transition within a cycle} = \frac{\mathbb{E}[\text{cost from } n \text{ up to the first return to } n]}{\mathbb{E}[\text{time from } n \text{ up to the first return to } n]} = \frac{C_{nn}(\mu)}{N_{nn}(\mu)}.$$ 

- Intuitively, the optimal average cost $\lambda^*$ should be equal to optimal average cycle cost, i.e.,

$$\lambda^* = \frac{C_{nn}(\mu^*)}{N_{nn}(\mu^*)}; \quad \text{or equivalently} \quad C_{nn}(\mu^*) - N_{nn}(\mu^*)\lambda^* = 0.$$ 

So, for any stationary policy $\mu$,

$$\lambda^* \leq \frac{C_{nn}(\mu)}{N_{nn}(\mu)} \quad \text{or equivalently} \quad C_{nn}(\mu) - N_{nn}(\mu)\lambda^* \geq 0.$$ 

- Thus, to obtain an optimal policy $\mu$, we must solve

$$\min_{\mu} \{C_{nn}(\mu) - N_{nn}(\mu)\lambda^*\}$$

Note that $C_{nn}(\mu) - N_{nn}(\mu)\lambda^*$ is the expected cost of $\mu$ starting from $n$ in the associated SSP with stage cost $g(i, u) - \lambda^*$, justified by

$$\mathbb{E} \left[ \sum_{k=0}^{K_{nt}(\mu)-1} (g(x_k, \mu(x_k)) - \lambda^* | x_0 = n) \right] = C_{nn}(\mu) - N_{nn}(\mu) \frac{\lambda^*}{\mathbb{E}[K_{nt}(\mu)]}.$$
• Let $h^*(i)$ be the optimal cost of the SSP (i.e., of the path from $x_0 = i$ to $t$) when starting at states $i = 1, \ldots, n$. Then by Proposition 1(b) in $h^*(1), \ldots, h^*(n)$ solve uniquely the Bellman’s equation:

$$h^*(i) = \min_{u \in U(i)} \left\{ g(i, u) - \lambda^* + \sum_{j=1}^{n} p_{ij}(u) h^*(j) \right\} = \min_{u \in U(i)} \left\{ g(i, u) - \lambda^* + \sum_{j=1}^{n-1} p_{ij}(u) h^*(j) + p_{in}(u) h^*(n) \right\} \quad (5.4.1)$$

• If $\mu^*$ is a stationary policy that minimizes the cycle cost, then $\mu^*$ must satisfy

$$h^*(n) = C_{nn}(\mu^*) - N_{nn}(\mu^*) \lambda^* = 0$$

See Figure 5.4.3.

Figure 5.4.3: $h^*(n)$ in the SSP is the expected cost of the path from $n$ to $t$ (i.e., of the cycle from $n$ to $n$ in the original problem) based on the original $g(i, u)$, minus $N_{nn}(\mu^*) \lambda^*$.

• We can then rewrite (5.4.1) as

$$h^*(n) = 0$$

$$\lambda^* + h^*(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u) h^*(j) \right\}, \quad \forall i = 1, \ldots, n$$

From the results on SSP, we know that this equation has a unique solution (as long as we impose the constraint $h^*(n) = 0$). Moreover, minimization of the RHS should give an optimal stationary policy.

• Interpretation: $h^*(i)$ is a relative or differential cost:

$$h^*(i) = \min \left\{ \mathbb{E}[\text{cost to go from } i \text{ to } n \text{ for the first time}] - \mathbb{E}[\text{cost if the stage cost were constant at } \lambda^* \text{ instead of at } g(j, u), \forall j] \right\}$$

In words, $h^*(i)$ is a measure of how much away from the average cost we are when starting from node $i$. 

185
5.4.4 Bellman’s equation

The following proposition provides the main results regarding Bellman’s equation:

**Proposition 5.4.1** Under Assumption 1, the following hold for the average cost per stage problem:

(a) The optimal average cost $\lambda^*$ is the same for all initial states and together with some vector $h^* = \{h^*(1), \ldots, h^*(n)\}$ satisfies Bellman’s equation

$$
\lambda^* + h^*(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u) h^*(j) \right\}, \quad i = 1, \ldots, n. \quad (5.4.2)
$$

Furthermore, if $\mu(i)$ attains the minimum in the above equation for all $i$, the stationary policy $\mu$ is optimal. In addition, out of all vectors $h^*$ satisfying this equation, there is a unique vector for which $h^*(n) = 0$.

(b) If a scalar $\lambda$ and a vector $h = \{h(1), \ldots, h(n)\}$ satisfy Bellman’s equation, then $\lambda$ is the average optimal cost per stage for each initial state.

(c) Given a stationary policy $\mu$ with corresponding average cost per stage $\lambda_{\mu}$, there is a unique vector $h_{\mu} = \{h_{\mu}(1), \ldots, h_{\mu}(n)\}$ such that $h_{\mu}(n) = 0$ and

$$
\lambda_{\mu} + h_{\mu}(i) = g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i)) h_{\mu}(j), \quad i = 1, \ldots, n.
$$

**Proof:** We proceed item by item:

(a) Let $\tilde{\lambda} = \min_{\mu} \frac{C_{nn}(\mu) - N_{nn}(\mu)}{N_{nn}(\mu)}$. Then, for all $\mu$,

$$
C_{nn}(\mu) - N_{nn}(\mu) \tilde{\lambda} \geq 0,
$$

with

$$
\frac{C_{nn}(\mu^*) - N_{nn}(\mu^*) \tilde{\lambda}}{h^*(n)} \quad \text{in the associated SSP}
$$

Consider the associated SSP with stage cost: $g(i, u) - \tilde{\lambda}$. Then, by Proposition 1(b), and using the fact that $p_{in}(u) = 0$, the costs $h^*(1), \ldots, h^*(n)$ solve uniquely the corresponding Bellman’s equation:

$$
h^*(i) = \min_{u \in U(i)} \left\{ g(i, u) - \tilde{\lambda} + \sum_{j=1}^{n-1} p_{ij}(u) h^*(j) \right\}, \quad i = 1, \ldots, n. \quad (5.4.3)
$$

Thus, we can rewrite $(5.4.3)$ as

$$
\tilde{\lambda} + h^*(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u) h^*(j) \right\}, \quad i = 1, \ldots, n. \quad (5.4.4)
$$

We will show that this implies $\tilde{\lambda} = \lambda^*$. 

---

186
Let \( \pi = \{\mu_0, \mu_1, \ldots\} \) be any admissible policy, let \( N \) be a positive integer, and for all \( k = 0, \ldots, N - 1 \), define \( J_k(i) \) using the recursion:

\[
J_0(i) = h^*(i), \quad i = 1, \ldots, n,
\]

\[
J_{k+1}(i) = g(i, \mu_{N-k-1}(i)) + \sum_{j=1}^{n} p_{ij}(\mu_{N-k-1}(i))J_k(j), \quad i = 1, \ldots, n. \tag{5.4.5}
\]

In words, \( J_N(i) \) is the \( N \)-stage cost of \( \pi \) when starting state is \( i \) and the terminal cost is \( h^* \).

