DDA6040
Dynamic Programming and Stochastic Control

Jingwei Zhang
School of Data Science
The Chinese University of Hong Kong, Shenzhen
zhangjingwei@cuhk.edu.cn

Lecture Notes[1]
(2025/2026, Semester 1)

Reference:
D. Bertsekas (2005). Dynamic Programming and Optimal Control. Athena Scientific, Boston, MA.
M. Puterman (2005). Markov Decisions Processes. Wiley, NJ.

[1]This notes is built based on materials including lecture notes from Prof. Rene Caldentey, Prof. Gustavo Vulcano and Prof. Peng Sun. Part of this material is based on the textbook by Dimitri Bertsekas, including a set of lecture notes publicly available online. Part of materials relating to approximate dynamic programming is based on notes by Prof. David Brown, Prof. Santiago Balseiro and Prof. Chen Chen. This notes is solely for your convenience. Please do not distribute!

Syllabus

Lecture Hours.
Monday (TA301) and Wednesday (TA301) 10:00 - 11:20 AM

Course Instructor.
Jingwei Zhang https://sites.google.com/view/jingwei-zhang
Email: zhangjingwei@cuhk.edu.cn
Office: Daoyuan 502B
Office Hours: Wednesday 3:30 PM - 4:30 PM

Teaching Assistant.
Surui Wang
Email: 224040338@link.cuhk.edu.cn

Course Description.
Dynamic Programming (DP) provides a set of general methods for making sequential, interrelated decisions under uncertainty. This course brings a new dimension to static models studied in the optimization course, by investigating dynamic systems and their optimization over time. The focus of the course is on modeling and deriving structural properties for discrete time, stochastic problems. The techniques are illustrated through concrete applications from Operations, Decision Sciences, Finance and Economics.
Prerequisites. An introductory course in Optimization and Probability.

Course Materials.

- D. Bertsekas (2005). Dynamic Programming and Optimal Control. Athena Scientific, Boston, MA.
- Lecture Notes prepared by the instructor to be distributed before the beginning of the class.
- M. Puterman (2005). Markov Decisions Processes. Wiley, NJ.

The course also includes some additional readings, mostly research papers that we will use to complement the material and discussion covered in class. Some of these papers described important applications of dynamic programming in Operations Management and other fields.

Course Assessment.
Assignments: 15%
Project: 15%
Midterm Exam: 35%

Final Exam: 35%

Course Contents.
The following is the list of sessions and topics that we will cover in this class. These topics serve as an introduction to Dynamic Programming. The coverage of the discipline is very selective: We concentrate on a small number of powerful themes that have emerged as the central building blocks in the theory of sequential optimization under uncertainty.

- Introduction to dynamic programming and optimal control (Informal)
- Classical model of dynamic programming (DP) in discrete time and finite time horizon, deterministic DP models and shortest path problem, different algorithms to find the shortest path, DP framework including uncertainty, Markov Decision process, properties of the value function and numerical methods.
- Infinite horizon and semi-Markov decision models.
- Fundamental properties and extensions of the classical DP model, Linear-Quadratic problem, connection between DP and supermodularity, state-space augmentation and the value of information.
- Optimality of (S, s) policies in a multi-period inventory control setting, single-leg multi-class revenue management problem, optimal stopping problem.
- Problems with inperfect information, efficient formulation of this problem, a sufficient set of statistics, revisit the LQ problem and review the Kalman filtering theory.
- Approximate dynamic programming, Lagrangian relaxation, information relaxation.

1 Optimal Control and Calculus of Variations (An Informal Introduction)

1.1 From Calculus to Calculus of Variations

In classical calculus, we are often interested in finding a solution to the optimization problem:

$$\min_{x \in \mathcal{X}} f(x),$$

where $\mathcal{X}$ is typically a finite-dimensional space, such as a subset of $\mathbb{R}^n$.

Let's recall the one-dimensional case where $\mathcal{X} = [a, b]$. If the function f is sufficiently smooth, we can find a minimum x^* by checking a few necessary conditions:

- 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 $x^* = b$.

Furthermore, we know that a solution is guaranteed to *exist* if f is continuous on $[a, b]$, and this solution is *unique* if f is strictly convex.

But what if the "thing" we want to optimize is not a point x in $\mathbb{R}^n$, but an entire function or curve? This question launches us from the world of calculus into the calculus of variations. Instead of choosing the best point, we will choose the best path.

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: Imagine a bead sliding under gravity from a point A to a lower point B. What is the shape of the wire it should slide on to make the journey in the shortest possible time (see Figure 1.1)? The name itself comes from Greek: brachistos (shortest) + chronos (time).

Johann issued this as a challenge to the mathematicians of Europe, secretly hoping to stump his older brother and fierce rival, Jakob Bernoulli. The challenge ignited the greatest minds of the era. Solutions came from Leibniz, L'Hôpital, and even Isaac Newton. Newton reportedly received the problem after a long day at the Royal Mint, solved it in a single night, and mailed his solution back anonymously. When Johann saw the elegant solution, he supposedly declared, "Tanquam ex ungue leonem" – "I recognize the lion by his claw."

Ultimately, five solutions were found, including one by Jakob, who not only solved the problem but also developed foundational methods for this new class of problems, laying the groundwork for the calculus of variations. The optimal curve, by the way, is a cycloid – the path traced by a point on the rim of a rolling wheel.

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).

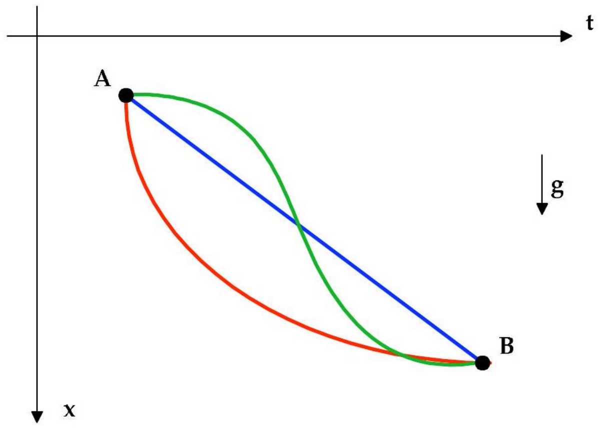


Figure 1.1: The Brachistochrone problem: Find the curve which would provide the faster time of transit to a particle sliding down it from Point A to Point B under the action of gravity.

The most fundamental problem in the calculus of variations is to find a function $x(t)$ that minimizes a functional, $J(x)$. A functional is simple a "function of a function" – it takes a function as input and returns a scalar value. The typical form is:

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

Here:

- $x(t)$ is the function or path we are trying to find.
- $\dot{x}(t) = \frac{\mathrm{d}}{\mathrm{d}t}x(t)$ is its time derivative.
- The function L is called the Lagrangian or the integrand. It defines the quantity to be accumulated over time (e.g., time, cost or energy). We assume L is at least twice differentiable.

Example 1.1 (Geodesic Problem)**.** Find the shortest path between two points in a plane. Here, $L(t, x(t), \dot{x}(t)) = \sqrt{1 + \dot{x}(t)^2}$.

Example 1.2 (Minimal Surface of Revolutions)**.** Find the curve that generates the smallest surface area when revolved around the t-axis. Here, $L(t, x(t), \dot{x}(t)) = 2\pi x(t)\sqrt{1 + \dot{x}(t)^2}$.

We need to define the set of "allowed" functions. A function $x(t)$ is piecewise C^1 on $[a, b]$ if it is continuous and its derivative $\dot{x}(t)$ is continuous everywhere except a finite number of points. We consider the problem where the start and end points of the path are fixed:

$$\min_{x(\cdot)} J(x), \text{ s.t. } x(a) = x_a,\ x(b) = x_b.$$

Example 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 μ 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.

1.2 Continuous-Time Optimal Control

The Optimal Control problem that we briefly introduce in this section 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):

  $$\dot{x}(t) = f(t, x(t), u(t)), \quad 0 \le t \le T, \text{given } x(0),$$

  that is,

  $$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = f_i(t, x(t), u(t)), \forall i = 1, \ldots, n,$$

  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 for an illustration of a control path and the associated state path.

– The control trajectory $\{u(t) \mid t \in [0, T]\}$ uniquely determines the state trajectory $\{x^u(t) \mid t \in [0, T]\}$. 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.

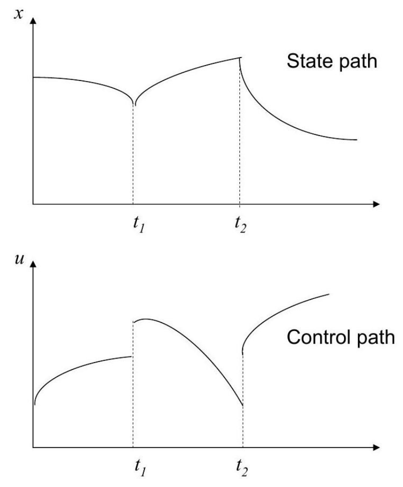


Figure 1.2: Control and state paths for a continuous-time optimal control problem under the regular assumptions.

$$\begin{array}{rll}
\text{Lagrange Problem:} & \min\limits_{u\in\mathcal{U}} J(x) = \int_0^T g(t,x(t),u(t))\mathrm{d}t & \\
\text{subject to} & \dot{x}(t) = f(t,x(t),u(t)), x(0) = x_0 & \text{(system dynamics)} \\
& \phi(x(T)) = 0 & \text{(boundary conditions)}
\end{array}$$

$$\begin{array}{rll}
\text{Mayer Problem:} & \min\limits_{u\in\mathcal{U}} h(x(T)) & \\
\text{subject to} & \dot{x}(t) = f(t,x(t),u(t)), x\,(0) = x_0 & \text{(system dynamics)} \\
& \phi(x(T)) = 0 & \text{(boundary conditions)}
\end{array}$$

$$\begin{array}{rll}
\text{Bolza Problem:} & \min\limits_{u\in\mathcal{U}} h(x(T)) + \int_0^T g(t,x(t),u(t))\mathrm{d}t & \\
\text{subject to} & \dot{x}(t) = f(t,x(t),u(t)), x\,(0) = x_0 & \text{(system dynamics)} \\
& \phi(x(T)) = 0 & \text{(boundary conditions)}
\end{array}$$

The functions f, h, g and ϕ are normally assumed to be continuously differentiable with respect to x; and f, g are continuous with respect to t and u.

Remark 1.1. All three versions of the optimal control problem are equivalent.

Example 1.4 (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 $|u(t)| \leq 1, \quad \forall t \in [0, T]$.
- 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 1.5 (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:

$$\dot{x}(t) = \gamma u(t)x(t), \text{ where } \gamma > 0 \text{ is the reinvestment benefit, } u(t) \in [0,1]$$

- Problem: The producer wants to maximize the total amount of product stored

$$\max_{u(t)\in[0,1]} \int_0^T (1-u(t))x(t)\mathrm{d}t,$$

  assuming $x(0)$ is given.

2 Discrete Dynamic Programming

Dynamic programming (DP) is a technique pioneered by Richard Bellman[2] in the 1950 's to model and solve problems where decisions are made in stages[3] 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 previously 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 (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 satisfied a (possibly stochastic) demand. The dynamics of this inventory system are depicted in Figure 2.1. 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.

Assumptions:

1. Leadtime $= 0$ (i.e., instantaneous replenishment)
2. Independent demands $w_0, w_1, \ldots, w_{N-1}$
3. Fully backlogged demand

[2]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 .

[3]For the most part we will consider applications in which these different stages correspond to different moments in time.

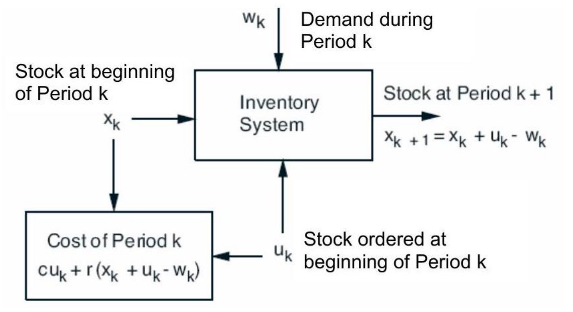

w_k
Demand during Period k
Stock at beginning of Period k
x_k
Inventory System
Stock at Period k + 1
$x_{k+1} = x_k + u_k - w_k$
Cost of Period k
$cu_k + r(x_k + u_k - w_k)$
u_k
Stock ordered at beginning of Period k

Figure 2.1: System dynamics for the Inventory Control problem.

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.

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

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

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

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 $P_k(\cdot \mid x_k, u_k)$
- 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

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

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, 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 random parameters $(w_0, w_1, \ldots, w_{N-1})$ 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 π, we can associate a trajectory $x^\pi = (x_0^\pi, x_1^\pi, \ldots, x_N^\pi)$ 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) over time when the policy π has been chosen. Note that in genera x^π is a stochastic process. The corresponding performance of policy $\pi = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$ is given by

$$J_\pi = \mathbb{E}_{w_0, w_1, \ldots, w_{N-1}} \left[g_N\left(x_N^\pi\right) + \sum_{k=0}^{N-1} g_k\left(x_k^\pi, \mu_k\left(x_k^\pi\right), w_k\right) \right].$$

If the initial state is fixed, i.e., $x_0^\pi = x_0$ for all feasible policy π then we denote the performance of policy π by $J_\pi\left(x_0\right)$.

The objective of dynamic programming is to optimize J_π over all policies π 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:

$$p_{ij}(u, k) = \mathbb{P}\left\{x_{k+1} = j \mid x_k = i, u_k = u\right\}.$$

Remark 2.1. Transition probabilities $\Longleftrightarrow$ System equation.

Proof. $\Rightarrow$) Given a transition probability representation,

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

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

$$x_{k+1} = w_k, \qquad \text{where } \mathbb{P}\left\{w_k = j \mid x_k = i, u_k = u\right\} = p_{ij}(u, k).$$

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

$$p_{ij}(u, k) = \mathbb{P}_k\left\{W_k(i, u, j) \mid x_k = i, u_k = u\right),$$

where the event W_k is defined as

$$W_k(i, u, j) = \left\{w \mid j = f_k(i, u, w)\right\}.$$

□

Example 2.2 (Scheduling)**.**

- Objective: Find the optimal sequence of operations A, B, C, D to produce a certain product.

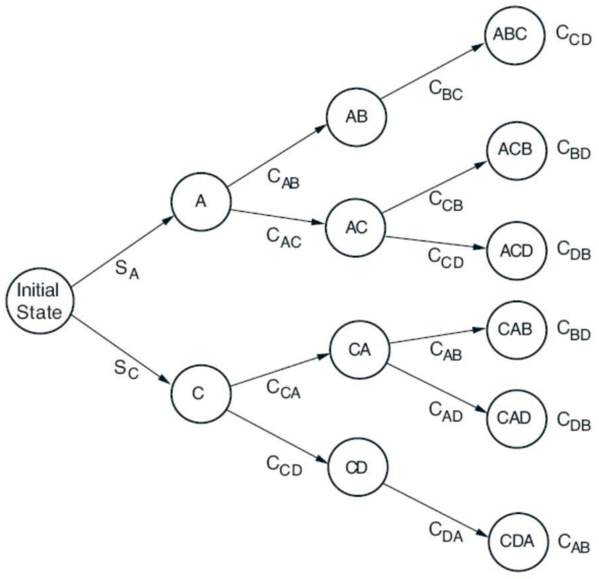


Figure 2.2: System dynamics for the Scheduling problem.

- Precedence constraints: $A \longrightarrow B, C \longrightarrow D$.
- 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.

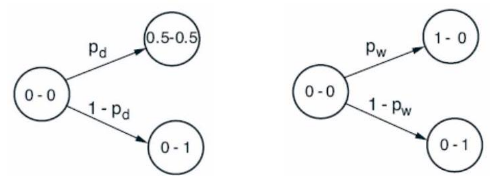
This example is represented in Figure 2.2. The optimal solution is described by a path of minimum cost that starts at the initial state and ends at some state at the terminal time. The cost of a path is the sum of the labels in the arcs plus the terminal cost (label in the leaf).

Example 2.3 (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").



(a) Timid Play (b) Bold Play

Figure 2.3: Transition probability graph for period $k = 0$ for the chess match.

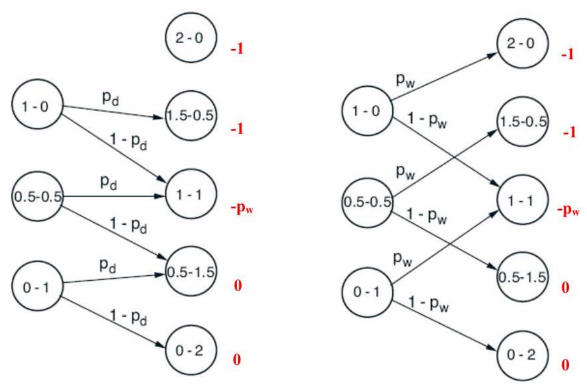
- State: vector with the two scores attained so far. It could also be the net score (difference between the scores).
- Each player has two playing styles, and can choose one of the two at will in each game:
  - Timid play: draws with probability $p_d > 0$, and loses w.p. $1 - p_d$.
  - Bold play: wins 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 2.3 we show the transition probabilities for period $k = 0$. In Figure 2.4 we show the transition probabilities for the second stage of the match (i.e., $k = 1$), and the cost of the terminal states. 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.

Theorem 2.1. 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 $\{u_0, u_1, \ldots, u_{N-1}\}$.



(a) Timid Play (b) Bold Play

Figure 2.4: Transition probability graph for period $k = 1$ for the chess match.

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

$$x_{k+1} = f_k\left(x_k, \mu\left(x_k\right)\right), \quad k = 0, 1, \ldots, N-1,$$

and the corresponding controls are perfectly predictable through the equation

$$u_k = \mu_k\left(x_k\right), \quad k = 0, 1, \ldots, N-1,$$

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 2.5), where:

- States $\Longleftrightarrow$ Nodes
- Each control applied over a state $x_k \Longleftrightarrow$ 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) $\Longleftrightarrow$ paths from initial state s to terminal states

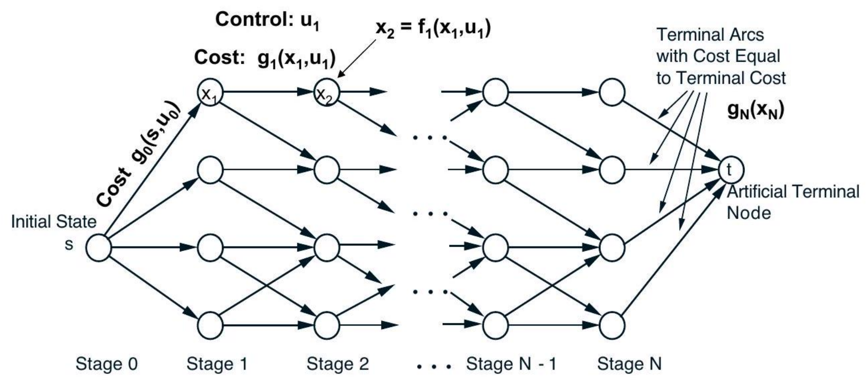


Figure 2.5: Construction of a transition graph for a deterministic finite-state system.

- Final stage $\Longleftrightarrow$ Artificial terminal node t
- Each state x_N at stage N $\Longleftrightarrow$ Connected to the terminal node t with an arc having cost $g_N(x_N)$
- One-step costs $g_k(i, u_k)$ $\Longleftrightarrow$ Cost of an arc a_{ij}^k (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}^N as the terminal cost of state $i \in S_N$.

  Assume $a_{it}^k = \infty$ if there is no control that drives from i to j.
- Cost of control sequence $\Longleftrightarrow$ 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:

$$J_N(i) = a_{it}^N, i \in S_N,$$
$$J_k(i) = \min_{j \in S_{k+1}} \left\{ a_{ij}^k + J_{k+1}(j) \right\}, i \in S_k, k = 0, 1, \ldots, N-1.$$

The optimal cost is $J_0(s)$ and equal to the length of the shortest path from s to t.

- Observation: An optimal path $s \longrightarrow t$ is also an optimal path $t \longrightarrow s$ 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:

$$\begin{aligned}
&\tilde{J}_N(j) = a_{sj}^0, j \in S_1, \\
&\tilde{J}_k(j) = \min_{i \in S_{N-k}} \left\{ a_{ij}^{N-k} + \tilde{J}_{k+1}(i) \right\}, j \in S_{N-k+1}, k = 0, 1, \ldots, N-1.
\end{aligned}$$

  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.

2.3 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:

$$\begin{aligned} J_k(i) =& \text{ Optimal cost of getting from } i \text{ to } t \text{ in } N-k \text{ moves,} \\ J_0(i) =& \text{ Cost of the optimal path from } i \text{ to } t \text{ in } N \text{ moves.} \end{aligned}$$

- DP algorithm:

$$J_k(i) = \min_{j=1,2,\ldots,N} \{a_{ij} + J_{k+1}(j)\}, \quad k = 0, 1, \ldots, N-2,$$

  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

$$j^* = \underset{1 \leq j \leq N}{\operatorname{argmin}} \{a_{ij} + J_{k+1}(j)\},$$

- 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 2.6 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 $t = 5$:

$$\begin{aligned} J_3(1) =& \text{ cost of getting from node 1 to node } 5 = 2 \\ J_3(2) =& \text{ cost of getting from node 2 to node } 5 = 7 \\ J_3(3) =& \text{ cost of getting from node 3 to node } 5 = 5 \\ J_3(4) =& \text{ cost of getting from node 4 to node } 5 = 3 \end{aligned}$$

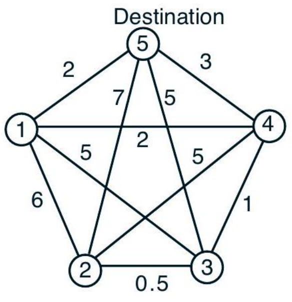


Figure 2.6: Shortest path problem data. There are $N = 4$ states, and a destination node $t = 5$.

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

$$J_2(1) = \min_{j=1,2,3,4} \{a_{1j} + J_3(j)\} = 2$$
$$J_2(2) = \min_{j=1,2,3,4} \{a_{2j} + J_3(j)\} = 5.5$$
$$J_2(3) = \min_{j=1,2,3,4} \{a_{3j} + J_3(j)\} = 4$$
$$J_2(4) = \min_{j=1,2,3,4} \{a_{4j} + J_3(j)\} = 3$$

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

$$J_1(1) = \min_{j=1,2,3,4} \{a_{1j} + J_2(j)\} = 2$$
$$J_1(2) = \min_{j=1,2,3,4} \{a_{2j} + J_2(j)\} = 4.5$$
$$J_1(3) = \min_{j=1,2,3,4} \{a_{3j} + J_2(j)\} = 4$$
$$J_1(4) = \min_{j=1,2,3,4} \{a_{4j} + J_2(j)\} = 3$$

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

$$J_0(1) = \min_{j=1,2,3,4} \{a_{1j} + J_1(j)\} = 2$$
$$J_0(2) = \min_{j=1,2,3,4} \{a_{2j} + J_1(j)\} = 4.5$$
$$J_0(3) = \min_{j=1,2,3,4} \{a_{3j} + J_1(j)\} = 4$$
$$J_0(4) = \min_{j=1,2,3,4} \{a_{4j} + J_1(j)\} = 3$$

Figure 2.7 shows the outcome of the shortest path (DP-type) algorithm applied over the graph in Figure 2.6.

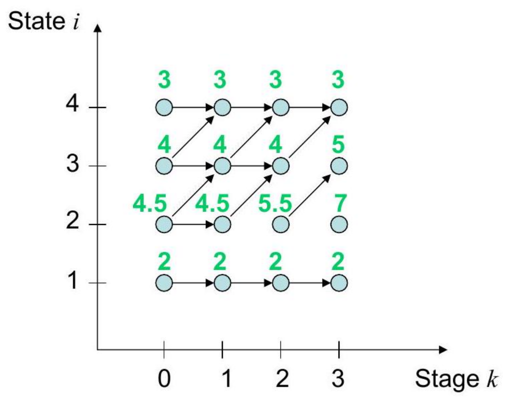


Figure 2.7: Outcome of the shortest path algorithm. The arcs represent the optimal control to follow from a given state i (node i in the graph) at a particular stage k (where stage k means $4-k$ transitions left to reach $t=5$). When there is more than one arc going out of a node, it represents the availability of more than one optimal control. The label next to each node shows the cost-to-go starting at the corresponding (stage, state) position.

2.3.1 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 π_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.[4]
- 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 $p\left(X_N \mid Z_N\right)$ 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

[4]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.

is an estimated sequence. We do not observe the true sequence, but just a proxy for it given by Z_N.