From (5.4.4), since \( \mu_{N-1}(\cdot) \) is just one admissible policy, we have

\[
\lambda + h^*(i) = \min_{\pi} \left\{ \sum_{j=1}^{n} p_{ij}(\mu_{N-1}(i))h^*(j) \mid \mu = (\mu_0, \mu_1, \ldots, \mu_n) \right\} = 0
\]

Thus, \( \lambda + J_0(i) \leq J_1(i), \quad i = 1, \ldots, n. \)

Then,

\[
J_2(i) = g(i, \mu_{N-2}(i)) + \sum_{j=1}^{n} p_{ij}(\mu_{N-2}(i))J_1(j) \geq \lambda + J_0(i)
\]

\[
\geq g(i, \mu_{N-2}(i)) + \lambda + \sum_{j=1}^{n} p_{ij}(\mu_{N-2}(i))J_0(j)
\]

\[
= \lambda + \lambda + h^*(i) \quad \text{(by equation (5.4.4))}
\]

\[
= 2\lambda + h^*(i), \quad i = 1, \ldots, n.
\]

By repeating this argument,

\[
k\lambda + h^*(i) \leq J_k(i), \quad k = 0, \ldots, N, \quad i = 1, \ldots, n.
\]

In particular, for \( k = N \),

\[
N\lambda + h^*(i) \leq J_N(i) \Rightarrow \lambda + \frac{h^*(i)}{N} \leq \frac{J_N(i)}{N}, \quad i = 1, \ldots, n. \tag{5.4.6}
\]

Equality holds in (5.4.6) if \( \mu_k(i) \) attains the minimum in (5.4.4) for all \( i, k \). Now,

\[
\lambda + \frac{h^*(i)}{N} \leq \frac{J_N(i)}{N}, \quad \text{as } N \to \infty
\]

where \( J_\pi(i) \) is the average cost per stage of \( \pi \), starting at \( i \). Then, we get

\[
\lambda \leq J_\pi(i), \quad i = 1, \ldots, n,
\]

for all admissible \( \pi \).

If \( \pi = \{\mu, \mu, \ldots\} \) where \( \mu(i) \) attains the minimum in (5.4.4) for all \( i, k \), we get

\[
\lambda = \min_{\pi} J_\pi(i) = \lambda^*, \quad i = 1, \ldots, n.
\]

Replacing \( \lambda \) by \( \lambda^* \) in equation (5.4.4), we obtain (5.4.2). Finally, “\( h^*(n) = 0 \)” jointly with (5.4.4) are equivalent to (5.4.3) for the associated SSP. But the solution to (5.4.3) is unique (due to Proposition 1(b)), so there must be a unique solution for the equations “\( h^*(n) = 0 \)” and (5.4.4).
(b) The proof follows from the proof of part (a), starting from equation (5.4.4).

(c) The proof follows from part (a), constraining the control set to \( \tilde{U}(i) = \{\mu(i)\} \).

Remarks:

- Proposition 5.4.1 can be shown under weaker conditions. In particular, it can be shown assuming that all stationary policies have a single recurrent class even if their corresponding recurrent classes do not have state \( n \) in common.

- It can also be shown assuming that for every pair of states \( i, j \), there is a stationary policy \( \mu \) under which there is a positive probability of reaching \( j \) starting from \( i \).

Example: A manufacturer, at each time:

1. May process all unfilled orders at cost \( K > 0 \), or process no order at all. The cost per unfilled order at each time is \( c > 0 \).

2. Receives an order w.p. \( p \), and no order w.p. \( 1 - p \).

- Maximum number of orders that can remain unfilled is \( n \). When there are \( n \) pending orders, he has to process.

- Objective: Find a processing policy that minimizes the total expected cost per stage.

- State: Number of unfilled orders. We set state 0 is the special state for the SSP formulation.

- Bellman’s equation: For states \( i = 0, 1, \ldots, n - 1 \),

\[
\lambda^* + h^*(i) = \min\{K + (1 - p)h^*(0) + ph^*(1), ci + (1 - p)h^*(i) + ph^*(i + 1)\}
\]

and for state \( n \),

\[
\lambda^* + h^*(n) = K + (1 - p)h^*(0) + ph^*(1).
\]

- Optimal policy: Process \( i \) unfilled orders if

\[
K + (1 - p)h^*(0) + ph^*(1) \leq ci + (1 - p)h^*(i) + ph^*(i + 1).
\]

- If we view \( h^*(i) \) as the differential cost associated with an optimal policy (or by interpreting \( h^*(i) \) as the optimal cost-to-go for the associated SSP), then \( h^*(i) \) should be monotonically nondecreasing with \( i \). This monotonicity implies that a threshold policy is optimal: “Process the orders if their number exceeds some threshold integer \( m \).”
5.4.5 Computational approaches

Value iteration

**Procedure:** Generate optimal $k$-stage costs by the DP algorithm starting from any $J_0$:

$$J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)J_k(j) \right\}, \quad \forall i.$$  \hfill (5.4.7)

**Claim:** $\lim_{k \to \infty} \frac{J_k(i)}{k} = \lambda^*, \quad \forall i.$

**Proof:** Let $h^*$ be a solution vector of Bellman’s equation:

$$\lambda^* + h^*(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)h^*(j) \right\}, \quad i = 1, \ldots, n.$$  \hfill (5.4.8)

From here, define the recursion

$$J_0^*(i) = h^*(i)$$

$$J_{k+1}^*(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)J_k^*(j) \right\}, \quad i = 1, \ldots, n.$$  

Like in the proof of Proposition 5.4.1(a), it can be shown that

$$J_k^*(i) = k\lambda^* + h^*(i), \quad i = 1, \ldots, n.$$  

On the other hand, it can be seen that

$$|J_k(i) - J_k^*(i)| \leq \max_{j=1, \ldots, n} |J_0(j) - h^*(j)|, \quad i = 1, \ldots, n,$$

because $J_k(i)$ and $J_k^*(i)$ are optimal costs for two $k$-stage problems that differ only in the corresponding terminal cost functions which are $J_0$ and $h^*$ respectively.

From the preceding two equations, we see that for all $k$,

$$|J_k(i) - (k\lambda^* - h^*(i))| \leq \max_{j=1, \ldots, n} |J_0(j) - h^*(j)|.$$  

Therefore,

$$-\max_{j=1, \ldots, n} |J_0(j) - h^*(j)| - h^*(i) \leq J_k(i) - k\lambda^* \leq \max_{j=1, \ldots, n} |J_0(j) - h^*(j)| - h^*(i),$$  

or equivalently,

$$|J_k(i) - k\lambda^*| \leq \max_{j=1, \ldots, n} |J_0(j) - h^*(j)| + \max_{j=1, \ldots, n} |h^*(j)|,$$

which implies

$$\left| \frac{J_k(i)}{k} - \lambda^* \right| \leq \frac{\text{constant}}{k}.$$  

Taking limit as $k \to \infty$ in both sides above gives

$$\lim_{k \to \infty} \frac{J_k(i)}{k} = \lambda^*.$$  

The only condition required is that Bellman’s equation (5.4.8) holds for some vector $h^*$.

**Remarks:**
Fixing the difficulties:

- Subtract the same constant from all components of the vector $J_k$:
  \[ J_k(i) := J_k(i) - C; \quad i = 1, \ldots, n. \]

- Consider the algorithm:
  \[
  h_k(i) = J_k(i) - J_k(s); \\
  
  h_{k+1}(i) = J_{k+1}(i) - J_{k+1}(s)
  \]
  for some fixed state $s$, and for all $i = 1, \ldots, n$. By using equation (5.4.7) for $i = 1, \ldots, n$,

  \[
  h_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)J_k(j) \right\} - \min_{u \in U(s)} \left\{ g(s, u) + \sum_{j=1}^{n} p_{sj}(u)J_k(j) \right\}
  \]

  \[
  = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)h_k(j) - J_k(s) \right\} - \min_{u \in U(s)} \left\{ g(s, u) + \sum_{j=1}^{n} p_{sj}(u)h_k(j) - J_k(s) \right\}
  \]

  \[
  = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)h_k(j) - J_k(s) \right\} - J_k(s) - \min_{u \in U(s)} \left\{ g(s, u) + \sum_{j=1}^{n} p_{sj}(u)h_k(j) \right\} + J_k(s)
  \]

- The above algorithm is called relative value iteration.
  
  - Mathematically equivalent to the value iteration method (5.4.7) that generates $J_k(i)$.
  - Iterates generated by the two methods differ by a constant (i.e., $J_k(s)$, since $J_k(i) = h_k(i) + J_k(s)$, $\forall i$).
  
- Big advantage of new method: Under Assumption 5.4.1 it can be shown that the iterates $h_k(i)$ are bounded, while this is typically not true for the “plain vanilla” method.

- It can be seen that if the relative value iteration converges to some vector $h$, then we have: $h(s) = 0$, and

  \[
  \lambda + h(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)h(j) \right\}
  \]

  By Proposition 5.4.1(b), this implies that $\lambda$ is indeed the optimal average cost per stage, and $h$ is the associated differential cost vector.