Viterbi algorithm. We know from conditional probability that

$$\mathbb{P}\left(X_N \mid Z_N\right) = \frac{\mathbb{P}\left(X_N, Z_N\right)}{\mathbb{P}\left(Z_N\right)},$$

for unconditional probabilities $\mathbb{P}\left(X_N, Z_N\right)$ and $\mathbb{P}\left(Z_N\right)$. Since $\mathbb{P}\left(Z_N\right)$ is a positive constant once Z_N is known, we can just maximize $\mathbb{P}\left(X_N, Z_N\right)$, where

$$\begin{aligned}
\mathbb{P}\left(X_N, Z_N\right) &= \mathbb{P}\left(x_0, x_1, \ldots, x_N, z_1, z_2, \ldots, z_N\right) \\
&= \pi_{x_0} \mathbb{P}\left(x_1, \ldots, x_N, z_1, z_2, \ldots, z_N \mid x_0\right) \\
&= \pi_{x_0} \mathbb{P}\left(x_1, z_1 \mid x_0\right) \mathbb{P}\left(x_2, \ldots, x_N, z_2, \ldots, z_N \mid x_0, x_1, z_1\right) \\
&= \pi_{x_0} p_{x_0 x_1} r\left(z_1; x_0, x_1\right) \qquad \underbrace{\mathbb{P}\left(x_2, \ldots, x_N, z_2, \ldots, z_N \mid x_0, x_1, z_1\right)}_{\substack{\mathbb{P}\left(x_2, z_2 \mid x_0, x_1, z_1\right) \mathbb{P}\left(x_3, \ldots, x_N, z_3, \ldots, z_N \mid x_0, x_1, z_1, x_2, z_2\right) \\ = p_{x_1} x_2 r\left(z_2; x_1, x_2\right) \mathbb{P}\left(x_3, \ldots, x_N, z_3, \ldots, z_N \mid x_0, x_1, z_1, x_2, z_2\right)}} \qquad .
\end{aligned}$$

Continuing in the same manner we obtain:

$$\mathbb{P}\left(X_N, Z_N\right) = \pi_{x_0} \prod_{k=1}^{N} p_{x_{k-1} x_k} r\left(z_k; x_{k-1}, x_k\right).$$

Instead of working with this function, we will maximize $\log \mathbb{P}\left(X_N, Z_N\right)$, or equivalently:

$$\min_{x_0, x_1, \ldots, x_N} \left\{ -\log\left(\pi_{x_0}\right) - \sum_{k=1}^{N} \log\left(p_{x_{k-1} x_k} r\left(z_k; x_{k-1}, x_k\right)\right) \right\}.$$

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 2.8 as follows:

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

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 $D_k\left(x_k\right)$ from s to all states x_k,

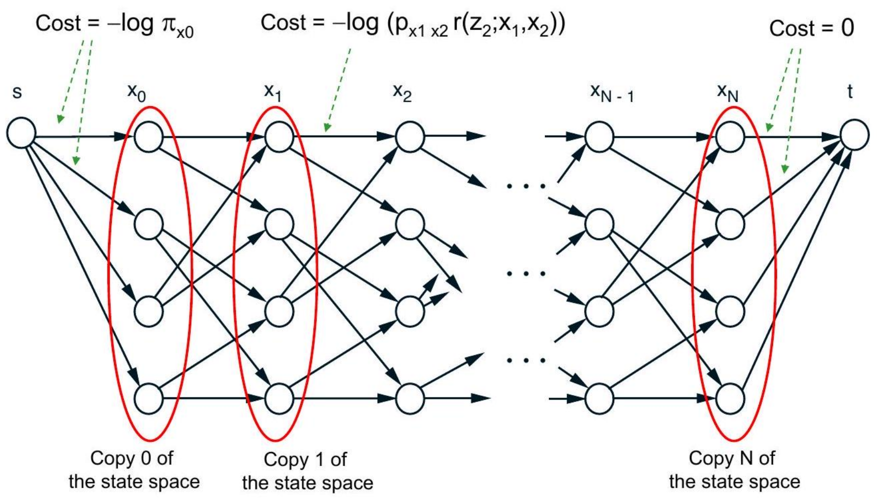

Cost = –log π_{x0}
Cost = –log ($p_{x1\,x2}\,r(z_2;x_1,x_2)$)
Cost = 0
s
x_0
x_1
x_2
x_{N-1}
x_N
t
Copy 0 of the state space
Copy 1 of the state space
Copy N of the state space

Figure 2.8: 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 space state (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 $p_{x_{k-1}x_k} > 0$.

on the basis of the observation sequence $z_1, \ldots, z_k$, and suppose that we observe z_{k+1}. Then for any $k = 1, \ldots, N-1$,

$$D_{k+1}(x_{k+1}) = \min_{x_k : p_{x_k x_{k+1}} > 0} \left\{ D_k(x_k) - \log\left(p_{x_k x_{k+1}} r(z_{k+1}; x_k, x_{k+1})\right) \right\},$$

starting from $D_0(x_0) = -\log \pi_{x_0}$.
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.
- 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 $\hat{X}_N = \{\hat{x}_1, \ldots, \hat{x}_N\}$

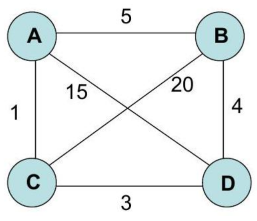


Figure 2.9: Basic graph for the TSP example with four cities.

that maximizes over all possible sequences $X_N = \{x_1, \ldots, x_N\}$ the conditional probability $\mathbb{P}(X_N \mid Z_N)$.

  * 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.

2.3.2 Shortest path algorithms

Computational implications of the equivalence shortest path problems $\Longleftrightarrow$ 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 2.4 (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 2.9. 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, $n \leq N$.

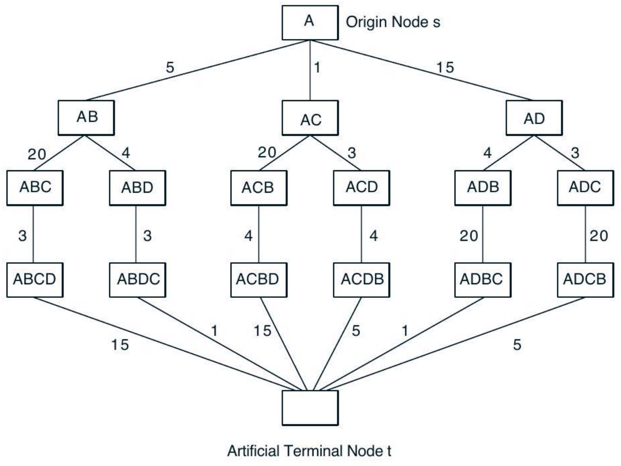


Figure 2.10: Structure of the shortest path execution graph for the TSP example.

- 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 2.10 shows the construction of the execution graph for the example described in Figure 2.9.

2.3.3 Alternative shortest path algorithms: Label correcting methods

Working on the shortest path execution graph as the one in Figure 2.10, 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 $a_{ij} \geq 0$.
- 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.

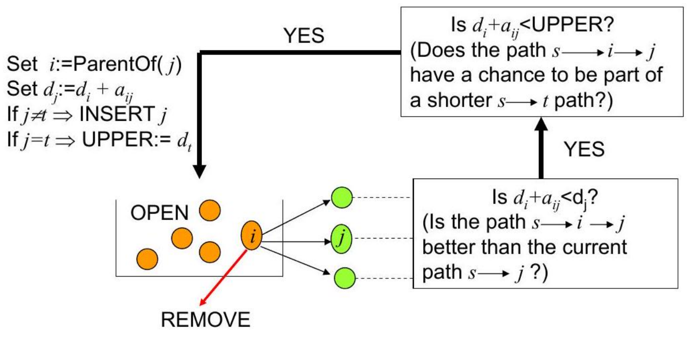

Set i:=ParentOf(j)
Set d_j:=d_i + a_{ij}
If $j \neq t \Rightarrow$ INSERT j
If $j=t \Rightarrow$ UPPER:= d_t
YES
Is d_i+a_{ij}<UPPER?
(Does the path $s \to i \to j$ have a chance to be part of a shorter $s \to t$ path?)
YES
OPEN
i
j
REMOVE
Is d_i+a_{ij}<d_j?
(Is the path $s \to i \to j$ better than the current path $s \to j$?)

Figure 2.11: Sketch of the Label Correcting Algorithm.

- 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).

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

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

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

Sp 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 2.11 sketches the Label Correcting Algorithm.

The execution of the algorithm over the TSP example above is represented in Figure 2.12 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.

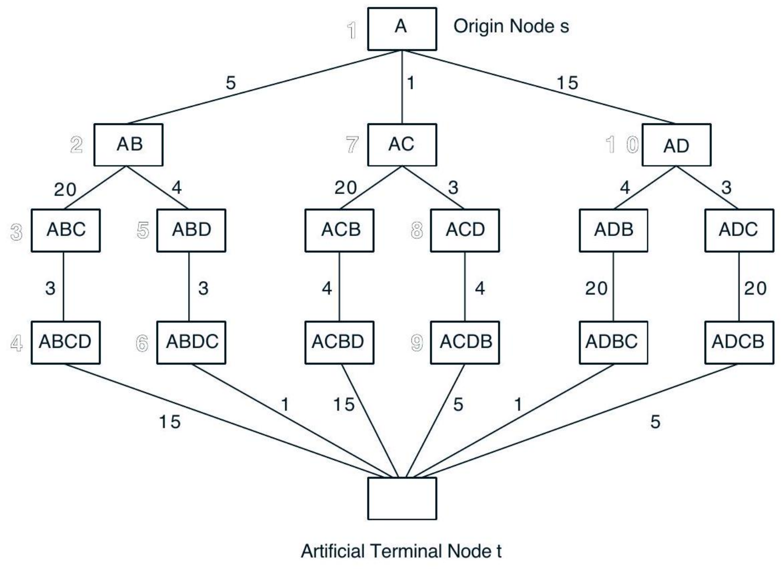


Figure 2.12: Labeling of the nodes in the execution graph when executing the LCA corresponds to the iteration of the LCA.

Iter No.	Node exiting OPEN	Label update / Observations	Status after iteration	
			OPEN	UPPER
0	-	-	{1}	∞
1	1	$d_2 := 5, d_7 := 1, d_{10} := 15, \text{ParentOf}(2,7,10) := 1$	{2,7,10}	∞
2	2	$d_3 := 25, d_5 := 9, \text{ParentOf}(3,5) := 2$	{3,5,7,10}	∞
3	3	$d_4 := 28.\text{ParentOf}(4) := 3$	{4,5,7,10}	∞
4	4	Reached terminal node t. Set $d_t := 43.\text{ParentOf}(t) := 4$.	{5,7,10}	43
5	5	$d_6 := d_5 + 3 = 12$. ParentOf(6) := 5	$\{6, 7, 10\}$	43
6	6	Reached terminal node t. Set $d_t := 13$. ParentOf(t) := 6.	$\{7, 10\}$	13
7	7	$d_8 := d_7 + 3 = 4$. Node ABC would have a label $d_7 + 20 = 21 >$ UPPER. So, it does not enter OPEN. ParentOf(8) := 7	$\{8, 10\}$	13
8	8	$d_9 := 8$. ParentOf(9) := 8	$\{9, 10\}$	13
9	9	$d_9 + a_{9t} = d_9 + 5 = 13 \geq$ UPPER	10	13
10	10	If picked ADB: $d_{10} + 4 = 19 >$UPPER. If picked ADC: $d_{10} + 3 = 18 >$UPPER.	Empty	13

Table 2.1: The optimal solution ABCD is found after examining nodes 1 through 10 in Figure 2.12, 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. 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 $= \infty$.

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 $s \to t$.

   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 ∞.

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_{j_k} 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 $d_t = \underbrace{d_{j_k} + a_{j_k t}}_{d^*}$, and UPPER $= d^*$.

   Similarly, node j_{k-1} will never enter OPEN with $d_{j_{k-1}}$ 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_{sj_1}. However, this happens at the first iteration of the algorithm, leading to a contradiction. Therefore, UPPER will be equal to the shortest distance $s \to t$.

□

Specific Label Correcting Methods.
Making the method efficient:

- Reduce the value of UPPER as quickly as possible (i.e., try to discover "good" $s \to t$ 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 Djikstra 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: $= \{i \neq t \mid d_i < \infty\}$.

- 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.

2.4 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

  $$x_{k+1} = f_k\left(x_k, u_k, w_k\right), k = 0, 1, \ldots, N-1,$$

  where

  - the state x_k is an element of a space S_k,
  - the control u_k verifies $u_k \in U_k\left(x_k\right) \subset C_k$, 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 $P_k\left(\cdot \mid x_k, u_k\right)$ 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 μ_k maps states x_k into controls $u_k = \mu_k\left(x_k\right)$, and is such that $\mu_k\left(x_k\right) \in U_k\left(x_k\right)$ 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

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

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

- The expected cost of a policy π starting at x_0 is

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

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

An optimal policy π^* is one that minimizes this cost; that is,

$$J_{\pi^*}\left(x_0\right) = \min_{\pi \in \Pi} J_\pi\left(x_0\right), \tag{2.1}$$

where Π 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, π^* is typically independent of x_0, because

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

  and $\mu_0^*(x_0)$ must be defined for all x_0.
- Even though we will be using the "min" operator in (2.1), it should be understood that the correct formal formulation should be $J_{\pi^*}(x_0) = \inf_{\pi\in\Pi} J_\pi(x_0)$.

Example 2.5 (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.

  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 $0 \leq k \leq N-1, g_k(i, u_k, w_k) = r(i) + c_f \mathbb{1}\{u_k = u_f\} + c_s \mathbb{1}\{u_k = u_f\}$.[5] For $k = N, g_N(i) = R(i)$.

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

- If the system is empty, then:

$$p_{0j}(u_f) = p_{0j}(u_s) = p_j, j = 0, 1, \ldots, n-1; \text{ and } p_{0n}(u_f) = p_{0n}(u_s) = \sum_{m=n}^{\infty} p_m$$

  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

$$\begin{aligned}
&p_{ij}(u_f) = 0, \text{ if } j < i-1 \\
&\text{(cannot have more than one service completion per period)}, \\
&p_{ij}(u_f) = q_f p_0, \text{ if } j = i-1 \\
&\text{(current customer finishes in this period and nobody arrives)}, \\
&p_{ij}(u_f) = \mathbb{P}\{j-i+1 \text{ arrivals, service completed}\} \\
&\qquad + \mathbb{P}\{j-i \text{ arrivals, service not completed}\} \\
&\qquad = q_f p_{j-i+1} + (1-q_f) p_{j-i}, \text{ if } i-1 < j < n-1, \\
&p_{i(n-1)}(u_f) = \mathbb{P}\{\text{at least } n-i \text{ arrivals, service completed}\} \\
&\qquad + \mathbb{P}\{n-1-i \text{ arrivals, service uncompleted}\} \\
&\qquad = q_f \sum_{m=n-i}^{\infty} p_m + (1-q_f) p_{n-1-i}, \\
&p_{in}(u_f) = \mathbb{P}\{\text{at least } n-i \text{ arrivals, service not completed}\} \\
&\qquad = (1-q_f) \sum_{m=n-i}^{\infty} p_m
\end{aligned}$$

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

[5]Here, $\mathbb{1}\{A\}$ is the indicator function, taking the value one if event A occurs, and zero otherwise.

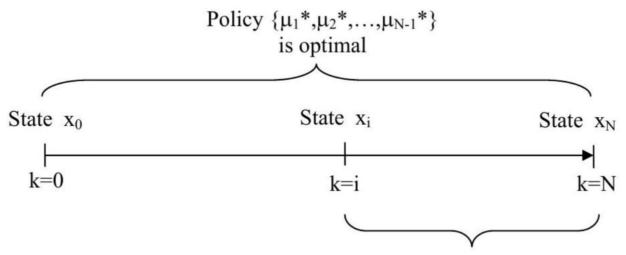


Figure 2.13: Principle of Optimality: The tail policy is optimal for the tail subproblem.

2.5 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 $\pi^* = \left\{\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\right\}$ be an optimal policy for the basic problem, and assume that when using π^* 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,

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

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

Figure 2.13 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 2.2. Consider the graph of costs for previous Example 2.2 given in Figure 2.14. Applying the DP algorithm from the terminal nodes (stage $N = 3$), and proceeding backwards, we get the representation in Figure 2.15. 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. Consider again the stochastic inventory problem described in Example 2.1. 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.

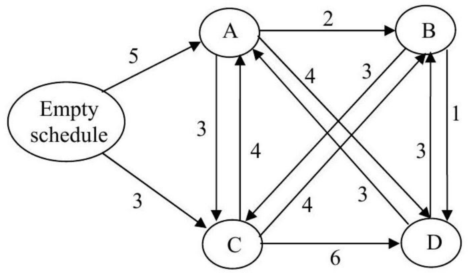


Figure 2.14: Graph of one-step switching costs for Example 2.2.

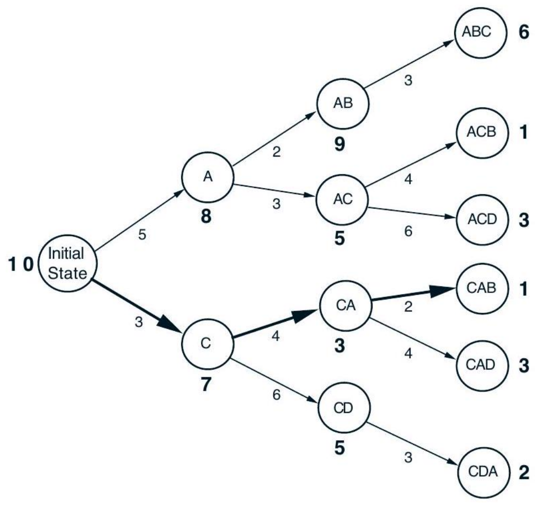


Figure 2.15: Transition graph for Example 2.2 Next to each node/state we show the cost of optimally completing the scheduling starting from that state. This is the optimal cost of the corresponding tail subproblem. The optimal cost for the original problem is equal to 10. The optimal schedule is $CABD$.

- Tail subproblems of length 1: The optimal cost for the last period is

$$J_{N-1}\left(x_{N-1}\right)=\min_{u_{N-1}\geq 0}\mathbb{E}_{w_{N-1}}\left[cu_{N-1}+r\left(x_{N-1}+u_{N-1}-w_{N-1}\right)\right].$$

  Note:

  - $u_{N-1}=\mu_{N-1}\left(x_{N-1}\right)$ depends on x_{N-1}.
  - $J_{N-1}\left(x_{N-1}\right)$ 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

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

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

- The value $J_0\left(x_0\right)$ 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

$$J_N\left(x_N\right)=g_N\left(x_N\right),$$

and go backwards using the recursion: for $k=0,1,\ldots,N-1$,

$$J_k\left(x_k\right)=\min_{u_k\in U_k(x_k)}\mathbb{E}_{w_k}\left[g_k\left(x_k,u_k,w_k\right)+J_{k+1}\left(f_k\left(x_k,u_k,w_k\right)\right)\right], \tag{2.2}$$

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

Proposition 2.3. For every initial state x_0, the optimal cost $J^*\left(x_0\right)$ of the basic problem is equal to $J_0\left(x_0\right)$, given by the last step of the DP algorithm. Furthermore, if $u_k^*=\mu_k^*\left(x_k\right)$ minimizes the RHS of (2.2) for each x_k and k, the policy $\pi^*=\left\{\mu_0^*,\ldots,\mu_{N-1}^*\right\}$ is optimal.

Observations:

- Justification: Proof by induction that $J_k\left(x_k\right)$ is equal to $J_k^*\left(x_k\right)$, 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 $\sum_{k=0}^{N-1}\left|S_k\right|\left|U_k\right|$, where $\left|S_k\right|$ is the size of the state space in period k, and $\left|U_k\right|$ 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 2.3. 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 π
- 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 $J_{k+1}(x_{k+1})$ 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

  $$\begin{aligned}
  &J_k^*(x_k)\\
  &= \min_{(\mu_k,\pi_{k+1})} \mathbb{E}_{w_k,w_{k+1},\ldots,w_{N-1}} \left[g_k(x_k, \mu_k(x_k), w_k) + g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right]\\
  &= \min_{\mu_k} \mathbb{E}_{w_k} \left[g_k(x_k, \mu_k(x_k), w_k) + \min_{\pi_{k+1}} \mathbb{E}_{w_{k+1},\ldots,w_{N-1}} \left[g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right]\right]\\
  &\quad \text{(this is the "informal step", since we are moving the min inside the } \mathbb{E}[\cdot]\\
  &= \min_{\mu_k} \mathbb{E}_{w_k} \left[g_k(x_k, \mu_k(x_k), w_k) + J_{k+1}^*(f_k(x_k, \mu_k(x_k), w_k)) \right] \quad \left(\text{by def. of } J_{k+1}^*\right)\\
  &= \min_{\mu_k} \mathbb{E}_{w_k} \left[g_k(x_k, \mu_k(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k(x_k), w_k)) \right] \quad \text{(by induction hypothesis)}\\
  &= \min_{u_k \in U_k(x_k)} \mathbb{E}_{w_k} \left[g_k(x_k, \mu_k(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k(x_k), w_k)) \right]\\
  &= J_k(x_k),
  \end{aligned}$$

  where the second to last equality follows from converting a minimization problem over functions μ_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.

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

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

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 $\mu_k^\epsilon(x_k) \in U_k(x_k)$ for the DP recursion (2.2) such that

$$\mathbb{E}_{w_k}\left[g_k(x_k, \mu_k^\epsilon(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k^\epsilon(x_k), w_k))\right] \leq J_k(x_k) + \epsilon. \tag{2.3}$$

Because of our former assumption, $J_k(x_k)$ generated by the DP algorithm is well defined and finite for all k and $x_k \in S_k$. Let $J_k^\epsilon(x_k)$ be the expected cost when using the policy $\left\{\mu_k^\epsilon, \ldots, \mu_{N-1}^\epsilon\right\}$. We will show by induction that for all x_k and k, it must hold that

$$J_k(x_k) \leq J_k^\epsilon(x_k) \leq J_k(x_k) + (N-k)\epsilon \tag{2.4}$$

$$J_k^*(x_k) \leq J_k^\epsilon(x_k) \leq J_k^*(x_k) + (N-k)\epsilon \tag{2.5}$$

$$J_k(x_k) = J_k^*(x_k) \tag{2.6}$$

- For $k = N-1$, we have

  $$J_{N-1}(x_{N-1}) = \min_{u_{N-1} \in U_{N-1}(x_{N-1})} \mathbb{E}_{w_{N-1}}\left[g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) + g_N(x_N)\right],$$
  $$J_{N-1}^*(x_{N-1}) = \min_{\pi^{N-1}} \mathbb{E}_{w_{N-1}}\left[g_{N-1}(x_{N-1}, \mu_{N-1}(x_{N-1}), w_{N-1}) + g_N(x_N)\right].$$

  Both minimizations guarantee the LHS inequalities in (2.4) and (2.5) when comparing versus

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

  with $\mu_{N-1}^\epsilon(x_{N-1}) \in U_{N-1}(x_{N-1})$. The RHS inequalities there hold just by the construction in (2.3). By taking $\epsilon \to 0$ in equations (2.4) and (2.5), it is also seen that $J_{N-1}(x_{N-1}) = J_{N-1}^*(x_{N-1})$.

- Suppose that equations (2.4)-(2.6) hold for period $k+1$. For period k, we have:

  $$\begin{aligned}
  &J_k^\epsilon(x_k)\\
  =&\mathbb{E}_{w_k}\left[g_k(x_k, \mu_k^\epsilon(x_k), w_k) + J_{k+1}^\epsilon(f_k(x_k, \mu_k^\epsilon(x_k), w_k))\right] \quad \text{(by definition of } J_k^\epsilon(x_k))\\
  \leq&\mathbb{E}_{w_k}\left[g_k(x_k, \mu_k^\epsilon(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k^\epsilon(x_k), w_k))\right] + (N-k-1)\epsilon \quad \text{(by IH)}\\
  \leq&J_k(x_k) + \epsilon + (N-k-1)\epsilon \quad \text{(by equation (2.3))}\\
  =&J_k(x_k) + (N-k)\epsilon.
  \end{aligned}$$

We also have

$$\begin{aligned}
&J_k^\epsilon(x_k)\\
=&\mathbb{E}_{w_k}\left[g_k\left(x_k,\mu_k^\epsilon(x_k),w_k\right)+J_{k+1}^\epsilon\left(f_k\left(x_k,\mu_k^\epsilon(x_k),w_k\right)\right)\right] \quad \text{(by definition of } J_k^\epsilon(x_k))\\
\geq&\mathbb{E}_{w_k}\left[g_k\left(x_k,\mu_k^\epsilon(x_k),w_k\right)+J_{k+1}\left(f_k\left(x_k,\mu_k^\epsilon(x_k),w_k\right)\right)\right] \quad \text{(by IH)}\\
\geq&\min_{u_k\in U_k(x_k)}\mathbb{E}_{w_k}\left[g_k\left(x_k,u_k,w_k\right)+J_{k+1}\left(f_k\left(x_k,u_k,w_k\right)\right)\right] \quad \text{(by taking min)}\\
=&J_k(x_k).
\end{aligned}$$

Combining the preceding two relations, we see that equation (2.4) holds.

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

$$\begin{aligned}
&J_k^\epsilon(x_k)\\
=&\mathbb{E}_{w_k}\left[g_k\left(x_k,\mu_k^\epsilon(x_k),w_k\right)+J_{k+1}^\epsilon\left(f_k\left(x_k,\mu_k^\epsilon(x_k),w_k\right)\right)\right] \quad \text{(by definition of } J_k^\epsilon(x_k))\\
\leq&\mathbb{E}_{w_k}\left[g_k\left(x_k,\mu_k^\epsilon(x_k),w_k\right)+J_{k+1}\left(f_k\left(x_k,\mu_k^\epsilon(x_k),w_k\right)\right)\right]+(N-k-1)\epsilon \quad \text{(by IH)}\\
\leq&\min_{u_k\in U_k(x_k)}\mathbb{E}_{w_k}\left[g_k\left(x_k,u_k,w_k\right)+J_{k+1}\left(f_k\left(x_k,u_k,w_k\right)\right)\right]+(N-k)\epsilon \quad \text{(by (2.3))}\\
\leq&\mathbb{E}_{w_k}\left[g_k\left(x_k,\mu_k(x_k),w_k\right)+J_{k+1}\left(f_k\left(x_k,\mu_k(x_k),w_k\right)\right)\right]+(N-k)\epsilon\\
&\text{(since } \mu_k(x_k) \text{ is an admissible control for period } k)\\
&\text{(Note that by IH, } J_{k+1} \text{ is the optimal cost starting from period } k+1)\\
\leq&\mathbb{E}_{w_k}\left[g_k\left(x_k,\mu_k(x_k),w_k\right)+J_{\pi^{k+1}}\left(f_k\left(x_k,\mu_k(x_k),w_k\right)\right)\right]+(N-k)\epsilon\\
&\text{(where } \pi^{k+1} \text{ is an admissible policy starting from period } k+1)\\
=&J_{\pi^k}(x_k)+(N-k)\epsilon, \quad \text{(for } \pi^k=(\mu_k,\pi^{k+1}))
\end{aligned}$$

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

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

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

$$J_k^*(x_k)\leq J_k^\epsilon(x_k)$$

Combining the preceding two relations, we see that equation (2.5) holds for period k. Finally, equation (2.6) follows from equations (2.4) and (2.5) by taking $\epsilon\to 0$, and the induction is complete.

$\square$

Example 2.6 (Linear-quadratic example)**.** A certain material is passed through a sequence of two ovens (see Figure 2.16). Let

- x_0 : Initial temperature of the material,

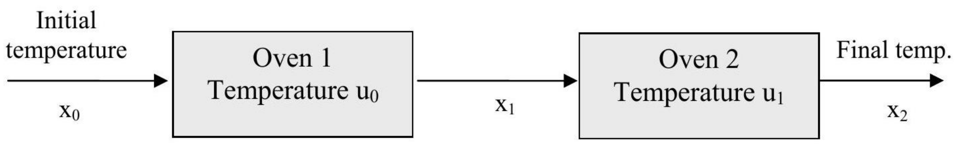


Figure 2.16: System dynamics of Example 2.6.

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

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

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

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 $g_2\left(x_2\right) = r\left(x_2 - T\right)^2$, so that the border condition for the algorithm is

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

Proceeding backwards, we have

$$\begin{aligned} J_1\left(x_1\right) &= \min_{u_1 \geq 0} \left\{u_1^2 + J_2\left(x_2\right)\right\} \\ &= \min_{u_1 \geq 0} \left\{u_1^2 + J_2\left((1-a)x_1 + au_1\right)\right\} \\ &= \min_{u_1 \geq 0} \left\{u_1^2 + r\left((1-a)x_1 + au_1 - T\right)^2\right\}. \end{aligned} \tag{2.7}$$

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 $u_1 = \mu_1^*\left(x_1\right)$ depending linearly on x_1. By substituting back this expression for u_1 into (2.7), we obtain a closed form expression for $J_1\left(x_1\right)$, which is quadratic in x_1.

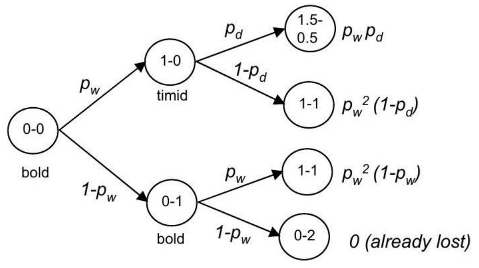


Figure 2.17: Optimal closed-loop policy for the two-game chess match. The payoffs included next to the leaves represent the total cumulative payoff for following that particular branch from the root.

Proceeding backwards further,

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

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.

2.6 Extensions

2.6.1 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 (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 2.17 illustrates this. The optimal payoff under the closed-loop policy is the sum of the payoffs in the leaves of Figure 2.17. This is because the four payoffs correspond to four mutually exclusive outcomes. The total payoff is

$$\begin{aligned}\mathbb{P}(\text{win}) &= p_w p_d + p_w^2\left((1-p_d) + (1-p_w)\right) \\ &= p_w^2(2-p_w) + p_w(1-p_w)p_d.\end{aligned} \tag{2.8}$$

  For example, if $p_w = 0.45$ and $p_d = 0.9$, we know that $\mathbb{P}(\text{win}) = 0.53$.

- 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

$$\underbrace{p_w p_d}_{\geq p_d^2 p_w} + \underbrace{p_w^2 (1 - p_d)}_{\geq 0} \geq p_d^2 p_w$$

Therefore, the maximum open-loop probability of winning the match is:

$$\max\{ \underbrace{p_w^2 (3 - 2p_w)}_{\text{Play (bold, bold)}}, \quad \underbrace{p_w p_d + p_w^2 (1 - p_d)}_{\text{Play (bold, timid) or (timid, bold)}} \quad \} = p_w^2 + p_w (1 - p_w) \max\{2p_w, p_d\}. \tag{2.9}$$

So,

- if $p_d > 2p_w$, then the optimal policy is to play either (timid,bold) or (bold, timid);
- if $p_d \leq 2p_w$, then the optimal policy is to play (bold, bold).

Again, if $p_w = 0.45$ and $p_d = 0.9$, then $\mathbb{P}(\text{win}) = 0.425$

- 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 (2.8)-(2.9):

$$\begin{aligned}\text{Value of Information} &= p_w^2 (2 - p_w) + p_w (1 - p_w) p_d - p_w^2 - p_w (1 - p_w) \max\{2p_w, p_d\} \\ &= p_w (1 - p_w) \min\{p_w, p_d - p_w\}.\end{aligned}$$

2.6.2 State Augmentation

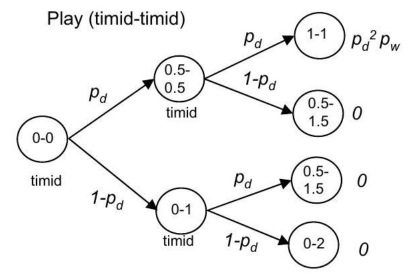
In the basic DP formulation, the random noise is independent across all periods, and the control depends just on the current state. In this regard, the system is of the Markovian type. In order to deal with a more general situation, we enlarge the state definition so that the current state captures information of the past. In some applications, this past information could be helpful for the future.

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,

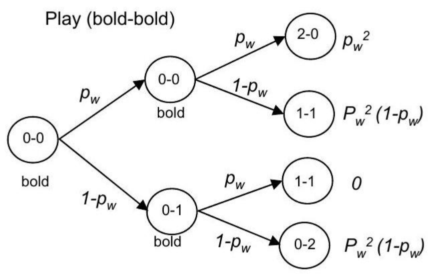
$$\begin{aligned} x_{k+1} &= f_k (x_k, x_{k-1}, u_k, u_{k-1}, w_k), \quad k = 1, \ldots, N-1 \\ x_1 &= f_0 (x_0, u_0, w_0). \end{aligned}$$

We redefine the system equation as follows:

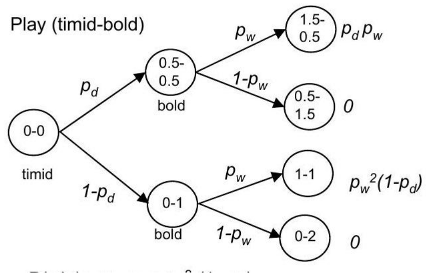
$$\underbrace{\begin{pmatrix} x_{k+1} \\ y_{k+1} \\ s_{k+1} \end{pmatrix}}_{\widetilde{x}_{k+1}} = \underbrace{\begin{pmatrix} f_k (x_k, y_k, u_k, s_k, w_k) \\ x_k \\ u_k \end{pmatrix}}_{\widetilde{f}_k(\widetilde{x}_k, u_k, w_k)},$$



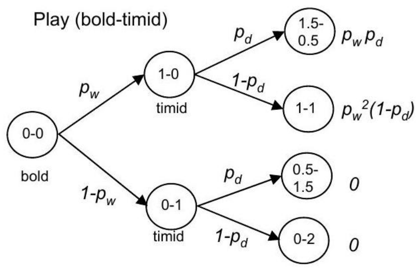
(a) $\mathrm{P(win)} = p_d^2 p_w$



(b) $\mathrm{P(win)} = {p_w}^2 + 2p_w^2(1-p_w) = p_w^2(3-2p_w)$



(c) $\mathrm{P(win)} = p_w p_d + {p_w}^2(1-p_d)$



(d) $\mathrm{P(win)} = p_w p_d + p_w^2(1-p_d)$

Figure 2.18: 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.

where $\widetilde{x}_k = (x_k, y_k, s_k) = (x_k, x_{k-1}, u_{k-1})$.

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:

$$J_N\left(x_N\right) = g_N\left(x_N\right),$$

and

$$\begin{aligned} & J_{N-1}\left(x_{N-1}, x_{N-2}, u_{N-2}\right) \\ = & \min_{u_{N-1} \in U_{N-1}(x_{N-1})} \mathbb{E}_{w_{N-1}} \left\{ g_{N-1}\left(x_{N-1}, u_{N-1}, w_{N-1}\right) + J_N\big(\underbrace{f_{N-1}\left(x_{N-1}, x_{N-2}, u_{N-1}, u_{N-2}, w_{N-1}\right)}_{x_N}\big) \right\}, \end{aligned}$$

and for $k = 1, \dots, N-2$,

$$J_k\left(x_k, x_{k-1} \dots, u_{k-1}\right) = \min_{u_k \in U_k(x_k)} \mathbb{E}_{w_k} \left\{ g_k\left(x_k, u_k, w_k\right) + J_{k+1}\left(f_k\left(x_k, x_{k-1}, u_k, u_{k-1}, w_k\right), x_k, u_k\right) \right\},$$

finally,

$$J_0\left(x_0\right) = \min_{u_0 \in U_0(x_0)} \mathrm{E}_{w_0} \left\{ g_0\left(x_0, u_0, w_0\right) + J_1\left(f_0\left(x_0, u_0, w_0\right), x_0, u_0\right) \right\}.$$

Correlated Disturbances. Assume that $w_0, w_1, \dots, w_{N-1}$ can be represented as the output of a linear system driven by independent r.v. For example, suppose that disturbances can be modeled as:

$$w_k = C_k y_{k+1}, \text{ where } y_{k+1} = A_k y_k + \xi_k, \quad k = 0, 1, \dots, N-1,$$

where C_k, A_k are matrices of appropriate dimension, and ξ_k are independent random vectors. By viewing y_k as an additional state variable; we obtain the new system equation:

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} f_k\left(x_k, u_k, C_k\left(A_k y_k + \xi_k\right)\right) \\ A_k y_k + \xi_k \end{pmatrix}$$

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:

$$\begin{tikzcd} {} \arrow[r, "\xi_k"] & \boxed{y_{k+1} = A_k y_k + \xi_k} \arrow[r, "y_{k+1}"] & \boxed{C_k} \arrow[r, "w_k"] & {} \end{tikzcd}$$

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.

$$J_N(x_N, y_N) = g_N(x_N),$$

$$J_k(x_k, y_k) = \min_{u_k \in U_k(x_k)} \mathbb{E}_{\xi_k} \left\{ g_k(x_k, u_k, C_k(A_k y_k + \xi_k)) + J_{k+1}(\underbrace{f_k(x_k, u_k, C_k(A_k y_k + \xi_k)}_{x_{k+1}}, \underbrace{A_k y_k + \xi_k}_{y_{k+1}}) \right\}.$$

2.6.3 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 priory 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 ξ_k is the r.v. taking value i w.p. p_i. So, when ξ_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: $\widetilde{w}_k = (w_k, \xi_k)$, where w_k stands for the outcome of the previous forecast in the current period, and ξ_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.

3 Infinite Horizon Problems

3.1 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 π,

  $$J_\pi(x_0) = \lim_{N\to\infty} \mathbb{E}_{w_0,w_1,\ldots}\left[\sum_{k=0}^{N-1} \alpha^k g\left(x_k, \mu_k\left(x_k\right), w_k\right)\right]$$

  The value function $J_\pi(x_0)$ should be finite for at least some admissible policies π 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, $J_\pi(x_0) <$ 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 ∞. Technically more challenging!

- Average cost problems (type (d)): Minimize over all admissible policies π,

  $$J_\pi(x_0) = \lim_{N\to\infty} \frac{1}{N}\mathbb{E}_{w_0,w_1,\ldots}\left[\sum_{k=0}^{N-1} g\left(x_k, \mu_k\left(x_k\right), w_k\right)\right]$$

  The approach works even if $J_\pi(x_0)$ is infinite for every policy π and initial state x_0.

3.2 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

  $$J_{k+1} = \min_{u\in U(x)} \mathbb{E}_w\left[g(x, u, w) + J_k(f(x, u, w))\right]. \quad \forall x$$

  Typical results for total cost problems:

- Relation valuable from a computational viewpoint:

$$J^*(x) = \lim_{N\to\infty} J_N(x), \quad \forall x$$

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,

$$J^*(x) = \min_{u\in U(x)} \mathbb{E}_w \left[g(x,u,w) + J^*(f(x,u,w))\right], \forall x. \quad \text{(Bellman's equation)}$$

- 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)).

3.3 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

$$J_\pi(i) = \lim_{N\to\infty} \mathbb{E}\left[\sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k)) \mid x_0 = i\right]$$

where α is a discount factor, with $0 < \alpha \leq 1$.

- Optimal cost from state i is $J^*(i) = \min_\pi J_\pi(i)$.

- 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 μ is optimal if

$$J_\mu(i) = J^*(i) = \min_\pi J_\pi(i), \forall i.$$

3.4 Stochastic shortest path problems

- Assume there is no discounting (i.e., $\alpha = 1$).
- Set of "normal states" $\{1, 2, \dots, n\}$.
- 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 3.1. 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 π, we have

  $$\rho_\pi = \mathbb{P}\left\{x_m \neq t \mid x_0 \neq t, \pi\right\} < 1$$

  That is, $\mathbb{P}\left\{x_m = t \mid x_0 \neq t, \pi\right\} > 0$.

- In terms of discrete-time Markov chains, Assumption 3.1 is claiming that t is accessible from any state i.
- Remark: Assumption 3.1 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 π, the corresponding cost $J_\pi(i)$ is ∞ for at least one state i.

- Note that the assumption implies that

  $$\mathbb{P}\left\{x_m \neq t \mid x_0 = i, \pi\right\} \leq \mathbb{P}\left\{x_m \neq t \mid x_0 \neq t, \pi\right\} = \rho_\pi < 1, \quad \forall i = 1, \dots, n$$

- 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 ρ_π, so that the max above is well defined (we do not need sup). Then,

  $$\mathbb{P}\left\{x_m \neq t \mid x_0 \neq t, \pi\right\} \leq \rho < 1$$

- For any π and any initial state i,

$$\begin{aligned}\mathbb{P}\{x_{2m} \neq t \mid x_0 = i, \pi\} &= \mathbb{P}\{x_{2m} \neq t \mid x_m \neq t, x_0 = i, \pi\} \times \mathbb{P}\{x_m \neq t \mid x_0 = i, \pi\} \\ &\leq \mathbb{P}\{x_{2m} \neq t \mid x_m \neq t, \pi\} \times \mathbb{P}\{x_m \neq t \mid x_0 \neq t, \pi\} \\ &\leq \rho^2\end{aligned}$$

and similarly,

$$\mathbb{P}\{x_{km} \neq t \mid x_0 = i, \pi\} \leq \rho^k, \quad i = 1, \ldots, n$$

- So,

$$|\mathbb{E}[\text{cost between times } km \text{ and } (k+1)m - 1]| \leq m \times \rho^k \times \max_{\substack{i=1,\ldots,n \\ u \in U(i)}} |g(u,i)| \tag{3.1}$$

and hence,

$$|J_\pi(i)| \leq \sum_{k=0}^{\infty} m\rho^k \max_{\substack{i=1,\ldots,n \\ u \in U(i)}} |g(u,i)| = \frac{m}{1-\rho} \max_{\substack{i=1,\ldots,n \\ u \in U(i)}} |g(u,i)|$$

- Key idea for the main result (to be presented below) is that the tail of the cost series vanishes, i.e.,

$$\lim_{K \to \infty} \sum_{k=mK}^{\infty} \mathbb{E}[g(x_k, \mu_k(x_k))] = 0$$

The reason is that $\lim_{K\to\infty} \mathbb{P}\{x_{mK} \neq t \mid x_0 = i, \pi\} = 0$.

Proposition 3.2. Under Assumption 3.1, 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

$$J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J_k(j) \right\}, \quad \forall i \tag{3.2}$$

converges to the optimal cost $J^*(i)$.

(b) The optimal costs $J^*(1), \ldots, J^*(n)$ satisfy Bellman's equation,

$$J^*(i) = \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J^*(j) \right\}, \quad i = 1, \ldots, n \tag{3.3}$$

and in fact they are the unique solution of this equation.

(c) For any stationary policy μ, 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 μ is optimal if and only if for every state i, $\mu(i)$ attains the minimum in Bellman's equation (3.3).

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:

$$\begin{aligned} J_\pi(x_0) &= \lim_{N\to\infty} \mathbb{E}\left[\sum_{k=0}^{N-1} g(x_k, \mu_k(x_k))\right] \\ &= \mathbb{E}\left[\sum_{k=0}^{mK-1} g(x_k, \mu_k(x_k))\right] + \lim_{N\to\infty} \mathbb{E}\left[\sum_{k=mK}^{N-1} g(x_k, \mu_k(x_k))\right] \end{aligned} \tag{3.4}$$

Let M be an upper bound on the cost of an m-stage cycle, assuming t is not reached during the cycle, i.e.,

$$M = m \max_{\substack{i=1,\ldots,n \\ u\in U(i)}} |g(i,u)|$$

Recall from (3.1) that

$$|\mathbb{E}[\text{cost during } K \text{ th cycle, between stages } Km \text{ and } (K+1)m-1]| \le M\rho^K, \tag{3.5}$$

so that

$$\left|\lim_{N\to\infty} \mathbb{E}\left[\sum_{k=mK}^{N-1} g(x_k, \mu_k(x_k))\right]\right| \le M \sum_{k=K}^{\infty} \rho^k = \frac{\rho^K M}{1-\rho} \tag{3.6}$$

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 π 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,

$$\begin{aligned}
\left|\mathbb{E}\left[J_0(x_{mK})\right]\right| &= \left|\sum_{i=1}^{n} \mathbb{P}\left\{x_{mK} = i \mid x_0 \neq t, \pi\right\} J_0(i) + \mathbb{P}\left\{x_{mK} = t \mid x_0 \neq t, \pi\right\} \underbrace{J_0(t)}_{0}\right| \\
&= \left|\sum_{i=1}^{n} \mathbb{P}\left\{x_{mK} = i \mid x_0 \neq t, \pi\right\} J_0(i)\right| \\
&\leq \underbrace{\left(\sum_{i=1}^{n} \mathbb{P}\left\{x_{mK} = i \mid x_0 \neq t, \pi\right\}\right)}_{\mathbb{P}\{x_{mK} \neq t \mid x_0 \neq t, \pi\}} \max_{i=1,\ldots,n} |J_0(i)| \\
&\leq \rho^K \max_{i=1,\ldots,n} |J_0(i)|
\end{aligned} \tag{3.7}$$

Now, we set the following bound (following equations (3.6) and (3.7)):

$$\begin{aligned}
\left|\mathbb{E}\left[J_0(x_{mK})\right] - \lim_{N\to\infty} \mathbb{E}\left[\sum_{k=mK}^{N-1} g\left(x_k, \mu_k(x_k)\right)\right]\right| &\leq \left|\mathbb{E}\left[J_0(x_{mK})\right]\right| + \left|\lim_{N\to\infty} \mathbb{E}\left[\sum_{k=mK}^{N-1} g\left(x_k, \mu_k(x_k)\right)\right]\right| \\
&\leq \rho^K \max_{i=1,\ldots,n} |J_0(i)| + M \sum_{k=K}^{\infty} \rho^k \\
&= \rho^K \max_{i=1,\ldots,n} |J_0(i)| + \frac{\rho^K M}{1-\rho}.
\end{aligned}$$

Taking the LHS above, and using (3.4), we have

$$\begin{aligned}
&\left|\mathbb{E}\left[J_0(x_{mK})\right] - \lim_{N\to\infty} \mathbb{E}\left[\sum_{k=mK}^{N-1} g\left(x_k, \mu_k(x_k)\right)\right]\right| \\
&= \left|\mathbb{E}\left[J_0(x_{mK})\right] + \mathbb{E}\left[\sum_{k=0}^{mK-1} g\left(x_k, \mu_k(x_k)\right)\right] - J_\pi(x_0)\right|
\end{aligned}$$

Then, we get the bounds

$$\begin{aligned}
-\rho^K \max_{i=1,\ldots,n} |J_0(i)| + J_\pi(x_0) - \frac{\rho^K M}{1-\rho} \leq &\mathbb{E}\left[J_0(x_{mK})\right] + \mathbb{E}\left[\sum_{k=0}^{mK-1} g\left(x_k, \mu_k(x_k)\right)\right] \\
\leq &\rho^K \max_{i=1,\ldots,n} |J_0(i)| + J_\pi(x_0) + \frac{\rho^K M}{1-\rho}
\end{aligned} \tag{3.8}$$

Note that

- The expected value of the middle term above is the mK-stage cost of policy π, starting from state x_0, with terminal cost $J_0\left(x_{mK}\right)$.
- The min of this mK-stage cost over all π is equal to the value $J_{mK}\left(x_0\right)$, which is generated by the DP recursion (3.2) after mK iterations.

Thus, taking the min over π in equation (3.8), we obtain for all x_0 and K,

$$-\rho^K \max_{i=1,\ldots,n} |J_0(i)| + J^*\left(x_0\right) - \frac{\rho^K M}{1-\rho} \leq J_{mK}\left(x_0\right) \leq \rho^K \max_{i=1,\ldots,n} |J_0(i)| + J^*\left(x_0\right) + \frac{\rho^K M}{1-\rho} \tag{3.9}$$

When taking limit as $K \to \infty$, the terms in LHS and RHS involving $\rho^K \to 0$, leading to

$$\lim_{K\to\infty} J_{mK}\left(x_0\right) = J^*\left(x_0\right), \quad \forall x_0$$

Since from (3.5)

$$\left|J_{mK+q}\left(x_0\right) - J_{mK}\left(x_0\right)\right| \leq \rho^K M, \quad q = 0, \ldots, m-1,$$

we see that for $q = 0, \ldots, m-1$,

$$-\rho^K M + J_{mK}\left(x_0\right) \leq J_{mK+q}\left(x_0\right) \leq J_{mK}\left(x_0\right) + \rho^K M$$

Taking limit as $K \to \infty$, we get

$$\lim_{K\to\infty} \left(\rho^K M + J_{mK}\left(x_0\right)\right) = J^*\left(x_0\right)$$

Thus, for any $q = 0, \ldots, m-1$,

$$\lim_{K\to\infty} J_{mK+q}\left(x_0\right) = J^*\left(x_0\right)$$

and hence,

$$\lim_{k\to\infty} J_k\left(x_0\right) = J^*\left(x_0\right)$$

(b) Existence: By taking limit as $k \to \infty$ in the DP iteration (3.2), 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 (3.2) starting from $J(1), \ldots, J(n)$ just replicates $J(1), \ldots, J(n)$. Then, from the convergence result of part (a), $J(i) = J^*(i), \forall i = 1, \ldots, n$.