- Disadvantage: Under Assumption 5.4.1, convergence is not guaranteed. However, convergence can be guaranteed for a simple variant:

  \[
  h_{k+1}(i) = (1-\tau)h_k(i) + \min_{u \in U(i)} \left\{ g(i, u) + \tau \sum_{j=1}^{n} p_{ij}(u)h_k(j) \right\} - \min_{u \in U(s)} \left\{ g(s, u) + \tau \sum_{j=1}^{n} p_{sj}(u)h_k(j) \right\},
  \]

  for $i = 1, \ldots, n$, and $\tau$ a constant satisfying $0 < \tau < 1$.  

- Pros: Very simple to implement

- Cons:
  
  - Since typically some of the components of $J_k$ diverge to $\infty$ or $-\infty$, direct calculation of $\lim_{k \to \infty} \frac{J_k(i)}{k}$ is numerically cumbersome.
  - Method does not provide a corresponding differential cost vector $h^*$. 

Policy iteration

- Start from an arbitrary stationary policy \( \mu^0 \).
- At a typical iteration, we have a stationary policy \( \mu^k \). We perform two steps per iteration:
  - **Policy evaluation:** Compute \( \lambda^k \) and \( h^k(i) \) of \( \mu^k \), using the \( n + 1 \) equations \( h^k(n) = 0 \), and for \( i = 1, \ldots, n \),
    \[
    \lambda^k + h^k(i) = g(i, \mu^k(i)) + \sum_{j=1}^{n} p_{ij}(\mu^k(i))h^k(j).
    \]
    If \( \lambda^{k+1} = \lambda^k \) and \( h^{k+1}(i) = h^k(i), \forall i \), stop. Otherwise, continue with the next step.
  - **Policy improvement:** Find a stationary policy \( \mu^{k+1} \) where for all \( i \), \( \mu^{k+1}(i) \) is such that
    \[
    \mu^{k+1}(i) = \arg \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)h^k(j) \right\},
    \]
    and repeat.

- The next proposition shows that each iteration of the algorithm makes some irreversible progress towards optimality.

**Proposition 5.4.2** Under Assumption 5.4.1, in the policy iteration algorithm, for each \( k \) we either have \( \lambda^{k+1} < \lambda^k \); or else we have
  \[
  \lambda^{k+1} = \lambda^k, \quad \text{and} \quad h^{k+1}(i) = h^k(i), \forall i.
  \]
  Furthermore, the algorithm terminates and the policies \( \mu^k \) and \( \mu^{k+1} \) obtained upon termination are optimal.

**Proof:** Denote \( \mu^k := \mu, \mu^{k+1} := \bar{\mu}, \lambda^k := \lambda, \lambda^{k+1} := \bar{\lambda}, h^k(i) := h(i), h^{k+1} := \bar{h}(i) \). Define for \( N = 1, 2, \ldots, \)
  \[
  h_N(i) = g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i))h_{N-1}(j), \quad i = 1, \ldots, n,
  \]
  where \( h_0(i) = h(i) \).
  Thus, we have
  \[
  \bar{\lambda} = J_{\bar{\mu}}(i) = \lim_{N \to \infty} \frac{1}{N} h_N(i), \quad i = 1, 2, \ldots, n. \quad (5.4.9)
  \]
  By definition of \( \bar{\mu} \) we have for all \( i = 1, \ldots, n \):
  \[
  h_1(i) = g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i))h_0(j) \quad \text{(from the iteration above)}
  \leq g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i))h_0(j) \quad \text{(because \( \bar{\mu} \) was the min of this RHS)}
  = \lambda + h_0(i) \quad \text{(because of Proposition 5.4.1)}.
  \]
From the equation above, we also obtain
\[
  h_2(i) = g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i))h_1(j)
\]
\[
  \leq g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i))(\lambda + h_0(j))
\]
\[
  = \lambda + g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i))h_0(j)
\]
\[
  \leq \lambda + g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i))h_0(j)
\]
\[
  \leq \lambda + \left( g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i))h_0(j) \right)
\]
\[
  = 2\lambda + h_0(i),
\]
and by proceeding similarly, we see that for all \( i, N \),
\[
  h_N(i) \leq N\lambda + h_0(i).
\]

Thus,
\[
  \frac{h_N(i)}{N} \leq \lambda + \frac{h_0(i)}{N}
\]
Taking limit as \( N \to \infty \), the LHS converges to \( \bar{\lambda} \) (from equation (5.4.9)), and the 2nd term in the RHS goes to zero, implying that \( \lambda \leq \lambda \).

- If \( \bar{\lambda} = \lambda \), the iteration that produces \( \mu^{k+1} \) is a policy improvement step for the associated SSP with cost per stage \( g(i, \mu^{k}) - \lambda \). Moreover, \( h(i) \) and \( \bar{h}(i) \) are the optimal costs starting from \( i \) and corresponding to \( \mu \) and \( \bar{\mu} \) respectively, in this associated SSP. Thus, \( \bar{h}(i) \leq h(i), \forall i \).

- Since there are only a finite number of stationary policies, there are also a finite number of \( \lambda \) (each one being the average cost per stage of each of the stationary policies). For each \( \lambda \) there is only a finite number of possible vectors \( h \) (see Proposition 5.4.1(c), where we can vary the reference \( h_{\mu}(n) = 0 \)).

- In view of the improvement properties already shown, no pair \( (\lambda, h) \) can be repeated without termination of the algorithm, implying that the algorithm must terminate with \( \bar{\lambda} = \lambda \) and \( \bar{h}(i) = h(i), \forall i \).

**Claim:** When the algorithm terminates, the policies \( \bar{\mu} \) and \( \mu \) are optimal.

**Proof:** Upon termination, we have for all \( i \),
\[
  \lambda + h(i) = \bar{\lambda} + \bar{h}(i)
\]
\[
  = g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i))\bar{h}(j) \quad \text{(by policy evaluation step)}
\]
\[
  = g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i))h(j) \quad \text{(because \( \bar{h}(j) = h(j), \forall j \))}
\]
\[
  = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)h(j) \right\} \quad \text{(by policy improvement step)}
\]
Therefore, \((\lambda, h)\) satisfy Bellman’s equation, and by Proposition 5.4.1(b), \(\lambda\) must be equal to the optimal average cost per stage. Furthermore, \(\bar{\mu}(i)\) attains the minimum in the RHS of Bellman’s equation (see the last two equalities above), and hence by Proposition 5.4.1(a), \(\bar{\mu}\) is optimal. Since we also have for all \(i\) (due to the self-consistency of the policy evaluation step),

\[
\lambda + h(i) = g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i))h(j),
\]

the same is true for \(\mu\).

5.5 Semi-Markov Decision Problems

5.5.1 General setting

- Stationary system with finite number of states and controls.
- State transitions occur at discrete times.
- Control applied at these discrete times and stays constant between transitions.
- Time between transitions is random, or may depend on the current state and the choice of control.
- Cost accumulates in continuous time, or maybe incurred at the time of transition.
- Example: Admission control in a system with restricted capacity (e.g., a communication link)
  - Customer arrivals: Poisson process.
  - Customers entering the system, depart after an exponentially distributed time.
  - Upon arrival we must decide whether to admit or block a customer.
  - There is a cost for blocking a customer.
  - For each customer that is in the system, there is a customer-dependent reward per unit of time.
  - Objective: Minimize time-discounted or average cost.
- Note that at transition times \(t_k\), the future of the system statistically depends only on the current state. This is guaranteed by not allowing the control to change in between transitions. Otherwise, we should include the time elapsed from the last transition as part of the system state.

5.5.2 Problem formulation

- \(x(t)\) and \(u(t)\): State and control at time \(t\). Stay constant between transitions.
- \(t_k\): Time of the \(k\)th transition \((t_0 = 0)\).
- \(x_k = x(t_k)\): We have \(x(t) = x_k\) for \(t_k \leq t \leq t_{k+1}\).
• \( u_k = u(t_k) \): We have \( u(t) = u_k \) for \( t_k \leq t \leq t_{k+1} \).

• In place of transition probabilities, we have transition distributions. For any pair (state \( i \), control \( u \)), specify the joint distribution of the transition interval and the next state:

\[
Q_{ij}(\tau, u) = \mathbb{P}\{t_{k+1} - t_k \leq \tau, x_{k+1} = j | x_k = i, u_k = u\}.
\]

• Two important observations:

1. Transition distributions specify the ordinary transition probabilities via

\[
p_{ij}(u) = \mathbb{P}\{x_{k+1} = j | x_k = i, u_k = u\} = \lim_{\tau \to \infty} Q_{ij}(\tau, u).
\]

We assume that for all states \( i \) and controls \( u \in U(i) \), the average transition time,

\[
\bar{\tau}_i(u) = \frac{1}{n} \sum_{j=1}^{n} \int_0^{\infty} \tau Q_{ij}(d\tau, u),
\]

is nonzero and finite, \( 0 < \bar{\tau}_i(u) < \infty \).

2. The conditional cumulative distribution function (c.d.f.) of \( \tau \) given \( i, j, \) and \( u \) is (assuming \( p_{ij}(u) > 0 \))

\[
\mathbb{P}\{x_{k+1} = j | x_k = i, u_k = u\} = \frac{Q_{ij}(u)}{p_{ij}(u)}. \tag{5.5.1}
\]

Thus, \( Q_{ij}(u) \) can be seen as a scaled c.d.f., i.e.,

\[
Q_{ij}(u) = \mathbb{P}\{x_{k+1} = j | x_k = i, u_k = u\} \times p_{ij}(u).
\]

**Important case: Exponential transition distributions**

• Important example of transition distributions:

\[
Q_{ij}(\tau, u) = p_{ij}(u)(1 - e^{-\nu_i(u)\tau}),
\]

where \( p_{ij}(u) \) are transition probabilities, and \( \nu_i(u) > 0 \) is called the transition rate at state \( i \).

• Interpretation: If the system is in state \( i \) and control \( u \) is applied,

- The next state will be \( j \) w.p. \( p_{ij}(u) \).
- The time between the transition to state \( i \) and the transition to the next state \( j \) is \( \text{Exp}(\nu_i(u)) \) (independently of \( j \));

\[
\mathbb{P}\{\text{transition time interval} > \tau | i, u\} = e^{-\nu_i(u)\tau}.
\]

• The exponential distribution is memoryless. This implies that for a given policy, the system is a continuous-time Markov chain (the future depends on the past through the present). Without the memoryless property, the Markov property holds only at the times of transition.
Cost structures

- There is a cost \( g(i, u) \) per unit time, i.e.,
  \[
g(i, u)dt = \text{cost incurred during small time period } dt
  \]

- There maybe an extra instantaneous cost \( \hat{g}(i, u) \) at the time of a transition (let’s ignore this for the moment).

- **Total discounted cost** of \( \pi = \{\mu_0, \mu_1, \ldots, \} \) starting from state \( i \) (with discount factor \( \beta > 0 \))
  \[
  \lim_{N \to \infty} E \left[ \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} e^{-\beta t} g(x_k, \mu_k(x_k))dt \mid x_0 = i \right]
  \]

- **Average cost per unit time** of \( \pi = \{\mu_0, \mu_1, \ldots, \} \) starting from state \( i \)
  \[
  \lim_{N \to \infty} \frac{1}{E[t_N \mid x_0 = i, \pi]} \left[ \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} g(x_k, \mu_k(x_k))dt \mid x_0 = i \right]
  \]

- We will see that both problems have equivalent discrete time versions.

A note on notation

- The scaled c.d.f. \( Q_{ij}(\tau, u) \) can be used to model discrete, continuous, and mixed distributions for the transition time \( \tau \).

- Generally, expected values of functions of \( \tau \) can be written as integrals involving \( dQ_{ij}(\tau, u) \).
  For example, from (5.5.1) (noting that there is no \( \tau \) in the denominator there), the conditional expected value of \( \tau \) given \( i, j, \) and \( u \) is written as
  \[
  E[\tau \mid i, j, u] = \int_0^\infty \tau \frac{dQ_{ij}(\tau, u)}{p_{ij}(u)}
  \]

- If \( Q_{ij}(\tau, u) \) is discontinuous and “staircase-like”, expected values can be written as summations.

5.5.3 Discounted cost problems

- For a policy \( \pi = \{\mu_0, \mu_1, \ldots, \} \), write
  \[
  J_\pi(i) = E[\text{cost of 1st transition}] + E[e^{-\beta \tau} J_{\pi_1}(j) \mid i, \mu_0(i)],
  \]  
  \[
  \text{(5.5.2)}
  \]

  where \( J_{\pi_1}(j) \) is the cost-to-go of the policy \( \pi_1 = \{\mu_1, \mu_2, \ldots, \} \).

- We calculate the two costs in the RHS. The expected cost of a single transition if \( u \) is applied at state \( i \) is
  \[
  G(i, u) = E_j[E_\tau[\text{transition cost} \mid j]]
  \]
  \[
  = \sum_{j=1}^n p_{ij}(u) \int_0^\infty \left( \int_0^\tau e^{-\beta t} g(i, u)dt \right) \frac{dQ_{ij}(\tau, u)}{p_{ij}(u)}
  \]
  \[
  = \sum_{j=1}^n \int_0^\infty \frac{1 - e^{-\beta \tau}}{\beta} g(i, u)dQ_{ij}(\tau, u),
  \]  
  \[
  \text{(5.5.3)}
  \]
where the 2nd equality follows from computing \( \mathbb{E}_\tau [\text{transition cost} | j] \) via integrating the tail of the nonnegative r.v. \( \tau \), and the 3rd one because \( \int_0^\infty e^{-\beta t} dt = (1 - e^{-\beta \tau})/\beta \).

Thus, \( \mathbb{E}[\text{cost of 1st transition}] \) is

\[
G(i, \mu_0(i)) = g(i, \mu_0(i)) \sum_{j=1}^n \int_0^\infty \frac{1 - e^{-\beta \tau}}{\beta} dQ_{ij}(\tau, \mu_0(i)).
\]

- Regarding the 2nd term in (5.5.2),

\[
\mathbb{E}[e^{-\beta \tau} J_{\pi_1}(j) | i, \mu_0(i)] = \mathbb{E}_j[\mathbb{E}[e^{-\beta \tau} | j, i, \mu_0(i)] J_{\pi_1}(j)] = \sum_{j=1}^n p_{ij}(\mu_0(i)) \left( \int_0^\infty e^{-\beta \tau} dQ_{ij}(\tau, \mu_0(i)) \right) J_{\pi_1}(j) = \sum_{j=1}^n m_{ij}(\mu_0(i)) J_{\pi_1}(j),
\]

where \( m_{ij}(u) \) is given by

\[
m_{ij}(u) = \int_0^\infty e^{-\beta \tau} dQ_{ij}(\tau, u).
\]

Note that \( m_{ij}(u) \) satisfies

\[
m_{ij}(u) < \int_0^\infty dQ_{ij}(\tau, u) = \lim_{\tau \to \infty} Q_{ij}(\tau, u) = p_{ij}(u).
\]

So, \( m_{ij}(u) \) can be viewed as the effective discount factor (the analog of \( \alpha p_{ij}(u) \) in the discrete-time case).

- So, going back to (5.5.2), \( J_{\pi}(i) \) can be written as

\[
J_{\pi}(i) = G(i, \mu_0(i)) + \sum_{j=1}^n m_{ij}(\mu_0(i)) J_{\pi_1}(j).
\]

**Equivalence to an SSP**

- Similar to the discrete-time case, introduce a stochastic shortest path problem with an artificial termination state \( t \).

- Under control \( u \), from state \( i \) the system moves to state \( j \) w.p. \( m_{ij}(u) \), and to the terminal state \( t \) w.p. \( 1 - \sum_{j=1}^n m_{ij}(u) \).