(c) Given stationary policy μ, 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.,

$$J_\mu(i) = g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i)) J_\mu(j), \quad \forall i = 1, \ldots, n$$

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 (3.3) if and only if we have

$$\begin{aligned} J^*(i) &= \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J^*(j) \right\} \\ &= g(i,\mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i)) J^*(j), \quad i = 1, \ldots, n \end{aligned}$$

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.

□

Observation: Part (c) provides a a way to compute $J_\mu(i), i = 1, \ldots, n$, for a given stationary policy μ, but the computation is substantial for large n (of order $O\left(n^3\right)$).

Example 3.1 (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
$$J^*(i) = \min_{u \in U(i)} \left\{ 1 + \sum_{j=1}^{n} p_{ij}(u) J^*(j) \right\}, \quad \forall i = 1, \ldots, n$$
- 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
$$m_i = 1 + \sum_{j=1}^{n} p_{ij} m_j, \quad \forall i = 1, \ldots, n$$
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.

Example 3.2 (Spider and a fly)**.**

- A spider and a fly move along a straight line.
- At the beginning of each period, the spider knows the position of the fly.

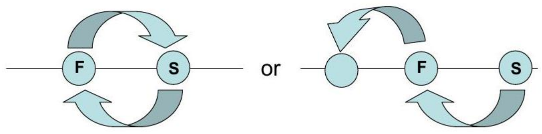


Figure 3.1: Transition probabilities for control M from state 1.

- 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 3.1. 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:

$$\begin{aligned} J^*(i) &= 1 + pJ^*(i) + (1-2p)J^*(i-1) + pJ^*(i-2), \quad i \geq 2. \\ J^*(1) &= 1 + \min\{\underbrace{2pJ^*(1)}_{M}, \underbrace{pJ^*(2) + (1-2p)J^*(1)}_{\bar{M}}\} \end{aligned}$$

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:

$$J^*(1) = 1 + \min\left\{2pJ^*(1), \frac{p}{1-p} + \frac{(1-2p)J^*(1)}{1-p}\right\}$$

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 $i \geq 3$.

3.5 Computational approaches

There are three main computational approaches used in practice for calculating the optimal cost function J^*.

3.5.1 Value iteration

The DP iteration

$$J_{k+1}(i) = \min_{i \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J_k(j) \right\}, \quad i = 1, \ldots, n$$

is called value iteration.

From equation (3.9), we know that the error

$$|J_{mK}(i) - J^*(i)| \leq D\rho^K, \text{ for some constant } D$$

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

$$J_{k+1}(j) + (N^*(j) - 1)\,\underline{\mathbf{c}}_k \leq J^*(j) \leq J_{\mu^k}(j) \leq J_{k+1}(j) + \left(N^k(j) - 1\right)\bar{c}_k$$

where

- μ^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 μ^k,
- $\underline{c}_k = \min_{i=1,\ldots,n} \{J_{k+1}(i) - J_k(i)\}$,
- $\bar{c}_k = \max_{i=1,\ldots,n} \{J_{k+1}(i) - J_k(i)\}$,

Unfortunately, the values $N^*(j)$ and $N^k(j)$ are easily computed or approximated only in some cases.

3.5.2 Policy iteration

- It generates a sequence $\mu^1, \mu^2, \ldots$ of stationary policies, starting with any stationary policy μ^0.
- At a typical iteration, given a policy μ^k, we perform two steps:
  (i) Policy evaluation step: Computes $J_{\mu^k}(i)$ as the solution of the linear system of equations

$$J(i) = g\left(i, \mu^k(i)\right) + \sum_{j=1}^{n} p_{ij}\left(\mu^k(i)\right) J(j), \quad i = 1, \ldots, n \tag{3.10}$$

  in the unknowns $J(1), \ldots, J(n)$.

  (ii) Policy improvement step: Computes a new policy μ^{k+1} as

$$\mu^{k+1}(i) = \arg\min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J_{\mu^k}(j) \right\}, \quad i = 1, \ldots, n \tag{3.11}$$

- The algorithm stops when $J_{\mu^k}(i) = J_{\mu^{k+1}}(i)$ for all i.

Proposition 3.3. Under Assumption 3.1, the policy iteration algorithm for the stochastic shortest path problem generates an improving sequence of policies (i.e., $J_{\mu^{k+1}}(i) \le J_{\mu^k}(i), \forall i, k$) and terminates with an optimal policy.

Proof. For any k, consider the sequence generated by the recursion

$$J_{N+1}(i) = g\left(i, \mu^{k+1}(i)\right) + \sum_{j=1}^{n} p_{ij}\left(\mu^{k+1}(i)\right) J_N(j), \quad i = 1, \ldots, n \tag{3.12}$$

where $N = 0, 1, \ldots$, and the solution to equation (3.10):

$$J_0(i) = J_{\mu^k}(i), \quad i = 1, \ldots, n$$

From equation (3.10), we have

$$\begin{aligned} J_0(i) &= g\left(i, \mu^k(i)\right) + \sum_{j=1}^{n} p_{ij}\left(\mu^k(i)\right) J_0(j) \\ &\geq g\left(i, \mu^{k+1}(i)\right) + \sum_{j=1}^{n} p_{ij}\left(\mu^{k+1}(i)\right) J_0(j) \quad \text{(from (3.11))} \\ &= J_1(i), \forall i \quad \text{(from iteration (3.12))} \end{aligned}$$

By using the above inequality we obtain

$$\begin{aligned} J_1(i) &= g(i, \mu^{k+1}(i)) + \sum_{j=1}^{n} p_{ij}(\mu^{k+1}(i)) J_0(j) \\ &\geq g(i, \mu^{k+1}(i)) + \sum_{j=1}^{n} p_{ij}(\mu^{k+1}(i)) J_1(j) \quad \text{(because } J_0(i) \geq J_1(i)) \\ &= J_2(i), \forall i \quad \text{(from iteration (3.12))}. \end{aligned}$$

Continuing similarly we get

$$J_0(i) \geq J_1(i) \geq \cdots \geq J_N(i) \geq J_{N+1}(i) \geq \cdots, \quad i = 1, \ldots, n$$

Since by Proposition 3.2(c), $J_N(i) \to J_{\mu^{k+1}}(i)$, we obtain

$$J_0(i) \geq J_{\mu^{k+1}}(i) \Rightarrow J_{\mu^k}(i) \geq J_{\mu^{k+1}}(i), \quad i = 1, \ldots, n, \quad k = 0, 1, \ldots \tag{3.13}$$

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 (3.13), which in particular means from (3.12),

$$J_0(i) = J_{\mu^k}(i) = J_1(i) = g\left(i, \mu^{k+1}(i)\right) + \sum_{j=1}^{n} p_{ij}\left(\mu^{k+1}(i)\right) J_{\mu^k}(j)$$

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 3.2(b), it follows that $J_{\mu^k}(i) = J^*(i)$, and that $\mu^k(i)$ is optimal. □

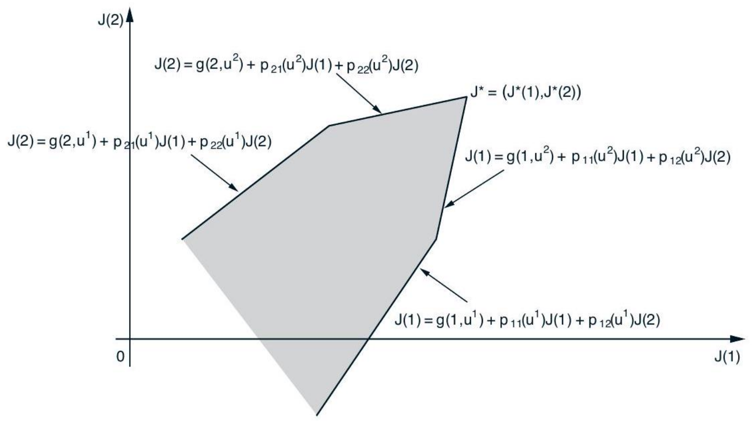

J(2)
J(2) = g(2,u^2) + p_21(u^2)J(1) + p_22(u^2)J(2)
J* = (J*(1),J*(2))
J(2) = g(2,u^1) + p_21(u^1)J(1) + p_22(u^1)J(2)
J(1) = g(1,u^2) + p_11(u^2)J(1) + p_12(u^2)J(2)
J(1) = g(1,u^1) + p_11(u^1)J(1) + p_12(u^1)J(2)
0
J(1)

Figure 3.2: Illustration of the LP solution method for infinite horizon DP.

3.5.3 Linear programming

Claim: J^* is the "largest" J that satisfies the constraints

$$J(i) \leq g(i,u) + \sum_{j=1}^{n} p_{ij}(u)J(j) \tag{3.14}$$

for all $i = 1, \ldots, n$, and $u \in U(i)$.

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 3.2(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 (3.14). □

Figure 3.2 illustrates a linear program associated with a two-state stochastic shortest path problem. The decision variables in this case are $J(1)$ and $J(2)$.

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.

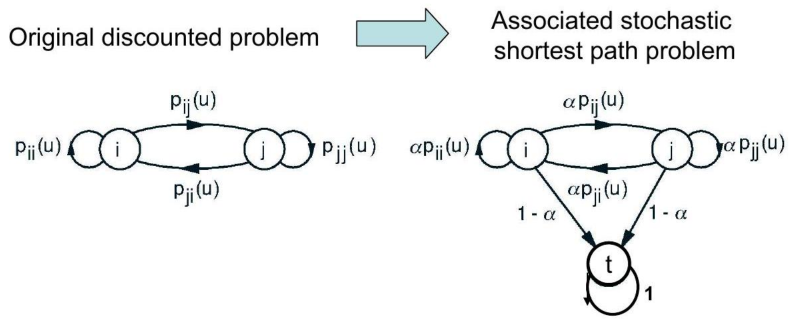

Original discounted problem
Associated stochastic shortest path problem
pij(u)
pii(u)
pjj(u)
pji(u)
αpij(u)
αpii(u)
αpjj(u)
αpji(u)
1 - α
1 - α
1

Figure 3.3: Illustration of the transformation from α-discounted to stochastic shortest path.

3.6 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 α. The instantaneous costs $g(i, u)$ are preserved. Figure 5.3.1 illustrates this transformation.
- Justification: Take a policy μ, 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 $g\left(x_k, \mu_k\left(x_k\right)\right)$, multiplied by the probability that state t has not been reached, which is α^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:
    Discounted: $J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \alpha \sum_{j=1}^n p_{ij}(u) J_k(j) \right\}, i = 1, \ldots, n.$
    Corresponding SSP: $J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^n \left(\alpha p_{ij}(u)\right) J_k(j) \right\}, i = 1, \ldots, n.$
- The results of SPP, summarized in Proposition 3.2, 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 3.4. The following hold for the discounted problem:

(a) Given any initial conditions $J_0(1), \ldots, J_0(n)$, the value iteration algorithm

$$J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i,u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J_k(j) \right\}, \quad \forall i \tag{3.15}$$

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 μ, 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 μ 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 (3.9)), we can show that

- $|J_k(i) - J^*(i)| \le D\alpha^k$, for some constant D.
- The error bounds become

$$J_{k+1}(j) + \frac{\alpha}{1-\alpha}\underline{c}_k \le J^*(j) \le J_{\mu^k}(j) \le J_{k+1}(j) + \frac{\alpha}{1-\alpha}\bar{c}_k,$$

where $\mu^k(j)$ attains the minimum in the k th value iteration (3.15) for all i, and

$$\underline{c}_k = \min_{i=1,\ldots,n} \left[J_{k+1}(i) - J_k(i)\right], \text{ and } \bar{c}_k = \max_{i=1,\ldots,n} \left[J_{k+1}(i) - J_k(i)\right]$$

Example 3.3 (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

$$J^*(x) = \max\left\{x, \frac{\mathbb{E}\left[J^*(w)\right]}{1+r}\right\}$$

- An optimal policy is to sell of and only if the current offer x_k is greater than or equal to $\bar{\alpha}$, where

$$\bar{\alpha} = \frac{\mathbb{E}\left[J^*(w)\right]}{1+r}$$

Example 3.4 (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

$$J^*(i) = \min\{\underbrace{K + \alpha(1-p)J^*(0) + \alpha p J^*(1)}_{\text{Process remaining orders}}, \underbrace{ci + \alpha(1-p)J^*(i) + \alpha p J^*(i+1)}_{\text{Do nothing}}\}$$

  for the states $i = 0, 1, \ldots, n-1$, and takes the form

$$J^*(n) = K + \alpha(1-p)J^*(0) + \alpha p J^*(1)$$

  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,

$$K + \alpha(1-p)J^*(0) + \alpha p J^*(1) \le cm + \alpha(1-p)J^*(m) + \alpha p J^*(m+1)$$

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 $J_k(i+1) \ge J_k(i)$ for all i. We will see that $J_{k+1}(i+1) \ge J_{k+1}(i)$ for all i. Consider first the case $i+1 < n$. Then, by induction hypothesis, we have

$$c(i+1) + \alpha(1-p)J_k(i+1) + \alpha p J_k(i+2) \ge ci + \alpha(1-p)J_k(i) + \alpha p J_k(i+1) \tag{3.16}$$

Define for any scalar γ,

$$F_k(\gamma) = \min\{K + \alpha(1-p)J_k(0) + \alpha p J_k(1), \gamma\}$$

Since $F_k(\gamma)$ is monotonically increasing in γ, we have from equation (3.16),

$$\begin{aligned} J_{k+1}(i+1) &= F_k\left(c(i+1) + \alpha(1-p)J_k(i+1) + \alpha p J_k(i+2)\right) \\ &\geq F_k\left(ci + \alpha(1-p)J_k(i) + \alpha p J_k(i+1)\right) \\ &= J_{k+1}(i) \end{aligned}$$

Finally, consider the case $i+1=n$. Then, we have

$$\begin{aligned} J_{k+1}(n) &= K + \alpha(1-p)J_k(0) + \alpha p J_k(1) \\ &\geq F_k\left(ci + \alpha(1-p)J_k(i) + \alpha p J_k(i+1)\right) \\ &= J_{k+1}(n-1) \end{aligned}$$

The induction is complete. □

3.7 Average cost-per-stage problems

3.7.1 General setting

- Stationary system with finite number of states and controls
- Minimize over admissible policies $\pi = \{\mu_0, \mu_1, \ldots\}$,

$$J_\pi(i) = \lim_{N\to\infty} \frac{1}{N}\mathbb{E}\left[\sum_{k=0}^{N-1} g\left(x_k, \mu_k\left(x_k\right)\right) \mid x_0 = i\right]$$

- Assume $0 \leq g\left(x_k, \mu_k\left(x_k\right)\right) < \infty$.
- 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 μ.[6] Let

$$K_{ij}(\mu) = \text{ first passage time from } i \text{ to } j \text{ under } \mu$$

[6]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.

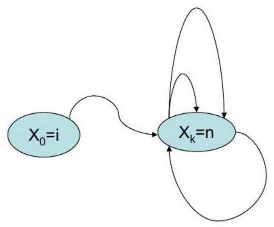


Figure 3.4: Each cycle can be viewed as a state trajectory of a corresponding SSP problem with termination state being n.

i.e., $K_{ij}(\mu)$ is the first index k such that $x_k = j$ starting from $x_0 = i$. Then,

$$J_\mu(i) = \underbrace{\lim_{N\to\infty} \frac{1}{N}\mathbb{E}\left[\sum_{k=0}^{K_{ij}(\mu)-1} g\left(x_k, \mu_k\left(x_k\right)\right) \mid x_0 = i\right]}_{=0} + \lim_{N\to\infty} \frac{1}{N}\mathbb{E}\left[\sum_{k=K_{ij}(\mu)}^{N-1} g\left(x_k, \mu_k\left(x_k\right)\right) \mid x_0 = i\right]$$

Therefore, $J_\mu(i) = J_\mu(j)$, for all i, j with $\mathbb{E}\left[K_{ij}(\mu)\right] < \infty$ (or equivalently, with $\mathbb{P}\left(K_{ij}(\mu) = \infty\right) = 0$.

- Because communication issues are so important, the methodology relies heavily on Markov chain theory.

3.7.2 Associated stochastic shortest path (SSP) problem

Assumption 3.5. 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 3.4. The SSP is obtained via the transformation described in Figure 3.5.

Let the cost at i of the SSP be $g(i, u) - \lambda^*$. We will show that

$$\text{Average cost problem} \equiv \text{Min cost cycle problem} \equiv \text{SSP problem}$$

3.7.3 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 $n \Rightarrow$ We want to find a stationary policy μ that minimizes the average cycle stage cost.

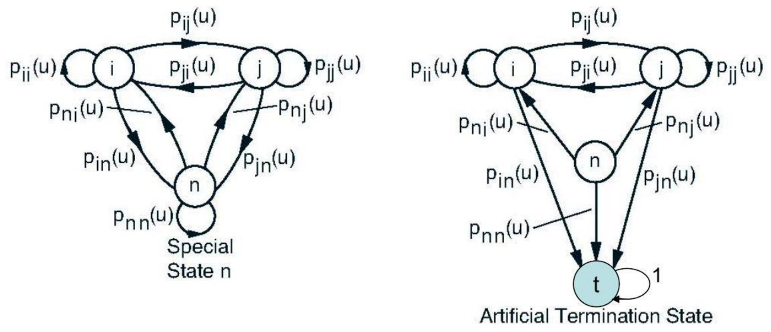


Figure 3.5: LHS: Original average cost per stage problem. RHS: Associated SSP problem. The original transition probabilities are adjusted as follows: probabilities from the states $i \neq t$ to state t are set equal to $p_{in}(u)$, the probabilities of transition from all states to state n are set to zero, and all other probabilities are left unchanged.

- Consider a minimum cycle cost problem: Find a stationary policy μ that minimizes:

$$\begin{gathered}\text{Expected cost per transition within a cycle}\\ =\frac{\mathbb{E}[\text{cost from } n \text{ up to the first return to } n]}{\mathbb{E}[\text{ time from } n \text{ up to the first return to } n]}=\frac{C_{nn}(\mu)}{N_{nn}(\mu)}.\end{gathered}$$

- Intuitively, the optimal average cost λ^* should be equal to optimal average cycle cost, i.e.,

$$\lambda^*=\frac{C_{nn}\left(\mu^*\right)}{N_{nn}\left(\mu^*\right)};\ \text{or equivalently } C_{nn}\left(\mu^*\right)-N_{nn}\left(\mu^*\right)\lambda^*=0$$

  So, for any stationary policy μ,

$$\lambda^* \leq \frac{C_{nn}(\mu)}{N_{nn}(\mu)} \text{ or equivalently } C_{nn}(\mu)-N_{nn}(\mu)\lambda^* \geq 0$$

- Thus, to obtain an optimal policy μ, we must solve

$$\min_{\mu}\left\{C_{nn}(\mu)-N_{nn}(\mu)\lambda^*\right\}$$

  Note that $C_{nn}(\mu)-N_{nn}(\mu)\lambda^*$ is the expected cost of μ starting from n in the associated SSP with stage cost $g(i,u)-\lambda^*$, justified by

$$\mathbb{E}\left[\sum_{k=0}^{K_{nt}(\mu)-1}\left(g\left(x_k,\mu\left(x_k\right)\right)-\lambda^* \mid x_0=n\right)\right]=C_{nn}(\mu)-\underbrace{N_{nn}(\mu)}_{\mathbb{E}[K_{nt}(\mu)]}\lambda^*$$

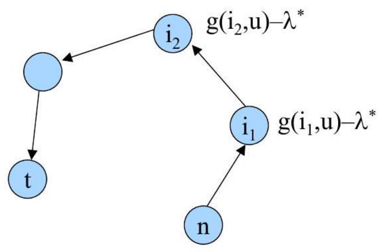


Figure 3.6: $h^*(n)$ in the SSP is the expected cost of the path from n to t (i.e., of the cycle from n to n in the original problem) based on the original $g(i,u)$, minus $N_{nn}(\mu^*)\,\lambda^*$.

- 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 $i = 1, \dots, n$. Then by Proposition 3.2(b) in $h^*(1), \dots, h^*(n)$ solve uniquely the Bellman's equation:

$$\begin{aligned} h^*(i) &= \min_{u \in U(i)} \left\{ g(i,u) - \lambda^* + \sum_{j=1}^{n} p_{ij}(u) h^*(j) \right\} \\ &= \min_{u \in U(i)} \left\{ g(i,u) - \lambda^* + \sum_{j=1}^{n-1} p_{ij}(u) h^*(j) + \underbrace{p_{in}(u)}_{=0,\text{ by construction}} h^*(n) \right\} \end{aligned} \tag{3.17}$$

- If μ^* is a stationary policy that minimizes the cycle cost, then μ^* must satisfy

$$h^*(n) = C_{nn}(\mu^*) - N_{nn}(\mu^*)\,\lambda^* = 0$$

See Figure 3.6.

- We can then rewrite (3.17) as

$$\begin{aligned} h^*(n) &= 0 \\ \lambda^* + h^*(i) &= \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) h^*(j) \right\}, \quad \forall i = 1, \dots, n \end{aligned}$$

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:

$$\begin{aligned} h^*(i) = \min\{&\mathbb{E}[\text{cost to go from } i \text{ to } n \text{ for the first time}] \\ &-\mathbb{E}\left[\text{cost if the stage cost were constant at } \lambda^* \text{ instead of at } g(j,u), \forall j\right]\} \end{aligned}$$

In words, $h^*(i)$ is a measure of how much away from the average cost we are when starting from node i.

3.7.4 Bellman's equation

The following proposition provides the main results regarding Bellman's equation:

Proposition 3.6. Under Assumption 3.5, the following hold for the average cost per stage problem:

(a) The optimal avergae cost λ^* is the same for all initial staes and together with some vector $h^* = \{h^*(1), \dots, h^*(n)\}$ satisfies Bellman's equation

$$\lambda^* + h^*(i) = \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) h^*(j) \right\}, \quad i = 1, \dots, n$$

Furthermore, if $\mu(i)$ attains the minimum in the above equation for all i, the stationary policy μ 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 λ and a vector $h = \{h(1), \dots, h(n)\}$ satisfy Bellman's equation, then λ is the average optimal cost per stage for each initial state.

(c) Given a stationary policy μ with corresponding average cost per stage λ_μ, there is a unique vector $h_\mu = \{h_\mu(1), \dots, h_\mu(n)\}$ such that $h_\mu(n) = 0$ and

$$\lambda_\mu + h_\mu(i) = g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i)) h_\mu(j), \quad i = 1, \dots, n$$

Proof. We proceed item by item:

(a) Let $\tilde{\lambda} = \min_\mu \frac{C_{nn}(\mu)}{N_{nn}(\mu)}$. Then, for all μ,

$$C_{nn}(\mu) - N_{nn}(\mu)\tilde{\lambda} \geq 0$$

with

$$\underbrace{C_{nn}(\mu^*) - N_{nn}(\mu^*)\,\tilde{\lambda}}_{h^*(n) \text{ in the associated SSP}} = 0 \Rightarrow h^*(n) = 0.$$

Consider the associated SSP with stage cost: $g(i,u) - \tilde{\lambda}$. Then, by Proposition 3.2(b), and using the fact that $p_{in}(u) = 0$, the costs $h^*(1), \dots, h^*(n)$ solve uniquely the corresponding Bellman's equation:

$$h^*(i) = \min_{u \in U(i)} \left\{ g(i,u) - \tilde{\lambda} + \sum_{j=1}^{n-1} p_{ij}(u) h^*(j) \right\}, \quad i = 1, \dots, n \tag{3.18}$$

Thus, we can rewrite (3.18) as

$$\tilde{\lambda} + h^*(i) = \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) h^*(j) \right\}, \quad i = 1, \ldots, n \tag{3.19}$$

We will show that this implies $\tilde{\lambda} = \lambda^*$.