- Bellman’s equation: For \( i = 1, \ldots, n \),

\[
J^*(i) = \min_{u \in U(i)} \left\{ G(i, u) + \sum_{j=1}^n m_{ij}(u) J^*(j) \right\}
\]

- Analogs of value iteration, policy iteration, and linear programming.
If in addition to the cost per unit of time \( g \), there is an extra (instantaneous) one-stage cost \( \hat{g}(i,u) \), Bellman’s equation becomes

\[
J^*(i) = \min_{u \in U(i)} \left\{ \hat{g}(i,u) + G(i,u) + \sum_{j=1}^{n} m_{ij}(u) J^*(j) \right\}
\]

Example 5.5.1 (Manufacturer’s production plan)

- A manufacturer receives orders with interarrival times uniformly distributed in \([0, \tau_{\text{max}}]\).
- He may process all unfilled orders at cost \( K > 0 \), or process none. The cost per unit of time of an unfilled order is \( c \). Maximum number of unfilled orders is \( n \).
- Objective: Find a processing policy that minimizes the total expected cost, assuming the discount factor is \( \beta < 1 \).
- The nonzero transition distributions are

\[
Q_{i1}(\tau, \text{Fill}) = Q_{i,i+1}(\tau, \text{No Fill}) = \min \left\{ 1, \frac{\tau}{\tau_{\text{max}}} \right\}
\]

- The one-stage expected cost \( G \) (see equation (5.5.3)) is

\[
G(i, \text{Fill}) = 0, \quad G(i, \text{Not Fill}) = \gamma ci,
\]

where

\[
\gamma = \sum_{j=1}^{n} \int_{0}^{\infty} \frac{1 - e^{-\beta \tau}}{\beta} dQ_{ij}(\tau, u) = \int_{0}^{\tau_{\text{max}}} \frac{1 - e^{-\beta \tau}}{\beta \tau_{\text{max}}} d\tau.
\]

- There is an instantaneous cost

\[
\hat{g}(i, \text{Fill}) = K, \quad \hat{g}(i, \text{Not Fill}) = 0.
\]

- The effective discount factors \( m_{ij}(u) \) in Bellman’s equation are

\[
m_{i1}(\text{Fill}) = m_{i,i+1}(\text{Not Fill}) = \alpha,
\]

where

\[
\alpha = \int_{0}^{\infty} e^{-\beta \tau} dQ_{ij}(\tau, u) = \int_{0}^{\tau_{\text{max}}} \frac{e^{-\beta \tau}}{\tau_{\text{max}}} d\tau = \frac{1 - e^{-\beta \tau_{\text{max}}}}{\beta \tau_{\text{max}}}.
\]

- Bellman’s equation has the form

\[
J^*(i) = \min\{K + \alpha J^*(1), \gamma ci + \alpha J^*(i + 1)\}, \quad i = 1, 2, \ldots
\]

As in the discrete-time case, it can be proved that \( J^*(i) \) is monotonically decreasing in \( i \). Therefore, there must exist an optimal threshold \( i^* \) such that the manufacturer must fill the orders if and only if their number \( i \) exceeds \( i^* \). \( \square \)
5.5.4 Average cost problems

• Cost function for the continuous time average cost per unit time problem (assuming that there is a special state that is recurrent under all policies) would be

\[
\lim_{T \to \infty} \frac{1}{T} \mathbb{E}\left[\int_0^T g(x(t), u(t)) \, dt\right]
\]

However, we will use instead the cost function

\[
\lim_{N \to \infty} \frac{1}{\mathbb{E}[t_N]} \mathbb{E}\left[\int_0^{t_N} g(x(t), u(t)) \, dt\right],
\]

where \( t_N \) is the completion time of the \( N \)th transition. This cost function is equivalent to the previous one under the conditions of the subsequent analysis.

• We now apply the SSP argument used for the discrete-time case. Divide trajectory into cycles marked by successive visits to \( n \). The cost at \( (i, u) \) is \( G(i, u) - \lambda^* \bar{\tau}_i(u) \), where \( \lambda^* \) is the optimal expected cost per unit of time. Each cycle is viewed as a state trajectory of a corresponding SSP problem with the termination state being essentially \( n \).

• Bellman’s equation for the average cost problem is

\[
h^*(i) = \min_{u \in U(i)} \left\{ G(i, u) - \lambda^* \bar{\tau}_i(u) + \sum_{j=1}^n p_{ij}(u) h^*(j) \right\}.
\]

• The expected transition times are

\( \bar{\tau}_i(\text{Fill}) = \bar{\tau}_i(\text{Not Fill}) = \frac{\tau_{\text{max}}}{2} \).

• The expected transition cost is

\( G(i, \text{Fill}) = 0, \quad G(i, \text{Not Fill}) = \frac{c_i \tau_{\text{max}}}{2} \),

and the instantaneous cost is

\( \hat{g}(i, \text{Fill}) = K, \quad \hat{g}(i, \text{Not Fill}) = 0. \)

• Bellman’s equation is

\[
h^*(i) = \min \left\{ K - \lambda^* \frac{\tau_{\text{max}}}{2} + h^*(1), \quad c_i \frac{\tau_{\text{max}}}{2} - \lambda^* \frac{\tau_{\text{max}}}{2} + h^*(i + 1) \right\}.
\]

• Again, it can be shown that a threshold policy is optimal.

5.6 Application: Multi-Armed Bandits

5.7 Exercises

Exercise 5.7.1 A computer manufacturer can be in one of two states. In state 1 his product sells well, while in state 2 his product sells poorly. While in state 1 he can advertise his product
in which case the one-stage reward is 4 units, and the transition probabilities are \( p_{11} = 0.8 \) and \( p_{12} = 0.2 \). If in state 1, he does not advertise, the reward is 6 units and the transition probabilities are \( p_{11} = p_{12} = 0.5 \). While in state 2, he can do research to improve his product, in which case the one-stage reward is \(-5\) units, and the transition probabilities are \( p_{21} = 0.7 \) and \( p_{22} = 0.3 \). If in state 2 he does not do the research, the reward is \(-3\), and the transition probabilities are \( p_{21} = 0.4 \), and \( p_{22} = 0.6 \). Consider the infinite horizon, discounted version of this problem.

(a) Show that when the discount factor \( \alpha \) is sufficiently small, the computer manufacturer should follow the "shortsighted" policy of not advertising (not doing research) while in state 1 (state 2). By contrast, when \( \alpha \) is sufficiently close to 1, he should follow the "farsighted" policy of advertising (doing research) while in state 1 (state 2).

(b) For \( \alpha = 0.9 \), calculate the optimal policy using policy iteration.

(c) For \( \alpha = 0.99 \), use a computer to solve the problem by value iteration.

Exercise 5.7.2 An energetic salesman works every day of the week. He can work in only one of two towns A and B on each day. For each day he works in town A (or B) his expected reward is \( r_A \) (or \( r_B \), respectively). The cost of changing towns is \( c \). Assume that \( c > r_A > r_B \), and that there is a discount factor \( \alpha < 1 \).

(a) Show that for \( \alpha \) sufficiently small, the optimal policy is to stay in the town he starts in, and that for \( \alpha \) sufficiently close to 1, the optimal policy is to move to town A (if not starting there) and stay in A for all subsequent times.

(b) Solve the problem for \( c = 3, r_A = 2, r_B = 1 \) and \( \alpha = 0.9 \) using policy iteration.

(c) Use a computer to solve the problem of part (b) by value iteration.

Exercise 5.7.3 A person has an umbrella that she takes from home to office and vice versa. There is a probability \( p \) of rain at the same time she leaves home or office independently of earlier weather. If the umbrella is in the place where she is and it rains, she takes the umbrella to go to the other place (this involves no cost). If there is no umbrella and it rains, there is a cost \( W \) for getting wet. If the umbrella is in the place where she is but it does not rain, she may take the umbrella to go to the other place (this involves an inconvenience cost \( V \)) or she may leave the umbrella behind (this involves no cost). Costs are discounted at a factor \( \alpha < 1 \).