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:

$$\begin{aligned} J_0(i) &= h^*(i), \quad i = 1, \ldots, n \\ J_{k+1}(i) &= g\left(i, \mu_{N-k-1}(i)\right) + \sum_{j=1}^{n} p_{ij}\left(\mu_{N-k-1}(i)\right) J_k(j), \quad i = 1, \ldots, n \end{aligned} \tag{3.20}$$

In words, $J_N(i)$ is the N-stage cost of π when starting state is i and the terminal cost is h^*. From (3.19), since $\mu_{N-1}(\cdot)$ is just one admissible policy, we have

$$\tilde{\lambda} + \underbrace{h^*(i)}_{J_0(i)} \le \underbrace{g\left(i, \mu_{N-1}(i)\right) + \sum_{j=1}^{n} p_{ij}\left(\mu_{N-1}(i)\right) h^*(j)}_{J_1(i),\text{ from (3.20), by setting } k=0}, \quad i = 1, \ldots, n$$

Thus,

$$\tilde{\lambda} + J_0(i) \le J_1(i), \quad i = 1, \ldots, n$$

Then,

$$\begin{aligned} J_2(i) &= g\left(i, \mu_{N-2}(i)\right) + \sum_{j=1}^{n} p_{ij}\left(\mu_{N-2}(i)\right) \underbrace{J_1(j)}_{\ge \tilde{\lambda} + J_0(i)} \\ &\ge g\left(i, \mu_{N-2}(i)\right) + \tilde{\lambda} + \sum_{j=1}^{n} p_{ij}\left(\mu_{N-2}(i)\right) J_0(j) \\ &\ge \tilde{\lambda} + \tilde{\lambda} + h^*(i) \quad \text{(by equation (3.19))} \\ &= 2\tilde{\lambda} + h^*(i), \quad i = 1, \ldots, n. \end{aligned}$$

By repeating this argument,

$$k\tilde{\lambda} + h^*(i) \le J_k(i), \quad k = 0, \ldots, N, i = 1, \ldots, n$$

In particular, for $k = N$,

$$N\tilde{\lambda} + h^*(i) \le J_N(i) \Rightarrow \tilde{\lambda} + \frac{h^*(i)}{N} \le \frac{J_N(i)}{N}, \quad i = 1, \ldots, n \tag{3.21}$$

Equality holds in (3.21) if $\mu_k(i)$ attains the minimum in (3.19) for all i, k. Now,

$$\underbrace{\tilde{\lambda} + \frac{h^*(i)}{N}}_{\to\tilde{\lambda} \text{ when } N\to\infty} \leq \underbrace{\frac{J_N(i)}{N},}_{\to J_\pi(i) \text{ when } N\to\infty}$$

where $J_\pi(i)$ is the average cost per stage of π, starting at i. Then, we get

$$\tilde{\lambda} \leq J_\pi(i), \quad i = 1, \ldots, n$$

for all admissible π.

If $\pi = \{\mu, \mu, \ldots\}$ where $\mu(i)$ attains the minimum in (3.19) for all i, k, we get

$$\tilde{\lambda} = \min_\pi J_\pi(i) = \lambda^*, \quad i = 1, \ldots, n$$

Replacing $\tilde{\lambda}$ by λ^* in equation (3.19), we obtain (3.17). Finally, " $h^*(n) = 0$ " jointly with (3.19) are equivalent to (3.18) for the associated SSP. But the solution to (3.18) is unique (due to Proposition 3.2(b)), so there must be a unique solution for the equations " $h^*(n) = 0$ " and (3.19).

(b) The proof follows from the proof of part (a), starting from equation (3.19).

(c) The proof follows from part (a), constraining the control set to $\tilde{U}(i) = \{\mu(i)\}$.

□

Remarks:

- Proposition 3.6 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 μ under which there is a positive probability of reaching j starting from i.

Example 3.5. A manufacturer, at each time:

- May process all unfilled orders at cost $K > 0$, or process no order at all. The cost per unfilled order at each time is $c > 0$.

- 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 $i = 0, 1, \ldots, n-1$,

$$\lambda^* + h^*(i) = \min\{\underbrace{K + (1-p)h^*(0) + ph^*(1)}_{\text{Process unfilled orders}}, \underbrace{ci + (1-p)h^*(i) + ph^*(i+1),}_{\text{Do nothing}}\}$$

  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".

3.7.5 Computational approaches

Value iteration
Procedure: Generate optimal k-stage costs by the DP algorithm starting from any J_0 :

$$J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J_k(j) \right\}, \forall i$$

Claim: $\lim_{k\to\infty} \frac{J_k(i)}{k} = \lambda^*, \forall i.$

Proof. Let h^* be a solution vector of Bellman's equation:

$$\lambda^* + h^*(i) = \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) h^*(j) \right\}, \quad i = 1, \ldots, n \tag{3.22}$$

From here, define the recursion

$$\begin{aligned} J_0^*(i) =& h^*(i) \\ J_{k+1}^*(i) =& \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(i) J_k^*(i) \right\} \end{aligned}$$

Like in the proof of Proposition 3.6(a), it can be shown that

$$J_k^*(i) = k\lambda^* + h^*(i), \quad i = 1, \dots, n$$

On the other hand, it can be seen that

$$|J_k(i) - J_k^*(i)| \leq \max_{j=1,\dots,n} |J_0(j) - h^*(j)|, \quad i = 1, \dots, n$$

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,

$$|J_k(i) - (k\lambda^* - h^*(i))| \leq \max_{j=1,\dots,n} |J_0(j) - h^*(j)|.$$

Therefore,

$$-\max_{j=1,\dots,n} |J_0(j) - h^*(j)| - \underbrace{h^*(i)}_{\leq \max_{j=1,\dots,n}|h^*(j)|} \leq J_k(i) - k\lambda^* \leq \max_{j=1,\dots,n} |J_0(j) - h^*(j)| - \underbrace{h^*(i)}_{\leq \max_{j=1,\dots,n}|h^*(j)|},$$

or equivalently,

$$|J_k(i) - k\lambda^*| \leq \max_{j=1,\dots,n} |J_0(j) - h^*(j)| + \max_{j=1,\dots,n} |h^*(j)|$$

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 (3.22) holds for some vector h^*. □

Remarks:

- Pros: Very simple to implement
- Cons:
  - Since typically some of the components of J_k diverge to ∞ 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^*.

Fixing the difficulties:

- Subtract the same constant from all components of the vector J_k:

$$J_k(i) := J_k(i) - C; \quad i = 1, \dots, n$$

- Consider the algorithm:

$$h_k(i) = J_k(i) - J_k(s)$$

for some fixed state s, and for all $i = 1, \dots, n$. By using equation (3.22) for $i = 1, \dots, n$,

$$\begin{aligned}
h_{k+1}(i) &= J_{k+1}(i) - J_{k+1}(s) \\
&= \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J_k(j) \right\} - \min_{u \in U(s)} \left\{ g(s,u) + \sum_{j=1}^{n} p_{sj}(u) J_k(j) \right\} \\
&= \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) \left(h_k(j) - J_k(s) \right) \right\} - \min_{u \in U(s)} \left\{ g(s,u) + \sum_{j=1}^{n} p_{sj}(u) \left(h_k(j) - J_k(s) \right) \right\} \\
&= \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) h_k(j) \right\} - J_k(s) - \min_{u \in U(s)} \left\{ g(s,u) + \sum_{j=1}^{n} p_{sj}(u) h_k(j) \right\} + J_k(s)
\end{aligned}$$

- The above algorithm is called relative value iteration.
  - Mathematically equivalent to the value iteration method (3.22) 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 3.5 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 3.6(b), this implies that λ is indeed the optimal average cost per stage, and h is the associated differential cost vector.

- Disadvantage: Under Assumption 3.5, 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 $i = 1, \dots, n$, and τ a constant satisfying $0 < \tau < 1$.

Policy iteration

- Start from an arbitrary stationary policy μ^0.
- At a typical iteration, we have a stationary policy μ^k. We perform two steps per iteration:
  - Policy evaluation: Compute λ^k and $h^k(i)$ of μ^k, using the $n+1$ equations $h^k(n) = 0$, and for $i = 1, \dots, n$,
  $$\lambda^k + h^k(i) = g\left(i, \mu^k(i)\right) + \sum_{j=1}^{n} p_{ij}\left(\mu^k(i)\right) h^k(j)$$
  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 μ^{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 3.7. Under Assumption 3.5, in the policy iteration algorithm, for each k we either have $\lambda^{k+1} < \lambda^k$; or else we have

$$\lambda^{k+1} = \lambda^k, \text{ and } h^{k+1}(i) = h^k(i), \forall i$$

Furthermore, the algorithm terminates and the policies μ^k and μ^{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 $N = 1, 2, \dots,$

$$h_N(i) = g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i)) h_{N-1}(j); \quad i = 1, \dots, n$$

where $h_0(i) = h(i)$.

Thus, we have

$$\bar{\lambda} = J_{\bar{\mu}}(i) = \lim_{N \to \infty} \frac{1}{N} h_N(i), \quad i = 1, 2, \dots, n \tag{3.23}$$

By definition of $\bar{\mu}$ we have for all $i = 1, \ldots, n$:

$$\begin{aligned}
h_1(i) &= g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i)) h_0(j) \quad \text{(from the iteration above)} \\
&\leq g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i)) h_0(j) \quad \text{(because } \bar{\mu} \text{ was the min of this RHS)} \\
&= \lambda + h_0(i) \quad \text{(because of Proposition 3.6).}
\end{aligned}$$

From the equation above, we also obtain

$$\begin{aligned}
h_2(i) &= g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i)) h_1(j) \\
&\leq g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i)) \left(\lambda + h_0(j)\right) \\
&= \lambda + g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i)) h_0(j) \\
&\leq \lambda + \underbrace{g(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i)) h_0(j)}_{\lambda + h_0(i)} \\
&= 2\lambda + h_0(i),
\end{aligned}$$

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 (3.23)), and the 2nd term in the RHS goes to zero, implying that $\bar{\lambda} \leq \lambda$.

- If $\bar{\lambda} = \lambda$, the iteration that produces μ^{k+1} is a policy improvement step for the associated SSP with cost per stage $g\left(i, \mu^k\right) - \lambda$. Moreover, $h(i)$ and $\bar{h}(i)$ are the optimal costs starting from i and corresponding to μ 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 λ (each one being the average cost per stage of each of the stationary policies). For each λ there is only a finite number of possible vectors h (see Proposition 3.6(c), where we can vary the reference $h_\mu(n) = 0$).

- In view of the improvement properties already shown, no pair (λ, h) can be repeated without termination of the algorithm, implying that the algorithm must terminate with $\bar{\lambda} = \lambda$ and $\bar{h}(i) = h(i), \forall i$.

$\square$

Claim: When the algorithm terminates, the policies $\bar{\mu}$ and μ are optimal.

Proof. Upon termination, we have for all i,

$$\begin{aligned}
\lambda + h(i) &= \bar{\lambda} + \bar{h}(i) \\
&= g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i))\bar{h}(j) \qquad \text{(by policy evaluation step)} \\
&= g(i, \bar{\mu}(i)) + \sum_{j=1}^{n} p_{ij}(\bar{\mu}(i))h(j) \qquad \text{(because } \bar{h}(j) = h(j), \forall j) \\
&= \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u)h(j) \right\} \qquad \text{(by policy improvement step)}
\end{aligned}$$

Therefore, (λ, h) satisfy Bellman's equation, and by Proposition 3.6(b), λ 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 3.6(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 μ. $\square$

3.8 Semi-Markov Decision Problems

3.8.1 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.

3.8.2 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:

$$Q_{ij}(\tau, u) = \mathbb{P}\left\{t_{k+1} - t_k \leq \tau, x_{k+1} = j \mid x_k = i, u_k = u\right\}$$

- Two important observations:
  1. Transition distributions specify the ordinary transition probabilities via

     $$p_{ij}(u) = \mathbb{P}\left\{x_{k+1} = j \mid x_k = i, u_k = u\right\} = \lim_{\tau \to \infty} Q_{ij}(\tau, u).$$

     We assume that for all states i and controls $u \in U(i)$, the average transition time,

     $$\bar{\tau}_i(u) = \sum_{j=1}^{n} \int_0^\infty \tau Q_{ij}(\,\mathrm{d}\tau, u)$$

     is nonzero and finite, $0 < \bar{\tau}_i(u) < \infty$.

2. The conditional cumulative distribution function (c.d.f.) of τ given i, j, and u is (assuming $p_{ij}(u) > 0$)

$$\mathbb{P}\left\{x_{k+1} = j \mid x_k = i, u_k = u\right\} = \frac{Q_{ij}(u)}{p_{ij}(u)} \tag{3.24}$$

Thus, $Q_{ij}(u)$ can be seen as a scaled c.d.f., i.e.,

$$Q_{ij}(u) = \mathbb{P}\left\{x_{k+1} = j \mid x_k = i, u_k = u\right\} \times p_{ij}(u).$$

Important case: Exponential transition distributions

- Important example of transition distributions:

$$Q_{ij}(\tau, u) = p_{ij}(u)\left(1 - e^{-\nu_i(u)\tau}\right),$$

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 $\operatorname{Exp}(\nu_i(u))$ (independently of j);

$$\mathbb{P}\{\text{transition time interval } > \tau \mid i, u\} = e^{-\nu_i(u)\tau}.$$

- 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.,

$$g(i, u)\mathrm{d}t = \text{cost incurred during small time period } \mathrm{d}t$$

- 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 $\beta > 0$)

$$\lim_{N\to\infty} \mathbb{E}\left[\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} e^{-\beta t} g\left(x_k, \mu_k\left(x_k\right)\right) \mathrm{d}t \mid x_0 = i\right]$$

- Average cost per unit time of $\pi = \{\mu_0, \mu_1, \ldots, \}$ starting from state i

$$\lim_{N \to \infty} \frac{1}{\mathbb{E}\left[t_N \mid x_0 = i, \pi\right]} \mathbb{E}\left[\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} g\left(x_k, \mu_k\left(x_k\right)\right) \mathrm{d}t \middle| x_0 = i\right]$$

- 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 τ.
- Generally, expected values of functions of τ can be written as integrals involving $\mathrm{d}Q_{ij}(\tau, u)$. For example, from (3.24) (noting that there is no τ in the denominator there), the conditional expected value of τ given i, j, and u is written as

$$\mathbb{E}[\tau \mid i, j, u] = \int_0^\infty \tau \frac{\mathrm{d}Q_{ij}(\tau, u)}{p_{ij}(u)}$$

- If $Q_{ij}(\tau, u)$ is discontinuous and "staircase-like", expected values can be written as summations.

3.8.3 Discounted cost problems

- For a policy $\pi = \{\mu_0, \mu_1, \ldots\}$, write

$$J_\pi(i) = \mathbb{E}[\text{cost of 1st transition}] + \mathbb{E}\left[e^{-\beta\tau} J_{\pi_1}(j) \middle| i, \mu_0(i)\right] \tag{3.25}$$

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

$$\begin{aligned} G(i, u) &= \mathbb{E}_j\left[\mathbb{E}_\tau[\text{transition cost} \mid j]\right] \\ &= \sum_{j=1}^n p_{ij}(u) \int_0^\infty \left(\int_0^\tau e^{-\beta t} g(i, u) \mathrm{d}t\right) \frac{\mathrm{d}Q_{ij}(\tau, u)}{p_{ij}(u)} \\ &= \sum_{j=1}^n \int_0^\infty \frac{1 - e^{-\beta\tau}}{\beta} g(i, u) \mathrm{d}Q_{ij}(\tau, u), \end{aligned} \tag{3.26}$$

where the 2nd equality follows from computing $\mathbb{E}_\tau[\text{transition cost} \mid j]$ via integrating the tail of the nonnegative r.v. τ, and the 3 rd one because $\int_0^\tau e^{-\beta t}\, \mathrm{d}t = \left(1 - e^{-\beta\tau}\right)/\beta$. Thus, $\mathbb{E}[\text{cost of 1st transition}]$ is

$$G\left(i, \mu_0(i)\right) = g\left(i, \mu_0(i)\right) \sum_{j=1}^n \int_0^\infty \frac{1 - e^{-\beta\tau}}{\beta} \mathrm{d}Q_{ij}\left(\tau, \mu_0(i)\right)$$

- Regarding the 2nd term in (3.25),

$$\begin{aligned}
\mathrm{E}\left[e^{-\beta\tau} J_{\pi_1}(j) \mid i, \mu_0(i)\right] &= \mathrm{E}_j\left[\mathrm{E}\left[e^{-\beta\tau} \mid j, i, \mu_0(i)\right] J_{\pi_1}(j)\right] \\
&= \sum_{j=1}^{n} p_{ij}\left(\mu_0(i)\right)\left(\int_0^\infty e^{-\beta\tau} \frac{\mathrm{d}Q_{ij}\left(\tau, \mu_0(i)\right)}{p_{ij}\left(\mu_0(i)\right)}\right) J_{\pi_1}(j) \\
&= \sum_{j=1}^{n} m_{ij}\left(\mu_0(i)\right) J_{\pi_1}(j),
\end{aligned}$$

  where $m_{ij}(u)$ is given by

$$m_{ij}(u) = \int_0^\infty e^{-\beta\tau} \mathrm{d}Q_{ij}(\tau, u)$$

  Note that $m_{ij}(u)$ satisfies

$$m_{ij}(u) < \int_0^\infty \mathrm{d}Q_{ij}(\tau, u) = \lim_{\tau\to\infty} Q_{ij}(\tau, u) = p_{ij}(u)$$

  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 (3.25), $J_\pi(i)$ can be written as

$$J_\pi(i) = G\left(i, \mu_0(i)\right) + \sum_{j=1}^{n} m_{ij}\left(\mu_0(i)\right) J_{\pi_1}(j)$$

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 3.6 (Manufacturer's production plan)**.**

- A manufacturer receives orders with interarrival times uniformly distributed in $[0, \tau_{\max}]$.
- 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

$$Q_{i1}(\tau, \text{Fill}) = Q_{i,i+1}(\tau, \text{No Fill}) = \min\left\{1, \frac{\tau}{\tau_{\max}}\right\}$$

- The one-stage expected cost G (see equation (3.26)) is

$$G(i, \text{Fill}) = 0, \quad G(i, \text{Not Fill}) = \gamma c i$$

where

$$\gamma = \sum_{j=1}^{n} \int_0^{\infty} \frac{1 - e^{-\beta\tau}}{\beta} \mathrm{d}Q_{ij}(\tau, u) = \int_0^{\tau \max} \frac{1 - e^{-\beta\tau}}{\beta \tau_{\max}} \mathrm{d}\tau$$

- 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

$$\alpha = \int_0^{\infty} e^{-\beta\tau} \mathrm{d}Q_{ij}(\tau, u) = \int_0^{\tau \max} \frac{e^{-\beta\tau}}{\tau_{\max}} \mathrm{d}\tau = \frac{1 - e^{-\beta\tau \max}}{\beta\tau \max}$$

- Bellman's equation has the form

$$J^*(i) = \min\left\{K + \alpha J^*(1), \gamma c i + \alpha J^*(i+1)\right\}, \quad i = 1, 2, \ldots$$

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^*.

3.8.4 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

$$\lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\int_0^T g(x(t),u(t))\mathrm{d}t\right]$$

  However, we will use instead the cost function

$$\lim_{N\to\infty} \frac{1}{\mathbb{E}\left[t_N\right]}\mathbb{E}\left[\int_0^{t_N} g(x(t),u(t))\mathrm{d}t\right]$$

  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 λ^* 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

$$\bar{\tau}_i(\text{Fill}) = \bar{\tau}_i(\text{Not Fill}) = \frac{\tau_{\max}}{2}$$

- The expected transition cost is

$$G(i,\text{Fill}) = 0, \quad G(i,\text{Not Fill}) = \frac{ci\tau_{\max}}{2}$$

  and the instantaneous cost is

$$\hat{g}(i,\text{Fill}) = K, \quad \hat{g}(i,\text{Not Fill}) = 0$$

- Bellman's equation is

$$h^*(i) = \min\left\{K - \lambda^*\frac{\tau_{\max}}{2} + h^*(1), ci\frac{\tau_{\max}}{2} - \lambda^*\frac{\tau_{\max}}{2} + h^*(i+1)\right\}.$$

- Again, it can be shown that a threshold policy is optimal.

4 Properties and Applications

4.1 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 θ (e.g., θ 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.

4.1.1 Lattices

Definition 4.1. 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 $y : x \wedge y = (\min\{x_1, y_1\}, \ldots, \min\{x_n, y_n\})$,
- Join of x and $y : x \vee y = (\max\{x_1, y_1\}, \ldots, \max\{x_n, y_n\})$.

Then

$$x \wedge y \leq x \leq x \vee y.$$

Definition 4.2. *A set $X \subset \mathbb{R}^n$ is said to be a sublattice of $\mathbb{R}^n$ if $\forall x, y \in X, x \wedge y \in X$ and $x \vee y \in X$.*

Example 4.1.

- $I = \left\{(x, y) \in \mathbb{R}^2 \mid 0 \leq x \leq 1, 0 \leq y \leq 1\right\}$ is a sublattice.
- $H = \left\{(x, y) \in \mathbb{R}^2 \mid x + y = 1\right\}$ 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 4.3. 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 4.1. 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.

4.1.2 Supermodularity and increasing differences

Let $S \subset \mathbb{R}^n, \Theta \subset \mathbb{R}^l$. Suppose that both S and Θ are sublattices.

Definition 4.4. A function $f : S \times \Theta \to \mathbb{R}$ is said to be supermodular in (x, θ) if for all $z = (x, \theta)$ and $z' = (x', \theta')$ in $S \times \Theta$:

$$f(z) + f\left(z'\right) \leq f\left(z \vee z'\right) + f\left(z \wedge z'\right),$$

Similarly, f is submodular if

$$f(z) + f\left(z'\right) \geq f\left(z \vee z'\right) + f\left(z \wedge z'\right).$$

Example 4.2. 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, θ).

Pick any (x, θ) and (x', θ') in $S \times \Theta$, and assume w.l.o.g. $x \geq x'$. There are two cases to consider:

1. $\theta \geq \theta' \Rightarrow (x, \theta) \vee (x', \theta') = (x, \theta)$, and $(x, \theta) \wedge (x', \theta') = (x', \theta')$. Then,

$$\underbrace{f(x, \theta)}_{x\theta} + \underbrace{f\left(x', \theta'\right)}_{x'\theta'} \leq \underbrace{f(\underbrace{(x, \theta) \vee \left(x', \theta'\right)}_{(x,\theta)})}_{x\theta} + \underbrace{f(\underbrace{(x, \theta) \wedge \left(x', \theta'\right)}_{(x',\theta')})}_{x'\theta'}$$

2. $\theta < \theta' \Rightarrow (x, \theta) \vee (x', \theta') = (x, \theta')$, and $(x, \theta) \wedge (x', \theta') = (x', \theta)$. Then,

$$\underbrace{f\left((x, \theta) \vee \left(x', \theta'\right)\right)}_{x\theta'} + \underbrace{f\left((x, \theta) \wedge \left(x', \theta'\right)\right)}_{x'\theta} = x\theta' + x'\theta$$

and we would have

$$\underbrace{f(x, \theta)}_{x\theta} + \underbrace{f\left(x', \theta'\right)}_{x'\theta'} \leq \underbrace{f\left((x, \theta) \vee \left(x', \theta'\right)\right)}_{x\theta'} + \underbrace{f\left((x, \theta) \wedge \left(x', \theta'\right)\right)}_{x'\theta}$$

if and only if

$$\begin{aligned} & x\theta + x'\theta' \leq x\theta' + x'\theta \\ \iff \quad & x\left(\theta' - \theta\right) - x'\left(\theta' - \theta\right) \geq 0 \\ \iff \quad & \underbrace{\left(x - x'\right)}_{\geq 0}\underbrace{\left(\theta' - \theta\right)}_{> 0} \geq 0, \end{aligned}$$

which is indeed the case.

Therefore, $f(x, \theta) = x\theta$ is supermodular in $S \times \Theta$.

Definition 4.5. For $S, \Theta \subset \mathbb{R}$, a function $f : S \times \Theta \to \mathbb{R}$ is said to satisfy increasing differences in (x, θ) if for all pairs (x, θ) and (x', θ') in $S \times \Theta$, if $x \geq x'$ and $\theta \geq \theta'$, then

$$f(x, \theta) - f\left(x', \theta\right) \geq f\left(x, \theta'\right) - f\left(x', \theta'\right).$$

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, θ) if the difference

$$f(x, \theta) - f\left(x', \theta\right), \text{ for } x \geq x'$$

is increasing in θ.