(a) Formulate this as an infinite horizon total cost discounted problem. Try to reduce the number of states of the model. Two or three states should be enough for this problem!

(b) Characterize the optimal policy as best as you can.

Exercise 5.7.4 An unemployed worker receives a job offer at each time period, which she may accept or reject. The offered salary takes one of \( n \) possible values \( w^1, \ldots, w^n \), with given probabilities, independently of preceding offers. If she accepts the offer, she must keep the job for the rest of her life at the same salary level. If she rejects the offer, she receives unemployment compensation \( c \) for the current period and is eligible to accept future offers. Assume that income is discounted by a factor \( \alpha < 1 \).
Hint: Define the states $s^i, i = 1, \ldots, n$, corresponding to the worker being unemployed and being offered a salary $w^i$, and $\bar{s}^i, i = 1, \ldots, n$, corresponding to the worker being employed at a salary level $w^i$.

(a) Show that there is a threshold $\bar{w}$ such that it is optimal to accept an offer if and only if its salary is larger than $\bar{w}$, and characterize $\bar{w}$.

(b) Consider the variant of the problem where there is a given probability $p_i$ that the worker will be fired from her job at any one period if her salary is $w^i$. Show that the result of part (a) holds in the case where $p_i$ is the same for all $i$. Argue what would happen in the case where $p_i$ depends on $i$.

**Exercise 5.7.5** An unemployed worker receives a job offer at each time period, which she may accept or reject. The offered salary takes one of $n$ possible values $w^1, \ldots, w^n$ with given probabilities, independently of preceding offers. If she accepts the offer, she must keep the job for the rest of her life at the same salary level. If she rejects the offer, she receives unemployment compensation $c$ for the current period and is eligible to accept future offers.

Suppose that there is a probability $p$ that the worker will be fired from her job at any one period, and further assume that $w^1 < w^2 < \cdots < w^n$.

Show that when the worker maximizes her average income per period, there is a threshold value $\bar{w}$ such that it is optimal to accept an offer if and only if her salary is larger than $\bar{w}$, and characterize $\bar{w}$.

**Hint:** Define the states $s^i, i = 1, \ldots, n$, corresponding to the worker being unemployed and being offered a salary $w^i$, and $\bar{s}^i, i = 1, \ldots, n$, corresponding to the worker being employed at a salary level $w^i$. 
Chapter 6

Point Process Control

The following chapter is based on Chapters I, II and VII in Brémaud’s book *Point Processes and Queues* (1981).

6.1 Basic Definitions

Consider some probability space \((\Omega, \mathcal{F}, \mathbb{P})\). A real-valued mapping \(X : \Omega \to \mathbb{R}\) is a random variable if for every \(C \in \mathcal{B}(\mathbb{R})\) the pre-image \(X^{-1}(C) \in \mathcal{F}\).

A filtration (or history) of a measurable space \((\Omega, \mathcal{F})\) is a collection \((\mathcal{F}_t\) of sub-\(\sigma\)-fields of \(\mathcal{F}\) such that for all \(0 \leq s \leq t\)

\[ \mathcal{F}_s \subseteq \mathcal{F}_t. \]

We denote by \(\mathcal{F}_\infty = \bigvee_{t \geq 0} \mathcal{F}_t := \sigma \left( \bigcup_{t \geq 0} \mathcal{F}_t \right)\).

A family \((X_t)_{t \geq 0}\) of real-valued random variables is called a stochastic process. The filtration generated by \(X_t\) is

\[ \mathcal{F}^X_t := \sigma \left( X_s : s \in [0, t] \right) . \]

For a fixed \(\omega \in \Omega\), the function \(X_t(\omega)\) is called a path of the stochastic process.

We say that the stochastic process is adapted to the filtration \(\mathcal{F}_t\) if \(\mathcal{F}^X_t \subseteq \mathcal{F}_t\) for all \(t \geq 0\). We say that \(X_t\) is \(\mathcal{F}_t\)-progressive if for all \(t \geq 0\) the mapping \((t, \omega) \to X_t(\omega)\) from \([0, t] \times \Omega \to \mathbb{R}\) is \(\mathcal{B}([0, t]) \otimes \mathcal{F}_t\)-measurable.

Let \(\mathcal{F}_t\) be a filtration. We define the \(\mathcal{F}_t\)-predictable \(\sigma\)-field \(\mathcal{P}(\mathcal{F}_t)\) as follows:

\[ \mathcal{P}(\mathcal{F}_t) := \sigma \left( (s, t] \times A : s \in [0, t] \text{ and } A \in \mathcal{F}_s \right) . \]

A stochastic process \(X_t\) is \(\mathcal{F}_t\)-predictable if \(X_t\) is \(\mathcal{P}(\mathcal{F}_t)\)-measurable.

**Proposition 6.1.1** A real-valued stochastic process \(X_t\) adapted to \(\mathcal{F}_t\) and left-continuous is \(\mathcal{F}_t\)-predictable

Given a filtration \(\mathcal{F}_t\), a process \(X_t\) is called a \(\mathcal{F}_t\)-martingale over \([0, c]\) if the following three conditions are satisfied
1. $X_t$ is adapted to $\mathcal{F}_t$.
2. $\mathbb{E}[|X_t|] < \infty$ for all $t \in [0, c]$.
3. $\mathbb{E}[X_t|\mathcal{F}_s] = X_s$ a.s., for all $0 \leq s \leq t \leq c$.

If the equality in (3) is replaced by $\geq$ ($\leq$) then $X_t$ is called a submartingale (supermartingale).

**Exercise 6.1.1** Let $X_t$ be a real-valued process with independent increment, that is, for all $0 \leq s \leq t$, $X_t - X_s$ is independent of $\mathcal{F}_s^X$. Suppose that $X_t$ is integrable and $\mathbb{E}[|X_t|] = 0$. Show that $X_t$ is a $\mathcal{F}_t$-martingale.

If in addition $X_t^2$ is integrable then $X_t^2$ is a $\mathcal{F}_t^X$-submartingale and $X_t^2 - \mathbb{E}[X_t^2]$ is a $\mathcal{F}_t$-martingale.

### 6.2 Counting Processes

**Definition 6.2.1** A sequence of random variables \( \{T_n : n \geq 0\} \) is called a point process if for each \( n \geq 0 \), \( T_n \in \mathcal{F} \) and
\[
\forall \omega \in \Omega \quad T_0(\omega) = 0 \quad \text{and} \quad T_n(\omega) < T_{n+1}(\omega) \quad \text{whenever} \quad T_n < \infty.
\]

We will only consider nonexplosive point process, that is, process for which \( \lim_{n \to \infty} T_n = \infty \) P-a.s.

Associated to a nonexplosive point process \( \{T_n\} \), we define the corresponding counting process \( \{N_t : t \geq 0\} \) as follows
\[
N_t = n \quad \text{if} \quad t \in [T_n, T_{n+1}).
\]

Thus, $N_t$ is a right-continuous step function starting at 0 (see figure). Since we are considering nonexplosive point processes, $N_t < \infty$ for all $t \geq 0$ P-a.s. In addition, if $\mathbb{E}[N_t]$ is finite for all $t$ then the point process is said to be integrable.

**Exercise 6.2.1 Simple Renewal Process:**

Consider a sequence \( \{X_n : n \geq 1\} \) of iid nonnegative random variables. We define the point process recursively as follows: $T_0 = 0$ and $T_n = T_{n-1} + X_n$, $n \geq 1$. A sufficient condition for the process $T_n$ to be nonexplosive is $\mathbb{E}[X] > 0$. 

202
One of the most famous renewal process is the Poisson process. In this case \( X \), the inter-arrival interval, has an exponential distribution with time homogenous rate \( \lambda \).