Theorem 4.2. Let $S, \Theta \subset \mathbb{R}$, and suppose $f : S \times \Theta \to \mathbb{R}$ is supermodular in (x, θ). 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 $f(x,\theta) + f(x',\theta) \leq f(x \vee x', \theta) + f(x \wedge x', \theta)$).
2. f satisfies increasing differences in (x, θ).

Proof of Theorem 4.2. For part (1), fix θ. Let $z = (x, \theta), z' = (x', \theta)$. Since f is supermodular in (x, θ) :

$$f(x,\theta) + f\left(x',\theta\right) \leq f\left(x \vee x', \theta\right) + f\left(x \wedge x', \theta\right),$$

or equivalently

$$f(z) + f\left(z'\right) \leq f\left(z \vee z'\right) + f\left(z \wedge z'\right)$$

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, $w \vee w' = z$ and $w \wedge w' = z'$. Since f is supermodular on $S \times \Theta$,

$$f(\underbrace{w}_{(x,\theta')}) + f(\underbrace{w'}_{(x',\theta)}) \leq f(\underbrace{w \vee w'}_{z=(x,\theta)}) + f(\underbrace{w \wedge w'}_{z'=(x',\theta')}).$$

Rearranging terms,

$$f(x,\theta) - f\left(x',\theta\right) \geq f\left(x,\theta'\right) - f\left(x',\theta'\right)$$

and so f also satisfies increasing differences, as claimed. □

Remark 4.1. 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 4.3. 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 of Theorem 4.3. 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 $y_1 > y_2$,

$$\int_{y_2}^{y_1} \int_{x_2}^{x_1} \frac{\partial^2}{\partial x \partial y} h(x, y) \mathrm{d}x \ \mathrm{d}y \geq 0.$$

So,

$$\int_{y_2}^{y_1} \frac{\partial}{\partial y} \left(h\left(x_1, y\right) - h\left(x_2, y\right)\right) \mathrm{d}y \geq 0$$

and thus,

$$h\left(x_1, y_1\right) - h\left(x_2, y_1\right) - \left(h\left(x_1, y_2\right) - h\left(x_2, y_2\right)\right) \geq 0,$$

or equivalently,

$$h\left(x_1, y_1\right) - h\left(x_2, y_1\right) \geq h\left(x_1, y_2\right) - h\left(x_2, y_2\right),$$

which shows that h satisfies increasing differences and hence is supermodular.

$\Rightarrow$) Suppose h is supermodular. Then, it satisfies increasing differences and so, for $x_1 > x_2, y_1 > y$,

$$\frac{h\left(x_1, y_1\right) - h\left(x_1, y\right)}{y_1 - y} \geq \frac{h\left(x_2, y_1\right) - h\left(x_2, y\right)}{y_1 - y}.$$

Letting $y_1 \to y$, we have

$$\frac{\partial}{\partial y} h\left(x_1, y\right) \geq \frac{\partial}{\partial y} h\left(x_2, y\right), \text{ when } x_1 \geq x_2$$

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. $\square$

4.1.3 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 parametric monotonicity we mean that the higher the value of θ, the higher the maximizer $x^*(\theta)$[7].

Let's give some intuition for strictly increasing differences implying 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 θ, and that S is compact). Pick any two values θ_1, θ_2 with $\theta_1 > \theta_2$. Let x_1, x_2 be values that are optimal at θ_1 and θ_2, respectively. Thus,

$$f(x_1, \theta_1) - f(x_2, \theta_1) \geq 0 \geq f(x_1, \theta_2) - f(x_2, \theta_2). \tag{4.1}$$

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, θ_1) and (x_1, θ_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 (4.1). So, we must have $x_1 \geq x_2$, where x_1, x_2 were arbitrary selections from the sets of optimal actions at θ_1 and θ_2, respectively.

In summary, if $S, \Theta \subset \mathbb{R}$, strictly increasing differences imply monotonicity of optimal actions in the parameter $\theta \in \Theta$.

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 $x_1 \ngeq x_2$ implies $x_1 < x_2$.

The following theorem relaxes the "strict" condition of the increasing differences to guarantee parametric monotonicity.

Theorem 4.4. Let S be a compact sublattice of $\mathbb{R}^n$, Θ be a sublattice of $\mathbb{R}^l$, and $f : S \times \Theta \to \mathbb{R}$ be a continuous function on S for each fixed θ. Suppose that f satisfies increasing differences in (x, θ), and is supermodular in x for each fixed θ. Let the correspondence $D^* : \Theta \to S$ be defined by

$$D^*(\theta) = \arg\max\{f(x, \theta) \mid x \in S\}.$$

Then,

[7]Note 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:

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 θ, i.e., $M(\theta) = f(x^*(\theta), \theta)$. Assume that f is continuously differentiable in (x, θ), and that x^* is continuously differentiable in θ. Then,

$$\frac{\partial}{\partial \theta} M(\theta) = \frac{\partial}{\partial \theta} f(y, \theta)\bigg|_{y = x^*(\theta)}.$$

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, θ), 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 of Theorem 4.4. For part (1): Since f is continuous on S for each fixed θ, and since S is compact, $D^*(\theta) \neq \emptyset$ for each θ. Fix θ 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 $p \to \infty$, 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 $x \wedge x' \notin D^*(\theta)$, we must have

$$f\left(x \wedge x', \theta\right) < f(x, \theta) = f\left(x', \theta\right).$$

However, supermodularity in x means

$$\underbrace{f(x, \theta) + f\left(x', \theta\right)}_{2f(x,\theta)} \leq f\left(x' \vee x, \theta\right) + f\left(x' \wedge x, \theta\right) < f\left(x' \vee x, \theta\right) + f(x, \theta),$$

which implies

$$f\left(x' \vee x, \theta\right) > f(x, \theta) = f\left(x', \theta\right),$$

which in turn contradicts the presumed optimality of x and x' at θ. 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 θ_1 and θ_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

$$\begin{aligned}
0 &\leq f(x_1, \theta_1) - f(x_1 \vee x_2, \theta_1) \quad \text{(by optimality of } x_1 \text{ at } \theta_1) \\
&\leq f(x_1 \wedge x_2, \theta_1) - f(x_2, \theta_1) \quad \text{(by supermodularity in } x) \\
&\leq f(x_1 \wedge x_2, \theta_2) - f(x_2, \theta_2) \quad \text{(by increasing differences in } (x, \theta)) \\
&\leq 0 \quad \text{(by optimality of } x_2 \text{ at } \theta_2),
\end{aligned}$$

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

$$f(x_1 \vee x_2, \theta_1) = f(x_1, \theta_1)$$

and so $x_1 \vee x_2$ is also an optimal action at θ_1. If $x_1 \not\supseteq x_2$, then we would have $x_1 \vee x_2 > x_1$, 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 $x_1 \not\supseteq x_2$. Then, $x_2 > x_1 \wedge x_2$. If f satisfies strictly increasing differences, then since $\theta_1 > \theta_2$, we have

$$f(x_2, \theta_1) - f(x_1 \wedge x_2, \theta_1) > f(x_2, \theta_2) - f(x_1 \wedge x_2, \theta_2),$$

so the third inequality in the string above becomes strict, contradicting the equality. □

Remark 4.2. For the case where $S, \Theta \subset \mathbb{R}$, from Theorem 4.2 it can be seen that if f is supermodular, it automatically verifies the hypotheses of Theorem 4.4, and therefore in principle supermodularity in $\mathbb{R}^2$ 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

$$z_i' \geq z_i, \text{ and } z_j' \geq z_j.$$

it is the case that

$$f\left(z_{-ij}, z_i', z_j'\right) - f\left(z_{-ij}, z_i, z_j'\right) \geq f\left(z_{-ij}, z_i', z_j\right) - f\left(z_{-ij}, z_i, z_j\right).$$

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 4.5. A function $f : Z \subset \mathbb{R}^n \to \mathbb{R}$ is supermodular on Z if and only if f has increasing differences on Z.

Proof of Theorem 4.5. The implication " $\Rightarrow$ " can be proved by a slight modification of part (2) in Theorem 4.2. To prove " $\Leftarrow$ ", pick any z and z' in Z. We are required to show that

$$f(z) + f\left(z'\right) \leq f\left(z \vee z'\right) + f\left(z \wedge z'\right).$$

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

$$z \vee z' = \left(z_1', \ldots, z_k', z_{k+1}, \ldots, z_n\right),$$

and

$$z \wedge z' = \left(z_1, \ldots, z_k, z_{k+1}', \ldots, z_n'\right).$$

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

$$z^{i,j} = \left(z'_1, \ldots, z'_i, z_{i+1}, \ldots, z_j, z'_{j+1}, \ldots, z'_n\right).$$

Then, we have

$$z^{0,k} = z \wedge z', \quad z^{k,n} = z \vee z', \quad z^{0,n} = z, \quad z^{k,k} = z'. \tag{4.2}$$

Since f has increasing differences on Z, it is the case that for all $0 \leq i < k \leq j < n$,

$$f\left(z^{i+1,j+1}\right) - f\left(z^{i,j+1}\right) \geq f\left(z^{i+1,j}\right) - f\left(z^{i,j}\right).$$

Therefore, we have for $k \leq j < n$,

$$\begin{aligned} f\left(z^{k,j+1}\right) - f\left(z^{0,j+1}\right) &= \sum_{i=0}^{k-1}\left[f\left(z^{i+1,j+1}\right) - f\left(z^{i,j+1}\right)\right] \\ &\geq \sum_{i=0}^{k-1}\left[f\left(z^{i+1,j}\right) - f\left(z^{i,j}\right)\right] \\ &= f\left(z^{k,j}\right) - f\left(z^{0,j}\right). \end{aligned}$$

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,

$$f\left(z^{k,n}\right) - f\left(z^{0,n}\right) \geq f\left(z^{k,k}\right) - f\left(z^{0,k}\right).$$

From (4.2), this is precisely the statement that

$$f\left(z \vee z'\right) - f(z) \geq f\left(z'\right) - f\left(z \wedge z'\right)$$

Since z and z' were chosen arbitrarily, f is shown to be supermodular on Z. □

Remark 4.3. From Theorem 4.5, it is sufficient to prove supermodularity (or increasing differences) to prove parametric monotonicity.

4.1.4 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 4.3 (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 $k = N, N-1, \ldots, 1$,

$$V_k(x, p) = \max_{0 \le u \le x} \left\{ p \int_0^1 V_{k-1}(x+u, \alpha) \mathrm{d}F(\alpha) + (1-p) \int_0^1 V_{k-1}(x-u, \alpha) \mathrm{d}F(\alpha) \right\}.$$

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.,

$$g_k(u, p) = p \int_0^1 V_{k-1}(x+u, \alpha) \mathrm{d}F(\alpha) + (1-p) \int_0^1 V_{k-1}(x-u, \alpha) \mathrm{d}F(\alpha).$$

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

$$\frac{\partial^2}{\partial u \partial p} g_k(u, p) \ge 0.$$

The derivation proceeds as follows:

$$\frac{\partial}{\partial p} g_k(u, p) = \int_0^1 V_{k-1}(x+u, \alpha) \mathrm{d}F(\alpha) - \int_0^1 V_{k-1}(x-u, \alpha) \mathrm{d}F(\alpha).$$

Then, by the Leibniz rule[8]

$$\frac{\partial^2}{\partial u \partial p} g_k(u,p) = \int_0^1 \frac{\partial}{\partial u} V_{k-1}(x+u,\alpha) \mathrm{d}F(\alpha) - \int_0^1 \frac{\partial}{\partial u} V_{k-1}(x-u,\alpha) \mathrm{d}F(\alpha).$$

Then,

$$\frac{\partial^2}{\partial u \partial p} g_k(u,p) \geq 0$$

if for all α,

$$\frac{\partial}{\partial u} \left[V_{k-1}(x+u,\alpha) - V_{k-1}(x-u,\alpha) \right] \geq 0,$$

or equivalently, if for all α,

$$V_{k-1}(x+u,\alpha) - V_{k-1}(x-u,\alpha)$$

increases in u, which follows if $V_{k-1}(z,\alpha)$ is increasing in z, which immediately holds because for $z' > z$,

$$V_{k-1}\left(z',\alpha\right) \geq V_{k-1}(z,\alpha),$$

since in the former we are maximizing over a bigger domain.

For $V_0(\cdot,\alpha)$, it holds because $G\left(z'\right) \geq G(z)$.

Example 4.4 (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.

[8]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 for that is $\mu_k^*(x,p)$ is continuously differentiable in x.

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:

$$V_0(i) = C(i)$$

and for $k = N, N-1, \ldots, 1$, and $i > 0$,

$$V_k(i) = \min_{u \geq 0} \{u + P(u)V_{k-1}(i-1) + (1 - P(u))V_{k-1}(i)\}. \tag{4.3}$$

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 (4.3). 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

$$\frac{\partial}{\partial u} g_k(i, u) = 1 + P'(u)\left[V_{k-1}(i-1) - V_{k-1}(i)\right]$$

and then

$$\frac{\partial^2}{\partial i \partial u} g_k(i, u) = \underbrace{P'(u)}_{\geq 0} \frac{\partial}{\partial i}\left[V_{k-1}(i-1) - V_{k-1}(i)\right] \leq 0,$$

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 4.6. 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.

Proof of Proposition 4.6. Define

$$\begin{aligned} A_{i,k}: &\quad V_{k+1}(i+1) - V_{k+1}(i) \leq V_k(i+1) - V_k(i), \quad k \geq 0 \\ B_{i,k}: &\quad V_{k+1}(i) - V_k(i) \leq V_{k+2}(i) - V_{k+1}(i), \quad k \geq 0 \\ C_{i,k}: &\quad V_k(i+1) - V_k(i) \leq V_k(i+2) - V_k(i+1), \quad k \geq 0 \end{aligned}$$

We proceed by induction on $n = k + i$. For $n = 0$ (i.e., $k = i = 0$):

$$\begin{aligned} A_{0,0}: &\quad \underbrace{V_1(1)}_{=0 \text{ from } (4.3)} - \underbrace{V_1(0)}_{0} \leq \underbrace{V_0(1)}_{0} - \underbrace{V_0(0)}_{0}, \\ B_{0,0}: &\quad \underbrace{V_1(0)}_{0} - \underbrace{V_0(0)}_{0} \leq \underbrace{V_2(0)}_{0} - \underbrace{V_1(0)}_{0}, \\ C_{0,0}: &\quad \underbrace{V_0(1)}_{C(1)} - \underbrace{V_0(0)}_{C(0)=0} \leq \underbrace{V_0(2)}_{C(2)} - \underbrace{V_0(1)}_{C(1)}, \end{aligned}$$

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 $i = 0 \Rightarrow A_{0,k} : V_{k+1}(1) - \underbrace{V_{k+1}(0)}_{0} \leq V_k(1) - \underbrace{V_k(0)}_{0}$, 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,

$$V_{k+1}(i) - V_k(i) = \bar{u} + P(\bar{u})\left[V_k(i-1) - V_k(i)\right]. \tag{4.4}$$

   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,

$$V_{k+1}(i+1) - V_k(i+1) \leq \bar{u} + P(\bar{u})\left[V_k(i) - V_k(i+1)\right]. \tag{4.5}$$

   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 4.4 and (4.5), 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 $\bar{u}$,

$$V_{k+2}(i) = \bar{u} + P(\bar{u})V_{k+1}(i-1) + (1 - P(\bar{u}))V_{k+1}(i),$$

or equivalently,

$$V_{k+2}(i) - V_{k+1}(i) = \bar{u} + P(\bar{u})\left[V_{k+1}(i-1) - V_{k+1}(i)\right]. \tag{4.6}$$

Also, since $\bar{u}$ is the minimizer for $V_{k+2}(i)$,

$$V_{k+1}(i) \leq \bar{u} + P(\bar{u})V_k(i-1) + (1 - P(\bar{u}))V_k(i),$$

so that

$$V_{k+1}(i) - V_k(i) \leq \bar{u} + P(\bar{u})\left[V_k(i-1) - V_k(i)\right]. \tag{4.7}$$

By IH, $A_{i-1,k}$, for $i - 1 + k = n - 1$, holds. So,

$$V_{k+1}(i) - V_{k+1}(i-1) \leq V_k(i) - V_k(i-1)$$

or equivalently,

$$V_k(i-1) - V_k(i) \leq V_{k+1}(i-1) - V_{k+1}(i).$$

Plugging it in (4.7), and using the RHS of (4.6), 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,

$$2V_k(i+1) \leq V_{k+1}(i+1) + V_{k-1}(i+1). \tag{4.8}$$

Hence, if we can show that,

$$V_{k-1}(i+1) + V_{k+1}(i+1) \leq V_k(i) + V_k(i+2), \tag{4.9}$$

then from (4.8) and (4.9) 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

$$\begin{aligned} V_k(i+2) - V_{k-1}(i+1) &= \bar{u} + P(\bar{u})V_{k-1}(i+1) + (1 - P(\bar{u}))V_{k-1}(i+2) - V_{k-1}(i+1) \\ &= \bar{u} + (1 - P(\bar{u}))\left[V_{k-1}(i+2) - V_{k-1}(i+1)\right]. \end{aligned}$$

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:

$$V_{k+1}(i+1) - V_k(i) \leq \bar{u} + (1 - P(\bar{u}))\left[V_k(i+1) - V_k(i)\right]$$

Then, equation (4.9) will follow if we can prove that

$$V_k(i+1) - V_k(i) \leq V_{k-1}(i+2) - V_{k-1}(i+1), \tag{4.10}$$

because then

$$\begin{aligned} V_{k+1}(i+1) - V_k(i) &\leq \bar{u} + (1 - P(\bar{u}))\left[V_{k-1}(i+2) - V_{k-1}(i+1)\right] \\ &= V_k(i+2) - V_{k-1}(i+1). \end{aligned}$$

Now, from $A_{i,k-1}$ (which holds by IH), it follows that

$$V_k(i+1) - V_k(i) \leq V_{k-1}(i+1) - V_{k-1}(i). \tag{4.11}$$

Also, from $C_{i,k-1}$ (which holds by IH), it follows that

$$V_{k-1}(i+1) - V_{k-1}(i) \leq V_{k-1}(i+2) - V_{k-1}(i+1). \tag{4.12}$$

Finally, (4.11) and (4.12) $\Rightarrow$ (4.10) $\Rightarrow$ (4.9), and we close this case.

In the end, the three inequalities hold, and the proof is completed. $\square$

4.2 Inventory Control

In this section, we study the inventory control problem discussed in Example 2.1.

4.2.1 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:

$$\mathbb{E}\left[\sum_{k=0}^{N-1} \left(cu_k + p \max\left\{0, w_k - x_k - u_k\right\} + h \max\left\{0, x_k + u_k - w_k\right\}\right)\right],$$

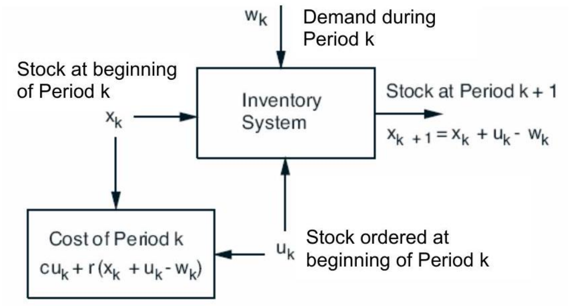
  where the costs are incurred based on the inventory (potentially, negative) available at the end of each period k.
- Suppose that $p > c$ (otherwise, if $c \geq p$, it would never be optimal to buy stock in the last period $N-1$ and possibly in the earlier periods).
- Most of the subsequent analysis 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 ∞, respectively.

Figure 4.1 illustrates the problem setup and the dynamics of the system.
By applying the DP algorithm, we have

$$J_N\left(x_N\right) = 0,$$
$$J_k\left(x_k\right) = \min_{u_k \geq 0} \left\{cu_k + H\left(x_k + u_k\right) + \mathbb{E}_{w_k}\left[J_{k+1}\left(x_k + u_k - w_k\right)\right]\right\},$$

where

$$H(y) = \mathbb{E}\left[r\left(y - w_k\right)\right] = p\mathbb{E}\left[\left(w_k - y\right)^+\right] + h\mathbb{E}\left[\left(y - w_k\right)^+\right].$$



w_k
Demand during Period k
Stock at beginning of Period k
x_k
Inventory System
Stock at Period k + 1
$x_{k+1} = x_k + u_k - w_k$
Cost of Period k
$cu_k + r(x_k + u_k - w_k)$
u_k
Stock ordered at beginning of Period k

Figure 4.1: System dynamics for the inventory control problem.

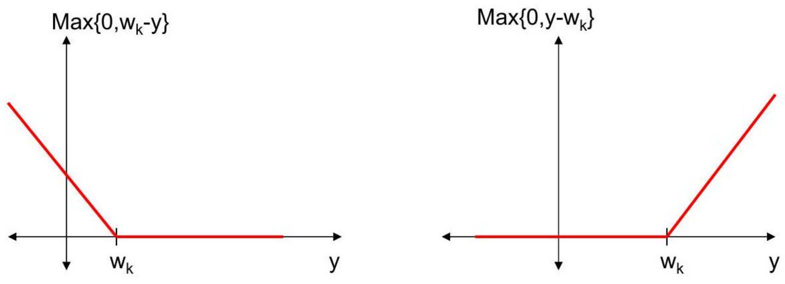


Figure 4.2: Graphical illustration of the two terms in the H function.

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 $y_k = x_k + \underbrace{u_k}_{\geq 0}$ (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

$$J_k(x_k) = \min_{y_k \geq x_k} \{cy_k + H(y_k) + \mathbb{E}_{w_k}[J_{k+1}(y_k - w_k)]\} - cx_k. \tag{4.13}$$

4.2.2 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 4.2 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 (4.13) 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

$$G_k(y) = cy + H(y) + \mathbb{E}_w[J_{k+1}(y - w)].$$

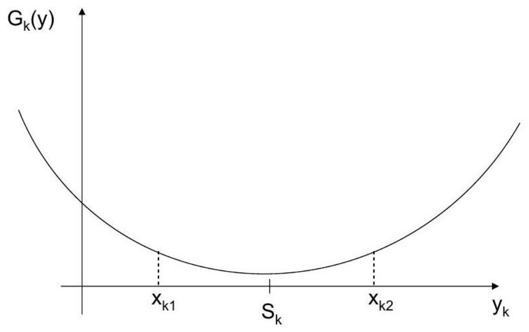


Figure 4.3: The function $G_k(y)$ has a "bowl shape". The minimum for $y_k \geq x_{k_1}$ is S_k; the minimum for $y_k \geq x_{k_2}$ is x_{k_2}.

In addition, if $G_k(y)$ has the shape shown in Figure 4.3, then the minimizer of $G_k(y)$, for $y_k \geq x_k$, is

$$y_k^* = \begin{cases} S_k & \text{if } x_k < S_k \\ x_k & \text{if } x_k \geq S_k \end{cases}$$

- 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

$$\mu_k^*(x_k) = \begin{cases} S_k - x_k & \text{if } x_k < S_k \\ 0 & \text{if } x_k \geq S_k \end{cases} \tag{4.14}$$

This control is known as basestock policy, with basestock level S_k.

To complete the proof of the optimality of the control policy (4.14), we need to prove the next result:

Proposition 4.7. The following three facts hold:

1. The value function $J_{k+1}(y)$ (and hence, $G_k(y)$) is convex in $y, \forall k$.
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 $k = N-1$,

$$G_{N-1}(y) = cy + H(y) + \mathbb{E}_w[\underbrace{J_N(y-w)}_{0}],$$

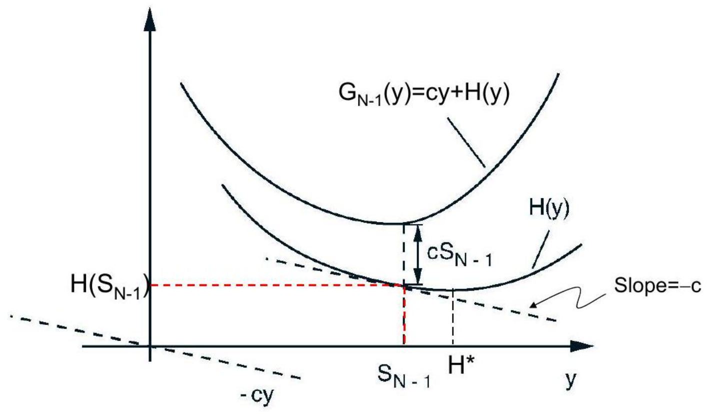


Figure 4.4: The function $G_{N-1}(y)$ is convex with unconstrained minimizer S_{N-1}.

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, $\lim_{|y|\to\infty} G_{N-1}(y) = \infty$[9]. Hence, the optimal control for the last period turns out to be

$$\mu^*_{N-1}(x_{N-1}) = \begin{cases} S_{N-1} - x_{N-1} & \text{if } x_{N-1} < S_{N-1} \\ 0 & \text{if } x_{N-1} \geq S_{N-1} \end{cases} \tag{4.15}$$

and from the DP algorithm in (4.13), we get

$$J_{N-1}(x_{N-1}) = \begin{cases} c\left(S_{N-1} - x_{N-1}\right) + H\left(S_{N-1}\right) & \text{if } x_{N-1} < S_{N-1} \\ H\left(x_{N-1}\right) & \text{if } x_{N-1} \geq S_{N-1} \end{cases} \tag{4.16}$$

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

$$J_{N-1}(x) = \begin{cases} -cx + cS_{N-1} + H\left(S_{N-1}\right) & \text{if } x < S_{N-1} \\ H(x) & \text{if } x \geq S_{N-1} \end{cases}$$

Figure 4.4 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

$$\frac{\partial}{\partial y}G_{N-1}(y) = c + \frac{\partial}{\partial y}H(y)$$

[9]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 $J_{N-1}(\cdot)$.

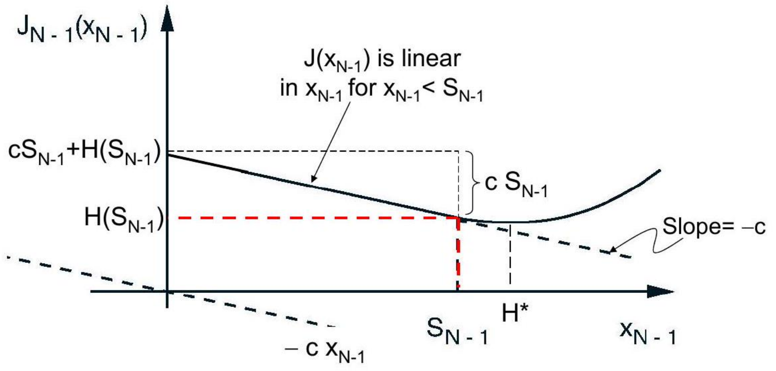


Figure 4.5: The function $J_{N-1}(x_{N-1})$ is convex with unconstrained minimizer H^*.

Evaluating the derivative at S_{N-1}[10], 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 4.5 plots $J_{N-1}(x_{N-1})$. Note that according to (4.16), 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 (4.15) above. □

Continuing with the proof of Proposition 4.7, 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 4.5 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

$$J_k(x_k) = \begin{cases} c(S_k - x_k) + H(S_k) + \mathbb{E}[J_{k+1}(S_k - w_k)] & \text{if } x_k < S_k \\ H(x_k) + \mathbb{E}[J_{k+1}(x_k - w_k)] & \text{if } x_k \geq S_k \end{cases},$$

[10]Note that the function $H(y)$, on a sample path basis, is not differentiable everywhere (see Figure 4.2). 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.

where S_k minimizes $G_k(y) = cy + H(y) + \mathbb{E}\left[J_{k+1}(y-w)\right]$. Furthermore, $J_k(y)$ is convex, $\lim_{|y|\to\infty} J_k(y) = \infty, G_{k-1}(y)$ is convex, and $\lim_{|y|\to\infty} G_{k-1}(y) = \infty$. □

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,

$$\frac{\partial}{\partial x}\mathbb{E}[g(x,w)] = \frac{\partial}{\partial x}\int_{w\in R_w} g(x,w)f_w(w)\mathrm{d}w.$$

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:

$$\frac{\partial}{\partial x}\int_{w\in R_w} g(x,w)f_w(w)\mathrm{d}w = \int_{w\in R_w}\left(\frac{\partial}{\partial x}g(x,w)\right)f_w(w)\mathrm{d}w$$

and so

$$\frac{\partial}{\partial x}\mathbb{E}[g(x,w)] = \mathbb{E}\left[\frac{\partial}{\partial x}g(x,w)\right].$$

4.2.3 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

$$J_N(x_N) = 0,$$
$$J_k(x_k) = \min_{u_k\geq 0}\left\{C(u_k) + H(x_k+u_k) + \mathbb{E}_{w_k}\left[J_{k+1}(x_k+u_k-w_k)\right]\right\},$$

where again

$$H(y) = p\mathbb{E}\left[(w-y)^+\right] + h\mathbb{E}\left[(y-w)^+\right].$$

Consider again

$$G_k(y) = cy + H(y) + \mathbb{E}\left[J_{k+1}(y-w)\right].$$

Then,

$$J_k(x_k) = \min\{\underbrace{G_k(x_k)}_{\text{Do not order}}, \underbrace{\min_{u_k>0}\{K + G_k(x_k+u_k)\}\}}_{\text{Order } u_k} - cx_k.$$

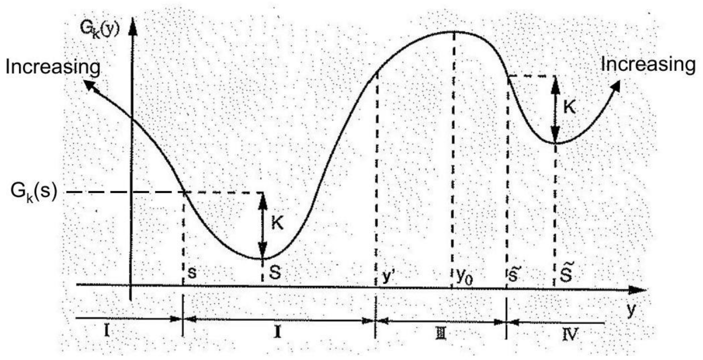


Figure 4.6: Potential form of the function $G_k(y)$ when the fixed cost is nonzero.

By changing variable $y_k = x_k + u_k$ like in the zero fixed-cost case, we get

$$J_k\left(x_k\right) = \min\left\{G_k\left(x_k\right), \min_{y_k > x_k}\left\{K + G_k\left(y_k\right)\right\}\right\} - cx_k. \tag{4.17}$$

When $K > 0, G_k$ is not necessarily convex[11], opening the possibility of very complicated optimal policies (see Figure 4.6). Under this kind of function $G_k(y)$, for the cost function (4.17), the optimal policy would be:

1. If $x_k \in$ Zone I $\Rightarrow G_k\left(x_k\right) > G_k(s), \forall x_k < s \Rightarrow$ Order $u_k^* = S - x_k$, such that $y_k^* = S$. Clearly, $G_k(S) + K < G_k\left(x_k\right), \forall x_k < s$. So, if $x_k \in$ Zone I, $u_k^* = S - x_k$.

2. If $x_k \in$ Zone II $\Rightarrow$

   - If $s < x_k < S$ and $u > 0$ (i.e., $y_k > x_k$) $\Rightarrow K + G_k\left(y_k\right) > G_k\left(x_k\right)$, and it is suboptimal to order.
   - If $S < x_k < y'$ and $u > 0$ (i.e., $y_k > x_k$) $\Rightarrow K + G_k\left(y_k\right) > G_k\left(x_k\right)$, and it is also suboptimal to order.

   So, if $x_k \in$ Zone II, $u_k^* = 0$.

3. If $x_k \in$ Zone III $\Rightarrow$ Order $u_k^* = \tilde{S} - x_k$, so that $y_k^* = \tilde{S}$, and $G_k\left(x_k\right) > K + G(\tilde{S})$, for all $y' < x_k < \tilde{s}$.

4. If $x_k \in$ Zone IV $\Rightarrow$ Do not order (i.e., $u_k^* = 0$), since otherwise $K + G_k\left(y_k\right) > G_k\left(x_k\right), \forall y_k > x_k$.

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.

[11]Note that G_k involves K through J_{k+1}.

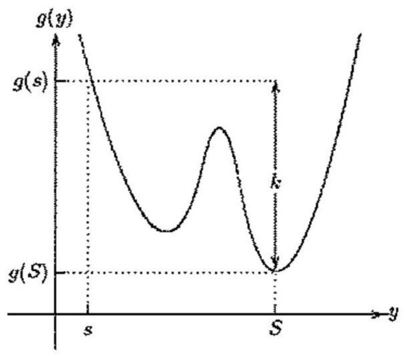


Figure 4.7: Graph of a k-convex function.

Definition 4.6. *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 4.7. Observation: Note that the situation described in Figure 4.6 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 $\tilde{S} = y_0 + z$, and $y = y_0$,

$$K + G_k(\tilde{S}) \geq G_k(y_0) + \underbrace{z}_{\geq 0}\underbrace{\frac{G_k(y_0) - G_k(y_0 - b)}{b}}_{\geq 0} \geq G_k(y_0)$$

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 4.8 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:

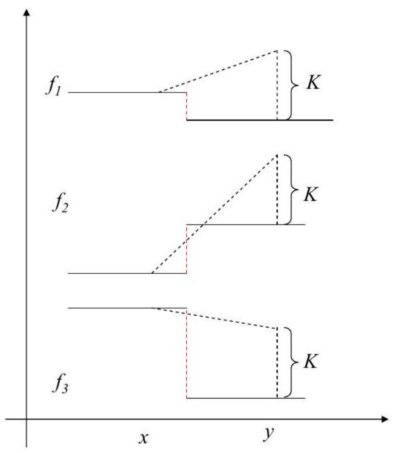


Figure 4.8: Function f_1 is K-convex; f_2 and f_3 are not.

Lemma 4.8. 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 $\mathbb{E}_w[g(y-w)]$ is also K-convex, provided $\mathbb{E}_w[|g(y-w)|] < \infty$, 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 4.8, we will show that the optimal policy is of the form

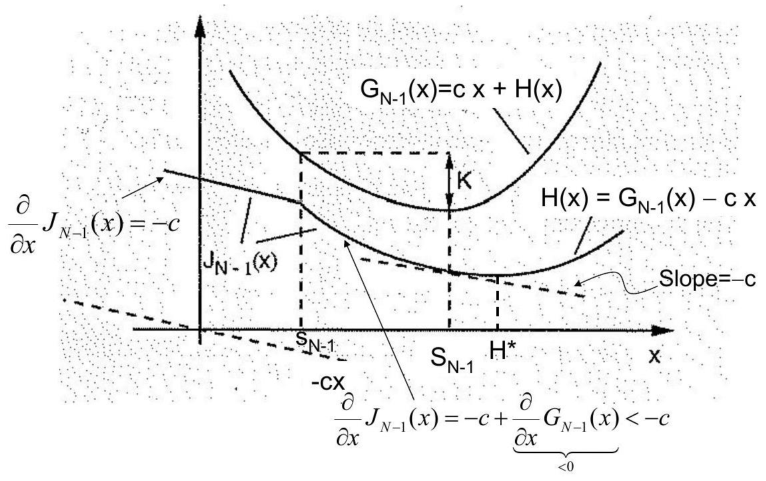
$$\mu_k^*(x_k) = \begin{cases} S_k - x_k & \text{if } x_k < s_k \\ 0 & \text{if } x_k \geq s_k \end{cases}$$

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.

4.2.4 Proof of the optimality of the (s, S) multiperiod policy

For stage $N-1$,

$$G_{N-1}(y) = cy + H(y) + \mathbb{E}_w[\underbrace{J_N(y-w)}_{0}]$$



$G_{N-1}(x)=c\,x + H(x)$
K
$H(x) = G_{N-1}(x) - c\,x$
$\frac{\partial}{\partial x} J_{N-1}(x) = -c$
$J_{N-1}(x)$
Slope=–c
s_{N-1}
S_{N-1}
H^*
x
-cx
$\frac{\partial}{\partial x} J_{N-1}(x) = -c + \underbrace{\frac{\partial}{\partial x} G_{N-1}(x)}_{<0} < -c$

Figure 4.9: Structure of the cost-to-go function when fixed cost is nonzero.

Therefore, $G_{N-1}(y)$ is clearly convex $\Rightarrow$ It is K-convex. Then, we have

$$J_{N-1}(x) = \min\left\{G_{N-1}(x), \min_{y>x}\{K + G_{N-1}(y)\}\right\} - cx,$$

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 4.9), 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

$$J_{N-1}(x) = \begin{cases} K + G_{N-1}(S_{N-1}) - cx & \text{for } x < s_{N-1} \\ G_{N-1}(x) - cx & \text{for } x \geq s_{N-1} \end{cases}. \tag{4.18}$$

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 4.9). Here, as we saw for the zero fixed ordering cost, the minimum 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

$$\begin{aligned} \frac{\partial}{\partial y} G_{N-1}(y) = c + \frac{\partial}{\partial y} H(y) \Rightarrow \\ \frac{\partial}{\partial y} G_{N-1}(S_{N-1}) = c + \frac{\partial}{\partial y} H(S_{N-1}) = 0 \Rightarrow \\ \frac{\partial}{\partial y} H(S_{N-1}) = -c < 0, \end{aligned}$$

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

$$K + J_{N-1}(y+z) \geq J_{N-1}(y) + z \left(\frac{J_{N-1}(y) - J_{N-1}(y-b)}{b} \right) \tag{4.19}$$

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 $y - b \geq s_{N-1} \Rightarrow J_{N-1}(x) = \underbrace{G_{N-1}(x)}_{\text{convex} \Rightarrow K\text{-convex}} - \underbrace{cx}_{\text{linear}}$, so by part (b) of Lemma 4.8, it is K-convex.
- If $y - b < s_{N-1} \Rightarrow$ in view of equation (4.18) we can write (4.19) as

$$\begin{aligned} K + J_{N-1}(y+z) &= K + G_{N-1}(y+z) - c(y+z) \\ &\geq \underbrace{G_{N-1}(y) - cy}_{J_{N-1}(y)} + z \left(\frac{\overbrace{G_{N-1}(y) - cy}^{J_{N-1}(y)} - \overbrace{(K + G_{N-1}(S_{N-1}) - c(y-b))}^{J_{N-1}(y-b)}}{b} \right), \end{aligned}$$

or equivalently,

$$K + G_{N-1}(y+z) \geq G_{N-1}(y) + z \left(\frac{G_{N-1}(y) - G_{N-1}(s_{N-1})}{b} \right) \tag{4.20}$$

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 $b > 0$,

$$K + G_{N-1}(y+z) \geq G_{N-1}(y) + z \left(\frac{G_{N-1}(y) - G_{N-1}(\overbrace{s_{N-1}}^{y-(y-s_{N-1})})}{y - s_{N-1}} \right)$$

Thus, K-convexity hold.

(ii) If y is such that $G_{N-1}(y) < G_{N-1}(s_{N-1}) \Rightarrow$ From part (d-i) in Lemma 4.8, for a scalar $y+z$,

$$\begin{aligned} K + G_{N-1}(y+z) &\geq K + G_{N-1}(S_{N-1}) \\ &= G_{N-1}(s_{N-1}) \quad \text{(by definition of } s_{N-1}) \\ &> G_{N-1}(y) \quad \text{(by hypothesis of this case).} \\ &\geq G_{N-1}(y) + z\left(\frac{G_{N-1}(y) - G_{N-1}(s_{N-1})}{b}\right), \end{aligned}$$

and equation (4.20) holds.

(iii) If $y = s_{N-1}$, then by K-convexity of G_{N-1}, note that (4.20) becomes

$$\begin{aligned} K + G_{N-1}(y+z) &\geq G_{N-1}(y) + z\left(\frac{G_{N-1}(y) - G_{N-1}(s_{N-1})}{b}\right) \\ &= G_{N-1}(y). \end{aligned}$$

From Lemma 4.8, 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 (4.19), the function $J_{N-1}(y)$ is linear $\Rightarrow$ It is K-convex.

Case 3: $y < s_{N-1} < y+z$.

Here, we can write (4.20) as

$$\begin{aligned} & K + J_{N-1}(y+z) \\ =& K + G_{N-1}(y+z) - c(y+z) \\ \geq& J_{N-1}(y) + z\left(\frac{J_{N-1}(y) - J_{N-1}(\overbrace{y-b}^{<y})}{b}\right) \\ =& K + G_{N-1}(S_{N-1}) - cy + z\left(\frac{K + G_{N-1}(S_{N-1}) - cy - (K + G_{N-1}(S_{N-1}) - c(y-b))}{b}\right) \\ =& K + G_{N-1}(S_{N-1}) - cy - \frac{czb}{b} \\ =& K + G_{N-1}(S_{N-1}) - c(y+z) \end{aligned}$$

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 4.8, part (d-i), since $G_{N-1}(\cdot)$ is K-convex.

This completes the proof of the claim. $\square$

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 $G_k(y)$:

$$G_{N-2}(y) = cy + H(y) + \underbrace{\mathbb{E}_w[\underbrace{J_{N-1}(y-w)}_{\substack{K\text{-convex from}\\ \text{Lemma 4.8-(c)}}}]}_{K\text{-convex from Lemma 4.8-(b)}}$$

- $G_{N-2}(y)$ 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 4.8-(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.

4.3 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 costumers classes labeled such that $p_1 > p_2 > \cdots > p_n$.

- Time indices run backwards in time.

- Class n arrives first, followed by classes $n-1, n-2, \cdots, 1$.

- 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.[12]
    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

$$V_j(x) = \mathbb{E}_{D_j}\left[\max_{0 \leq u \leq \min\{D_j, x\}} \{p_j u + V_{j-1}(x-u)\}\right] \tag{4.21}$$

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 $\mathbb{E}[\cdot]$. We prove below that this is w.l.o.g. for the kind of setting that we are dealing with here.

4.3.1 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

$$V_t(x, w_t) = \max_{u(x,w_t) \in U_t(x,w_t)} \mathbb{E}_{w_{t+1}}\left[g_t\left(x, u\left(x, w_t\right), w_t\right) + V_{t+1}\left(f_t\left(x, u\left(x, w_t\right), w_t\right), w_{t+1}\right)\right]$$

is equivalent to

$$V_t(x) = \mathbb{E}_{w_t}\left[\max_{u \in U_t(x,w_t)} g_t\left(x, u, w_t\right) + V_{t+1}\left(f_t\left(x, u, w_t\right)\right)\right] \tag{4.22}$$

which is more convenient to handle.

Proof. State space augmentation argument:

1. Reindex disturbances by defining $\widetilde{w}_t = w_{t+1}, t = 1, \ldots, T-1$.

[12]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.

2. Augment state to include the new disturbance, and define the system equation:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = \begin{pmatrix} f_t\left(x_t, u_t, w_t\right) \\ \widetilde{w}_t \end{pmatrix}$$

3. Starting from $(x_0, y_0) = (x, w_1)$, the standard DP recursion is:

$$V_t(x,y) = \max_{u \in U_t(x,y)} \mathbb{E}_{\widetilde{w}_t}\left[g_t(x,u,y) + V_{t+1}\left(f_t(x,u,y), \widetilde{w}_t\right)\right]$$

4. Define $G_t(x) = \mathbb{E}_{\widetilde{w}_t}\left[V_t\left(x, \widetilde{w}_t\right)\right]$

5. Note that:

$$\begin{aligned} & V_t(x,y) \\ = & \max_{u \in U_t(x,y)} \mathbb{E}_{\widetilde{w}_t}\left[g_t(x,u,y) + V_{t+1}\left(f_t(x,u,y), \widetilde{w}_t\right)\right] \\ = & \max_{u \in U_t(x,y)} \left\{g_t(x,u,y) + \mathbb{E}_{\widetilde{w}_t}\left[V_{t+1}\left(f_t(x,u,y), \widetilde{w}_t\right)\right]\right\} \text{ (because } g_t(\cdot) \text{ does not depend on } \widetilde{w}_t) \\ = & \max_{u \in U_t(x,y)} \left\{g_t(x,u,y) + G_{t+1}\left(f_t(x,u,y)\right)\right\} \text{ (by definition of } G_{t+1}(\cdot)) \end{aligned}$$

6. Replace y by w_t and take expectation with respect to w_t on both sides above to obtain:

$$\underbrace{\mathbb{E}_{w_t}\left[V_t\left(x, w_t\right)\right]}_{G_t(x)} = \mathbb{E}_{w_t}\left[\max_{u(x) \in U_t(x,w_t)} \left\{g_t\left(x,u,w_t\right) + G_{t+1}\left(f_t\left(x,u,w_t\right)\right)\right\}\right]$$

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 $G_t(x) \triangleq \mathbb{E}_{\widetilde{w}_t}\left[V_t\left(x, \widetilde{w}_t\right)\right] = \mathbb{E}_w\left[V_t(x,w)\right]$. 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 (4.22).

$\square$

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 $u^*\left(x, w_t\right)$.
4. We collect the current period reward $g_t\left(x, u\left(x, w_t\right), w_t\right)$.
5. We move to the next state $t+1$.

4.3.2 Structure of the value function

Now, we turn to our original RM problem. First, we define the marginal value of capacity,

$$\Delta V_j(x) = V_j(x) - V_j(x-1) \tag{4.23}$$

and proceed to characterize the structure of the value function.

Proposition 4.9. 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 4.10. Suppose $g : \mathbb{Z}_+ \to \mathbb{R}$ is concave. Let $f : \mathbb{Z}_+ \to \mathbb{R}$ be defined by:

$$f(x) = \max_{a=0,\ldots,m} \{ap + g(x-a)\}$$

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:

   $$f(x) = \hat{f}(x) + px; \text{ where } \hat{f}(x) = \max_{x-m \leq y \leq x} \{-yp + g(y)\}$$

   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. $y^* = \operatorname{argmax}_{y \geq 0} h(y)$. 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

$$\begin{aligned} &\hat{f}(x+1) - \hat{f}(x) \\ =&[-(x+1)p + g(x+1)] - [-xp + g(x)] \\ =&-p + g(x+1) - g(x) \\ \leq&-p + g(x) - g(x-1) \quad \text{(because } g(x) \text{ is concave)} \\ =&\hat{f}(x) - \hat{f}(x-1) \end{aligned}$$

   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

$$\begin{aligned} &\hat{f}(x+1) - \hat{f}(x) \\ =&[-(x+1-m)p + g(x+1-m)] - [-(x-m)p + g(x-m)] \\ =&-p + g(x+1-m) - g(x-m) \\ \leq&-p + g(x-m) - g(x-m-1) \quad \text{(because } g(x) \text{ is concave)} \\ =&\hat{f}(x) - \hat{f}(x-1). \end{aligned}$$

   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 4.9. 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

  $$V_j(x) = \mathbb{E}_{D_j}\left[\max_{0\le u\le \min\{D_j,x\}} \{p_j u + V_{j-1}(x-u)\}\right]$$

  For any realization of D_j, the function

  $$H(x, D_j) = \max_{0\le u\le \min\{D_j,x\}} \{p_j u + V_{j-1}(x-u)\}$$

  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 (4.21), we can express it as follows:

$$\begin{aligned} V_j(x) &= \mathbb{E}_{D_j}\left[\max_{0\le u\le \min\{D_j,x\}} \{p_j u + V_{j-1}(x-u)\}\right] \\ &= V_{j-1}(x) + \mathbb{E}_{D_j}\left[\max_{0\le u\le \min\{D_j,x\}} \left\{\sum_{z=1}^{u} (p_j - \Delta V_{j-1}(x+1-z))\right\}\right], \end{aligned} \tag{4.24}$$

where we are using (4.23) to write $V_{j-1}(x-u)$ as a sum of increments:

$$\begin{aligned} &V_{j-1}(x-u) \\ =&V_{j-1}(x) - \sum_{z=1}^{u} V_{j-1}(x+1-z) \\ =&V_{j-1}(x) - [\Delta V_{j-1}(x) + \Delta V_{j-1}(x-1) + \cdots + \\ &+\Delta V_{j-1}(x+1-(u-1)) + \Delta V_{j-1}(x+1-u)] \\ =&V_{j-1}(x) - [V_{j-1}(x) - V_{j-1}(x-1) + V_{j-1}(x-1) - V_{j-1}(x-2) + \cdots + \\ &+ V_{j-1}(x+2-u) - V_{j-1}(x+1-u) + V_{j-1}(x+1-u) - V_{j-1}(x-u)] \end{aligned}$$

Note that all terms in the RHS except for the last one cancel out. The inner sum in (4.24) is defined to be zero when $u = 0$.

Part (ii): $\Delta V_{j+1}(x) \ge \Delta V_j(x), \forall j$.