**Exercise 6.2.2 Queueing Process:**
A simple queueing process \( Q_t \) is a nonnegative integer-valued process \( o \) of the form
\[
Q_t = Q_0 + A_t - D_t,
\]
where \( A_t \) (arrival process) and \( D_t \) (departure process) are two nonexplosive point processes without common jumps. Note that by definition \( D_t \leq Q_0 + A_t \). □

**Exercise 6.2.3** Let \( N_t(1) \) and \( N_t(2) \) be two nonexplosive point processes without common jumps and let \( Q_0 \) be a nonnegative integer-valued random variable. Define \( X_t = Q_0 + N_t(1) - N_t(2) \) and \( m_t = \min_{0 \leq s \leq t} \{ X_s \} \). Show that \( Q_t = X_t - m_t \) is a simple queueing process with arrival process \( A_t = N_t(1) \), departure process \( D_t = \int_0^t 1( Q_s > 0 ) \, dN_s(2) \), and \( m_t = -\int_0^t 1( Q_s = 0 ) \, dN_s(2) \). □

Poisson processes are commonly used in practice to represent point process. One possible explanation for this popularity is its inherent mathematical tractability. However, a well-known result in renewal theory says that the sum of a large number of independent renewal processes converges –as the number of summands goes to infinity– to a Poisson process. So, the Poisson process can be in fact a good approximation to many applications in practice. To make the Poisson process even more realistic we would like to have more flexibility on the arrival rate \( \lambda \). We can achieve this generalization as follows.

**Definition 6.2.2** (Doubly Stochastic or Conditional Poisson Process) Let \( N_t \) be a point process adapted to a filtration \( \mathcal{F}_t \), and let \( \lambda_t \) be a nonnegative measurable process. Suppose that
\[
\lambda_t \text{ is } \mathcal{F}_0 - \text{measurable for all } t \geq 0,
\]
and that
\[
\int_0^t \lambda_s \, ds < \infty \quad P\text{-a.s., for all } t \geq 0.
\]
If for all \( s \in [0, t] \) the increment \( N_t - N_s \) is independent of \( \mathcal{F}_s \) given \( \mathcal{F}_0 \) and
\[
P[ N_t - N_s = k | \mathcal{F}_s ] = \frac{1}{k!} \exp \left( \int_s^t \lambda_u \, du \right) \left( \int_s^t \lambda_u \, du \right)^k
\]
then \( N_t \) is called a \((P - \mathcal{F}_t)\)-doubly stochastic Poisson process with stochastic intensity \( \lambda_t \).

The process \( \lambda_t \) is referred as the intensity of the process. A special case of a doubly stochastic Poisson process occurs when \( \lambda_t = \lambda \in \mathcal{F}_0 \). Another example is the case where \( \lambda_t = f(t, Y_t) \) for a measurable nonnegative function \( f \) and a process \( Y_t \) such that \( \mathcal{F}_\infty \subseteq \mathcal{F}_0 \).

**Exercise 6.2.4** Show that for a doubly stochastic Poisson process \( N_t \) is such that \( \mathbb{E}[ \int_0^t \lambda_s \, ds ] < \infty \) for all \( t \geq 0 \) then
\[
M_t = N_t - \int_0^t \lambda_s \, ds
\]
is a \( \mathcal{F}_t \)-martingale. □
Based on this observation we have the following important result.

**Proposition 6.2.1** If \( N_t \) is an integrable doubly stochastic Poisson process with \( \mathcal{F}_t \)-intensity \( \lambda_t \), then for all nonnegative \( \mathcal{F}_t \)-predictable processes \( C_t \)

\[
E \left[ \int_0^\infty C_s dN_s \right] = E \left[ \int_0^\infty C_s \lambda_s \, ds \right]
\]

where \( \int_0^t C_s \, dN_s := \sum_{n \geq 1} C_{T_n} \mathbb{1}(T_n \leq t) \). It turns out that the converse is also true. This was first proved by Watanabe in a less general setting.

**Proposition 6.2.2** (Watanabe (1964)) Let \( N_t \) be a point process adapted to the filtration \( \mathcal{F}_t \), and let \( \lambda(t) \) be a locally integrable nonnegative function. Suppose that \( N_t - \int_0^t \lambda_s \, ds \) is an \( \mathcal{F}_t \)-martingale.

Then \( N_t \) is Poisson process with intensity \( \lambda(t) \), that is, for all \( 0 \leq s \leq t \), \( N_t - N_s \) is a Poisson random variable with parameter \( \int_s^t \lambda_u \, du \) independent of \( \mathcal{F}_s \).

Motivated by this result we define the notion of stochastic intensity for an arbitrary point process as follows.

**Definition 6.2.3** (Stochastic Intensity)

Let \( N_t \) be a point process adapted to some filtration \( \mathcal{F}_t \), and let \( \lambda_t \) be a nonnegative \( \mathcal{F}_t \)-progressive process such that for all \( t \geq 0 \)

\[
\int_0^t \lambda_s \, ds < \infty \quad \mathcal{P} - \text{a.s.}
\]

If for all nonnegative \( \mathcal{F}_t \) predictable processes \( C_t \), the equality

\[
E \left[ \int_0^\infty C_s \, dN_s \right] = E \left[ \int_0^\infty C_s \lambda_s \, ds \right]
\]

is verified, then we say that \( N_t \) admits the \( \mathcal{F}_t \)-intensity \( \lambda_t \).

**Exercise 6.2.5** Let \( N_t \) be a point process with the \( \mathcal{F}_t \)-intensity \( \lambda_t \). Show that if \( \lambda_t \) id \( \mathcal{G}_t \)-progressive for some filtration \( \mathcal{G}_t \) such that \( \mathcal{F}_t^N \subseteq \mathcal{G}_t \leq \mathcal{F}_t \) \( t \geq 0 \) then \( \lambda_t \) is also the \( \mathcal{G}_t \)-intensity \( N_t \).

Similarly to the Poisson process, we can connect point processes with stochastic intensities to martingales.

**Proposition 6.2.3** (Integration Theorem)

If \( N_t \) admits the \( \mathcal{F}_t \)-intensity \( \lambda_t \) (where \( \int_0^t \lambda_s \, ds < \infty \) a.s.) then \( N_t \) is nonexplosive and

1. \( M_t = N_t - \int_0^t \lambda_s \, ds \) is an \( \mathcal{F}_t \)-local martingale.
2. if \( X_t \) is \( \mathcal{F}_t \)-predictable process such that \( E \left[ \int_0^t |X_s| \lambda_s \, ds \right] < \infty \) then \( \int_0^t X_s \, dM_s \) is an \( \mathcal{F}_t \)-martingale.
3. if \( X_t \) is \( \mathcal{F}_t \)-predictable process such that \( \int_0^t |X_s| \lambda_s \, ds < \infty \) a.s. then \( \int_0^t X_s \, dM_s \) is an \( \mathcal{F}_t \)-local martingale.
6.3 Optimal Intensity Control

In this section we study the problem of controlling a point process. In particular, we focus on the case where the controller can affect the intensity of the point process. This type of control differs from impulsive control where the controller has the ability to add or erase some of the point in the sequence.

We consider a point process \( N_t \) that we wish to control. The control \( u \) belongs to a set \( U \) of admissible controls. We will assume that \( U \) consists on the set of real-valued processes defined on \((\Omega, \mathcal{F})\) adapted to \( \mathcal{F}_t^N \) in addition for each \( t \in [0, T] \) we assume that \( u_t \in U_t \). In addition, for each \( u \in U \) the point process \( N_t \) admits a \((\mathcal{P}_u, \mathcal{F}_t)\)-intensity \( \lambda_t(u) \). Here, \( \mathcal{F}_t \) is some filtration associated to \( N_t \).

The performance measure that we will consider is given by

\[
J(u) = \mathbb{E}_u \left[ \int_0^T C_s(u) \, ds + \phi_T(u) \right].
\]

(6.3.1)

The expectation in \( J(u) \) above is taken with respect to \( \mathcal{P}_u \). The function \( C_t(u) \) is an \( \mathcal{F}_t \)-progressive process and \( \phi_T(u) \) is a \( \mathcal{F}_T \)-measurable random variable.

We will consider a problem with complete information so that \( \mathcal{F}_t \equiv \mathcal{F}_t^N \). In addition, we assume local dynamics

\[
\begin{align*}
  u_t &= u(t, N_t) \quad \text{is } \mathcal{F}_t^N \text{-predictable} \\
  \lambda_t(u) &= \lambda(t, N_t, u_t) \\
  C_t(u) &= C(t, N_t, u_t) \\
  \phi_T &= \phi_T(T, N_T).
\end{align*}
\]

Exercise 6.3.1 Consider the cost function

\[
J(u) = \mathbb{E} \left[ \sum_{0 < T_n \leq T} k_{T_n}(u) \right].
\]

Where \( k_t(u) \) is a nonnegative \( \mathcal{F}_t \)-measurable process. Show that this cost function can be written in the from given by equation (6.3.1). □

6.3.1 Dynamic Programming for Intensity Control

Theorem 6.3.1 (Hamilton-Jacobi Sufficient Conditions)

Suppose there exists for each \( n \in N_+ \) a differentiable bounded \( \mathcal{F}_t \)-progressive mapping \( V(t, \omega, n) \) such that all \( \omega \in \Omega \) and all \( n \in N_+ \)

\[
\frac{\partial}{\partial t} V(t, \omega, n) + \inf_{v \in U_t} \{ \lambda(t, \omega, n, v) \left[ V(t, \omega, n + 1) - V(t, \omega, n) \right] + C(t, \omega, n, v) \} = 0 \quad (6.3.2)
\]

\[
V(T, \omega, n) = \phi(T, \omega, n). \quad (6.3.3)
\]

and suppose there exists for each \( n \in N_+ \) an \( \mathcal{F}_t^N \)-predictable process \( u^*(t, \omega, n) \) such that \( u^*(t, \omega, n) \) achieves the minimum in equation (6.3.2). Then, \( u^* \) is the optimal control.
Exercise 6.3.2 Proof the theorem. □

This Theorem lacks of practical applicability because of Value Function is in general path dependent. The analysis can be greatly simplified if we assume that the problem is Markovian.

Corollary 6.3.1 (Markovian Control)
Suppose that $\lambda(t, \omega, n, v)$, $C(t, \omega, n, v)$, and $\phi(t, \omega, n)$ do not dependent on $\omega$ and that there is a function $V(t, n)$ such that
\[
\frac{\partial}{\partial t} V(t, n) + \inf_{v \in U_t} \{\lambda(t, n, v) [V(t, n + 1) - V(t, n)] + C(t, n, v)\} = 0 \quad (6.3.4)
\]
\[
V(T, n) = \phi(T, n). \quad (6.3.5)
\]
suppose that the minimum is achieved by a measurable function $U^*(t, n)$. Then, $u^*$ is the optimal control.

6.4 Applications to Revenue Management

In this section we present an application of point process optimal control based on the work by Gallego and van Ryzin (1994)\textsuperscript{1}.

6.4.1 Model Description and HJB Equation

Consider a seller that owns $I$ units of a product that wants to sell over a fixed time period $T$. Demand for the product is characterized by a point process $N_t$. Given a price policy $p_t$, $N_t$ admits the intensity $\lambda(p_t)$. The seller’s problem is to select a price strategy $\{p_t : t \in [0, T]\}$ (a predictable process) that maximizes the expected revenue over the selling horizon. That is,
\[
\max J_p(t, I) := \mathbb{E}_p \left[ \int_0^T p_s \, dN_s \right]
\]
subject to $\int_0^T dN_s \leq I$ $\mathcal{P}_p$ a.s.

In order to ensure that the problem is feasible we will assume that there exist a price $p_\infty$ such that $\lambda(p_\infty) = 0$ a.s. In the case, we define the set of admissible pricing policies $\mathcal{A}$, as the set of predictable $p(t, I - N_t)$ policies such that $p(t, 0) = p_\infty$. The seller’s problem becomes to maximize over $p \in \mathcal{A}$ the expected revenue $J_p(t, I)$.

Using corollary 6.3.1, we can write the optimality condition for this problem as follows:
\[
\frac{\partial}{\partial t} V(t, n) + \sup_p \{\lambda(p) [V(t, n) - V(t, n - 1)] - p \lambda(p)\} = 0
\]
\[
V(T, n) = 0.
\]

Let us make the following transformation of time $t \leftarrow T - t$, that is, $t$ measures the remaining selling time. Also, instead of looking at the price $p$ as the decision variable we use $\lambda$ as the control and

\textsuperscript{1}Optimal Dynamic Pricing of Inventories with Stochastic Demand over Finite Horizons, \textit{Mgmt. Sci.} 40, 999-1020.
p(\lambda) as the inverse demand function. The revenue rate \( r(\lambda) := \lambda p(\lambda) \) is assumed to be regular, i.e., continuous, bounded, concave, has a bounded maximizer \( \lambda^* = \min\{\lambda = \arg\max\{r(\lambda)\}\} \) and such that \( \lim_{\lambda \to 0} r(\lambda) = 0 \). Under this new definitions the HJB equation becomes

\[
\frac{\partial}{\partial t} V(t, n) = \sup_{\lambda} \{ r(\lambda) - \lambda [V(t, n) - V(t, n - 1)] \} = 0
\]

\( V(0, n) = 0 \).

**Proposition 6.4.1** If \( \lambda(p) \) is a regular demand function then there exists a unique solution to the HJB equation. Further, the optimal intensities satisfies \( \lambda^*(t, n) \leq \lambda^* \) for all \( n \) for all \( 0 \leq t \leq T \).

Closed-form solution to the HJB equation are generally intractable however, the optimality condition can be exploited to get some qualitative results about the optimal solution.

**Theorem 6.4.1** The optimal value function \( V^*(t, n) \) is strictly increasing and strictly concave in both \( n \) and \( t \). Furthermore, there exists an optimal intensity \( \lambda^*(n, t) \) that is strictly increasing in \( n \) and strictly decreasing in \( t \).

The following figure plots a sample path of price and inventory under an optimal policy.

![Sample Path of Price and Inventory](image)

Figure 6.4.1: Path of an optimal price policy and its inventory level. Demand is a time homogeneous Poisson process with intensity \( \lambda(p) = \exp(-0.1p) \), the initial inventory is \( C_0 = 20 \), and the selling horizon is \( H = 100 \). The dashed line corresponds to the minimum price \( p^{\text{min}} = 10 \).

### 6.4.2 Bounds and Heuristics

The fact that the HJB is intractable in most cases creates the need for alternative solution methods. One possibility, that we consider here, is the use of the *certainty equivalent* version of the problem.
That is, the deterministic control problem resulting from changing all uncertainty by its expected value. In this case, the deterministic version of the problem is given by

\[ V^D(T,n) = \max_\lambda \int_0^T r(\lambda_s) \, ds \]

subject to \[ \int_0^T \lambda_s \, ds \leq x. \]

The solution to this time-homogeneous problem can be found easily. Let \( \lambda^D(T,n) = \frac{n}{T} \), that is, the \( \text{run-out rate}. \) Then, it is straightforward to show that the optimal deterministic rate is \( \lambda^D = \min\{\lambda^*, \lambda^0(T,n)\} \) and the optimal expect revenue is \( V^D(T,n) = T \min\{r(\lambda^*), r(\lambda^0(T,n))\} \).

At least two things make the deterministic solution interesting.

**Theorem 6.4.2** If \( \lambda(p) \) is a regular demand function then for all \( n \geq 0 \) and \( t \geq 0 \)

\[ V^*(t,n) \leq V^D(t,n). \]

Thus, the deterministic value function provides an upper bound on the optimal expected revenue.

In addition, if we fixed the price at \( p^D \) and we denote by \( V^{FP}(t,n) \) the expected revenue collected from this fixed price strategy then we have the following important result.

**Theorem 6.4.3**

\[ \frac{V^{FP}(t,n)}{V^*(t,n)} \geq 1 - \frac{1}{2\sqrt{\min\{n, \lambda^* t\}}}. \]

Therefore, the fixed price policy is asymptotically optimal as the number of product \( n \) or the selling horizon \( t \) become large.
Chapter 7

Papers and Additional Readings