From (4.24) we can write:

$$V_{j+1}(x) = V_j(x) + \mathbb{E}_{D_{j+1}}\left[\max_{0\le u\le \min\{D_{j+1},x\}} \left\{\sum_{z=1}^{u} (p_{j+1} - \Delta V_j(x+1-z))\right\}\right].$$

Similarly, we can write:

$$V_{j+1}(x-1) = V_j(x-1) + \mathbb{E}_{D_{j+1}}\left[\max_{0\leq u\leq \min\{D_{j+1},x-1\}}\left\{\sum_{z=1}^{u}(p_{j+1}-\Delta V_j(x-z))\right\}\right]$$

Subtracting both equalities, we get:

$$\begin{aligned}
&\Delta V_{j+1}(x)\\
=&\Delta V_j(x) + \mathbb{E}_{D_{j+1}}\left[\max_{0\leq u\leq \min\{D_{j+1},x\}}\left\{\sum_{z=1}^{u}(p_{j+1}-\Delta V_j(x+1-z))\right\}\right]\\
&-\mathbb{E}_{D_{j+1}}\left[\max_{0\leq u\leq \min\{D_{j+1},x-1\}}\left\{\sum_{z=1}^{u}(p_{j+1}-\Delta V_j(x-z))\right\}\right]\\
\geq&\Delta V_j(x) + \mathbb{E}_{D_{j+1}}\left[\max_{0\leq u\leq \min\{D_{j+1},x\}}\left\{\sum_{z=1}^{u}(p_{j+1}-\Delta V_j(x-z))\right\}\right]\\
&-\mathbb{E}_{D_{j+1}}\left[\max_{0\leq u\leq \min\{D_{j+1},x-1\}}\left\{\sum_{z=1}^{u}(p_{j+1}-\Delta V_j(x-z))\right\}\right]\\
\geq&\Delta V_j(x),
\end{aligned}$$

where the first inequality holds from part (i) in Proposition 4.9, 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. □

4.3.3 Structure of the optimal policy

The good feature of formulation (4.24) is that it is very insightful about the structure of the optimal policy. In particular, from part (i) of Proposition 4.9, 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

$$y_j^* = \max\{x : 0\leq x\leq C, p_{j+1} < \Delta V_j(x)\}, \quad j = 1,2,\ldots,n-1 \tag{4.25}$$

and we assume $y_0^* = 0$ and $y_n^* = C$. Figure (4.10) 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.

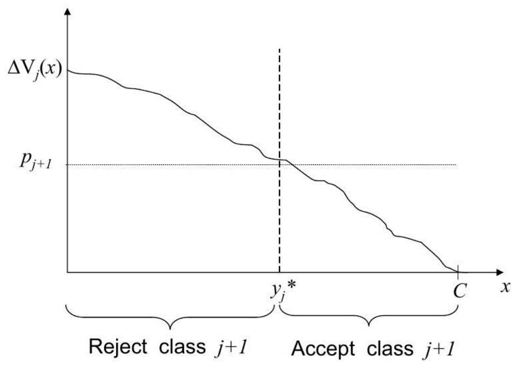


Figure 4.10: Calculation of the optimal protection level y_j^*.

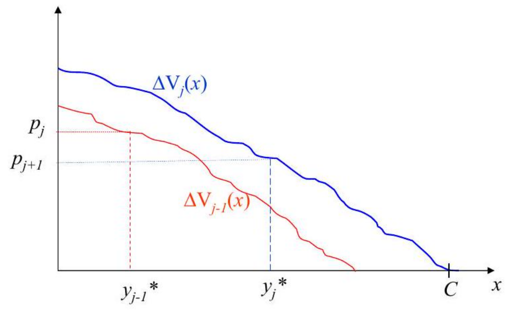


Figure 4.11: Nesting feature of the optimal protection levels.

The optimal control at stage $j+1$ is then

$$\mu_{j+1}^*(x, D_{j+1}) = \min\left\{\left(x - y_j^*\right)^+, D_{j+1}\right\}.$$

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 4.9 implies the nested protection structure

$$y_1^* \le y_2^* \le \cdots \le y_{n-1}^* \le y_n^* = C$$

This is illustrated in Figure 4.11. 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 $y_{j-1}^* \le y_j^*$.

4.3.4 Computational complexity

Using the optimal control, the single-leg RM problem (4.21) could be reformulated as

$$V_j(x) = \mathbb{E}_{D_j}\left[p_j \min\left\{\left(x - y^*_{j-1}\right)^+, D_j\right\} + V_{j-1}\left(x - \min\left\{\left(x - y^*_{j-1}\right)^+, D_j\right\}\right)\right] \tag{4.26}$$

This procedure is repeated starting from $j = 1$ and working backward to $j = n$.

- For discrete-demand distributions, computing the expectation in (4.26) for each state x requires evaluating at most $O(C)$ terms since

$$\min\left\{\left(x - y^*_{j-1}\right)^+, D_j\right\} \leq C.$$

  Since there are C states (capacity levels), the complexity at each stage is $O\left(C^2\right)$.

- The critical values y^*_j can then be identified from (4.25) 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 $\left[y^*_{j-1}, C\right]$. 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 $O\left(nC^2\right)$.

4.3.5 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 (4.12) is a snapshot from Expedia.com showing this practice from American Airlines.

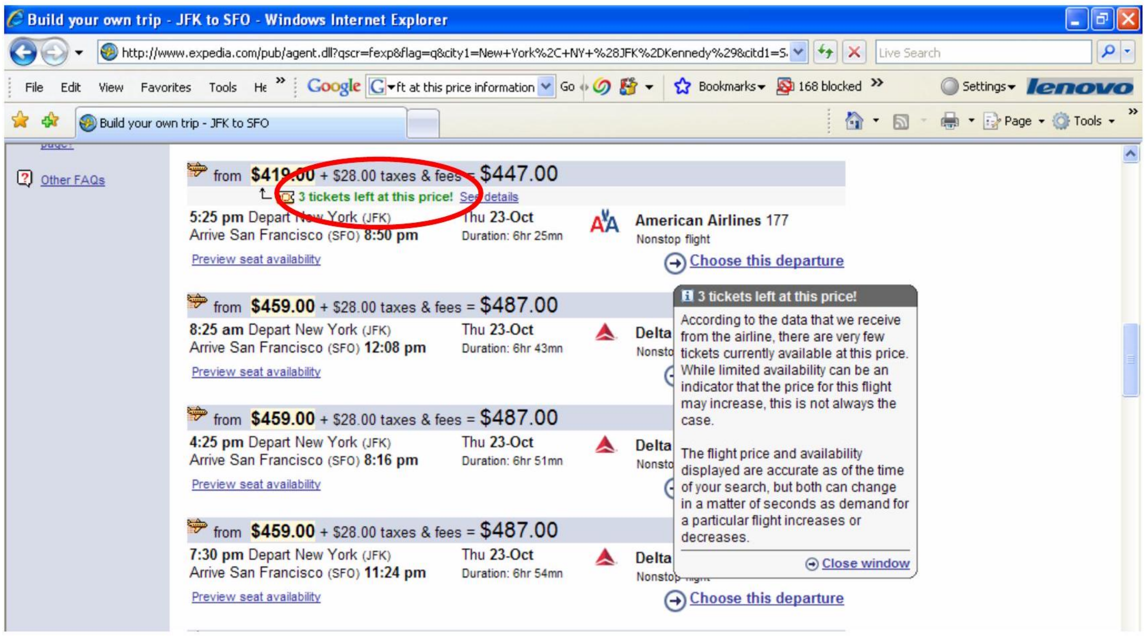
4.4 Optimal Stopping and Scheduling Problems

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.

4.4.1 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 4.13 illustrates this.


Build your own trip - JFK to SFO - Windows Internet Explorer
http://www.expedia.com/pub/agent.dll?qscr=fexp&flag=q&city1=New+York%2C+NY+%28JFK%2DKennedy%29&citd1=S
Live Search
File Edit View Favorites Tools He
Google
Go
Bookmarks
168 blocked
Settings
lenovo
Build your own trip - JFK to SFO
Page
Tools
Other FAQs

from **$419.00** + $28.00 taxes & fees = **$447.00**
3 tickets left at this price! See details
5:25 pm Depart New York (JFK) Thu 23-Oct
Arrive San Francisco (SFO) **8:50 pm** Duration: 6hr 25mn
American Airlines 177
Nonstop flight
Preview seat availability
Choose this departure

from **$459.00** + $28.00 taxes & fees = **$487.00**
8:25 am Depart New York (JFK) Thu 23-Oct
Arrive San Francisco (SFO) **12:08 pm** Duration: 6hr 43mn
Delta
Nonsto
Preview seat availability

from **$459.00** + $28.00 taxes & fees = **$487.00**
4:25 pm Depart New York (JFK) Thu 23-Oct
Arrive San Francisco (SFO) **8:16 pm** Duration: 6hr 51mn
Delta
Nonsto
Preview seat availability

from **$459.00** + $28.00 taxes & fees = **$487.00**
7:30 pm Depart New York (JFK) Thu 23-Oct
Arrive San Francisco (SFO) **11:24 pm** Duration: 6hr 54mn
Delta
Nonstop
Preview seat availability
Choose this departure

3 tickets left at this price!
According to the data that we receive from the airline, there are very few tickets currently available at this price. While limited availability can be an indicator that the price for this flight may increase, this is not always the case.

The flight price and availability displayed are accurate as of the time of your search, but both can change in a matter of seconds as demand for a particular flight increases or decreases.

Close window

Figure 4.12: Optimal protection levels at American Airlines.

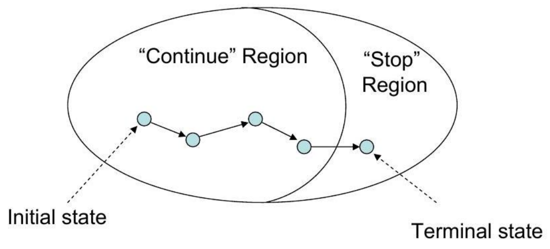


Figure 4.13: Each policy consists of a partition of the state space into the stop and the continue regions.

- Domain of states remains the same throughout the process.

Example 4.5 (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 $w_0, w_1, \dots w_{N-1}$, with $w_i \in [0, \bar{w}]$.
- 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:
$$\mathbb{E}_{w_0,\dots w_{N-1}}\left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k, w_k)\right]$$
where
$$g_N(x_N) = \begin{cases} x_N & \text{if } x_N \neq T, \\ 0 & \text{if } x_N = T \end{cases} \quad \text{(i.e., the seller must accept the offer by time } N\text{)}$$
and for $k = 0, 1, \dots, N-1$,
$$g_k(x_k, \mu_k, w_k) = \begin{cases} (1+r)^{N-k} x_k & \text{if } x_k \neq T \text{ and } \mu_k = u_1 \\ 0 & \text{otherwise} \end{cases}$$
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.

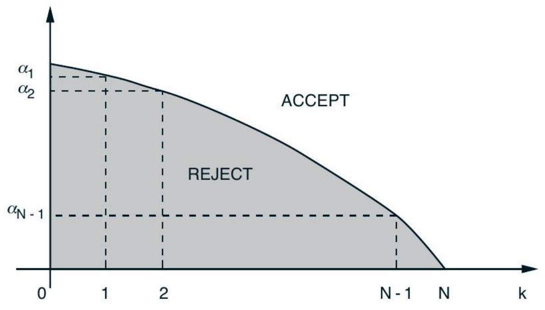


Figure 4.14: Optimal policy of accepting/rejecting offers in the asset selling problem.

- DP formulation

$$J_N(x_N) = \begin{cases} x_N & \text{if } x_N \neq T \\ 0 & \text{if } x_N = T \end{cases} \tag{4.27}$$

For $k = 0, 1, \ldots, N-1$,

$$J_k(x_k) = \begin{cases} \max\{\underbrace{(1+r)^{N-k}x_k}_{\text{Sell}}, \underbrace{\mathbb{E}[J_{k+1}(w_k)]}_{\text{Wait}}\} & \text{if } x_k \neq T \\ 0 & \text{if } x_k = T \end{cases} \tag{4.28}$$

- Optimal policy: Accept offer only when

$$x_k > \alpha_k \triangleq \frac{\mathbb{E}[J_{k+1}(w_k)]}{(1+r)^{N-k}}$$

Note that α_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 4.14 represents this threshold structure.

Proposition 4.11. 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

$$V_k(x_k) \triangleq \frac{J_k(x_k)}{(1+r)^{N-k}}, \quad x_k \neq T$$

We can rewrite equations (4.27) and (4.28) as follows:

$$\begin{aligned} V_N(x_N) &= x_N \\ V_k(x_k) &= \max\left\{x_k, (1+r)^{-1}\mathbb{E}_w[V_{k+1}(w)]\right\}, \quad k = 0, \ldots, N-1 \end{aligned} \tag{4.29}$$

Hence, defining $\alpha_N = 0$ (since we have to accept no matter what in the last period), we get

$$\alpha_k = \frac{\mathbb{E}_w\left[V_{k+1}(w)\right]}{1+r}, k = 0, 1, \ldots, N-1$$

Next, we compare the value function at periods $N-1$ and N : For $k = N$ and $k = N-1$, we have

$$V_N(x) = x$$
$$V_{N-1}(x) = \max\{x, \underbrace{(1+r)^{-1}\mathbb{E}_w\left[V_N(w)\right]}_{\alpha_{N-1}}\} \geq V_N(x)$$

Given that we have a stationary system, from the monotonicity of DP, we know that

$$V_1(x) \geq V_2(x) \geq \cdots \geq V_N(x), \quad \forall x$$

Since $\alpha_k = \frac{\mathbb{E}_w[V_{k+1}(w)]}{1+r}$ and $\alpha_{k+1} = \frac{\mathbb{E}_w[V_{k+2}(w)]}{1+r}$, we have $\alpha_k \geq \alpha_{k+1}$. $\square$

Compute limiting α

Next, we explore the question: What if the selling horizon is very long? Note that equation (4.29) can be written as $V_k\left(x_k\right) = \max\left\{x_k, \alpha_k\right\}$, where

$$\begin{aligned}\alpha_k &= (1+r)^{-1}\mathbb{E}_w\left[V_{k+1}(w)\right] \\ &= \frac{1}{1+r}\int_0^{\alpha_{k+1}} \alpha_{k+1} \mathrm{~d}F(w) + \frac{1}{r+1}\int_{\alpha_{k+1}}^{\infty} w \mathrm{~d}F(w) \\ &= \frac{\alpha_{k+1}}{1+r}F\left(\alpha_{k+1}\right) + \frac{1}{1+r}\int_{\alpha_{k+1}}^{\infty} w \mathrm{~d}F(w)\end{aligned} \tag{4.30}$$

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 $k = 0, 1, \ldots, N-1$,
$$0 \leq \frac{1}{1+r}\int_{\alpha_{k+1}}^{\infty} w \mathrm{~d}F(w) \leq \frac{1}{1+r}\int_0^{\infty} w \mathrm{~d}F(w) = \frac{\mathbb{E}[w]}{1+r}$$
3. From equation (4.30) and Proposition 4.11:
$$\alpha_k \leq \frac{\alpha_{k+1}}{1+r} + \frac{\mathbb{E}[w]}{1+r} \leq \alpha_k\frac{1}{1+r} + \frac{\mathbb{E}[w]}{1+r} \Rightarrow \alpha_k < \frac{\mathbb{E}[w]}{r}$$

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

$$(1+r)\bar{\alpha} = F(\bar{\alpha})\bar{\alpha} + \int_{\bar{\alpha}}^{\infty} w dF(w)$$

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

$$G(\alpha) \triangleq \frac{1}{r+1}\left(F(\alpha)\alpha + \int_{\alpha}^{\infty} w \ \mathrm{d}F(w)\right)$$

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 $[0, \mathbb{E}[w]/r]$, starting from any arbitrary $\alpha_0 \in [0, \mathbb{E}[w]/r]$.

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.

Example 4.6 (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 $k = 0, \ldots, N-1$,

$$J_k(x_k) = \begin{cases} \min\{ \underbrace{x_k}_{\text{Purchase}}, \underbrace{\mathbb{E}\left[J_{k+1}(w_k)\right]}_{\text{Wait}}\} & \text{if } x_k \neq T, \\ 0 & \text{if } x_k = T. \end{cases}$$

- Optimal policy: Purchase if and only if

$$x_k < \alpha_k \triangleq \mathbb{E}_w\left[J_{k+1}(w)\right]$$

where

$$\alpha_k \triangleq \mathbb{E}_w\left[J_{k+1}(w)\right] = \mathbb{E}_w\left[\min\{w, \alpha_{k+1}\}\right] = \int_0^{\alpha_{k+1}} w \ \mathrm{d}F(w) + \int_{\alpha_{k+1}}^{\infty} \alpha_{k+1} \ \mathrm{d}F(w).$$

With terminal condition:

$$\alpha_{N-1} = \int_0^{\infty} w \ \mathrm{d}F(w) = \mathbb{E}[w]$$

Analogously to the asset selling problem, it must hold that

$$\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_{N-1} = \mathbb{E}[w]$$

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

$$x_{k+1} = \lambda x_k + \xi_k, \text{ where } 0 < \lambda < 1$$

and where $\xi_1, \xi_2, \ldots, \xi_{N-1}$ are i.i.d. with $\mathbb{E}[\xi] = \bar{\xi} > 0$.

DP Algorithm:

$$J_N(x_N) = x_N$$
$$J_k(x_k) = \min\left\{x_k, \mathbb{E}\left[J_{k+1}(x_k\lambda + \xi_k)\right]\right\}, \quad k = 0, \ldots, N-1.$$

In particular, for $k = N-1$, we have

$$\begin{aligned} J_{N-1}(x_{N-1}) &= \min\left\{x_{N-1}, \mathbb{E}\left[J_N(x_{N-1}\lambda + \xi_{N-1})\right]\right\} \\ &= \min\left\{x_{N-1}, x_{N-1}\lambda + \bar{\xi}\right\} \end{aligned}$$

Optimal policy at time $N-1$: Purchase only when $x_{N-1} < \alpha_{N-1}$, where α_{N-1} comes from

$$x_{N-1} < \lambda x_{N-1} + \bar{\xi} \quad \Longleftrightarrow \quad x_{N-1} < \alpha_{N-1} \triangleq \frac{\bar{\xi}}{1-\lambda}$$

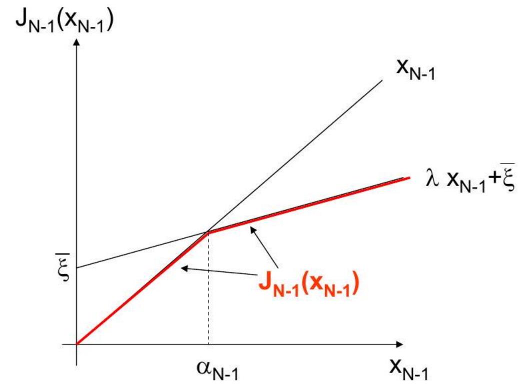


Figure 4.15: Structure of the value function $J_{N-1}(x)$ when prices are correlated.

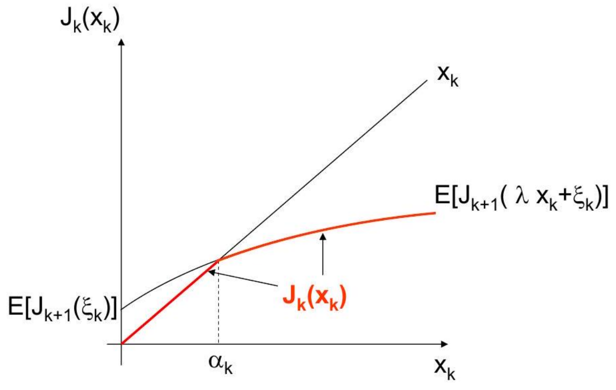


Figure 4.16: Structure of the value function $J_k\left(x_k\right)$ when prices are correlated.

In addition, we can see that

$$J_{N-1}(x) = \min\{x, \lambda x + \bar{\xi}\} \le x = J_N(x)$$

Using the stationarity of the system and the monotonicity property of DP, we have that for any x, $J_k(x) \le J_{k+1}(x), k = 0, \ldots, N-1$.

Moreover, $J_{N-1}(x)$ is concave and increasing in x (see Figure 4.15). 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 4.16). 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 α_k is the unique positive solution of the equation

$$x = \mathbb{E}\left[J_{k+1}\left(\lambda x + \xi_k\right)\right]$$

Notice that the relation $J_k(x) \le J_{k+1}(x)$ for all x and k implies that

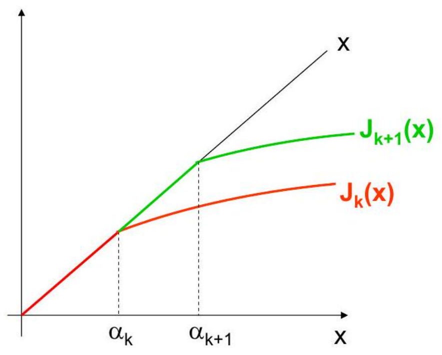


Figure 4.17: Structure of the value functions $J_k(x)$ and $J_{k+1}(x)$ when prices are correlated.

$$\alpha_k \leq \alpha_{k+1}, \quad k = 0, \ldots, N-2$$

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 4.17.

4.4.2 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

$$T_k = \left\{ x : t(x) \leq \min_{u \in U(x)} \mathbb{E}\left[g(x, u, w) + J_{k+1}(f(x, u, w)) \right] \right\} \tag{4.31}$$

- Note that $J_{N-1}(x) \leq J_N(x), \forall x$. This holds because

$$J_{N-1}(x) = \min \left\{ t(x), \min_{u_{N-1} \in U(x)} \mathbb{E}\left[g(x, u_{N-1}, w) + J_N(f(x, u_{N-1}, w)) \right] \right\} \leq t(x) = J_N(x)$$

Using the monotonicity of the DP, we have $J_k(x) \leq J_{k+1}(x), k = 0, 1, \ldots, N-1$. Since

$$T_{k+1} = \left\{ x : t(x) \leq \min_{u \in U(x)} \mathbb{E}\left[g(x, u, w) + J_{k+2}(f(x, u, w)) \right] \right\} \tag{4.32}$$

and the RHS in (4.31) is less or equal than the RHS in (4.32), we have

$$T_0 \subseteq T_1 \subseteq \cdots \subseteq T_k \subseteq T_{k+1} \subseteq \cdots \subseteq T_{N-1} \tag{4.33}$$

- 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,

$$f(x,u,w) \in T_{N-1}, \quad \forall x \in T_{N-1}, u \in U(x) \text{ and } w \tag{4.34}$$

  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 $x \in T_{N-1}$,

$$\begin{aligned} &\min_{u \in U(x)} \mathbb{E}\left[g(x,u,w) + J_{N-1}(f(x,u,w))\right] \\ =&\min_{u \in U(x)} \mathbb{E}[g(x,u,w) + t(f(x,u,w))] \\ \geq& t(x) \text{ (because of (4.31) applied to } k = N-1) \end{aligned}$$

  Since

$$J_{N-2}(x) = \min\left\{t(x), \min_{u \in U(x)} \mathbb{E}\left[g(x,u,w) + J_{N-1}(f(x,u,w))\right]\right\}$$

  then $x \in T_{N-2}$, or equivalently $T_{N-1} \subseteq T_{N-2}$. This, together with (4.33), implies $T_{N-1} = T_{N-2}$. Proceeding similarly, we obtain $T_k = T_{N-1}, \forall k$.

Conclusion: If condition (4.34) 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-steplook-ahead policy. Such a policy turns out to be optimal in several types of applications.

Example 4.7 (Asset selling with past offers retained)**.** Take the previous asset selling problem in Section 4.4.1, 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 $k = 0, 1, \ldots, N-1$,

$$V_k(x_k) = \max\left\{x_k, (1+r)^{-1}\mathbb{E}_{w_k}\left[V_{k+1}\left(\max\left\{x_k, w_k\right\}\right]\right\}$$

The one-step stopping set is:

$$T_{N-1} = \left\{x : x \geq (1+r)^{-1}\mathbb{E}_w[\max\{x, w\}]\right\}$$

Define $\bar{\alpha}$ as the x that satisfies the equation

$$x = \frac{\mathbb{E}_w[\max\{x, w\}]}{1+r}$$

so that $T_{N-1} = \{x : x \geq \bar{\alpha}\}$. Thus,

$$\begin{aligned}\bar{\alpha} &= \frac{1}{1+r}\mathbb{E}_w[\max\{\bar{\alpha}, w\}] \\ &= \frac{1}{1+r}\left(\int_0^{\bar{\alpha}} \bar{\alpha}\mathrm{d}F(w) + \int_{\bar{\alpha}}^{\infty} w\ \mathrm{d}F(w)\right) \\ &= \frac{1}{1+r}\left(\bar{\alpha}F(\bar{\alpha}) + \int_{\bar{\alpha}}^{\infty} w\ \mathrm{d}F(w)\right),\end{aligned}$$

or equivalently,

$$(1+r)\bar{\alpha} = \bar{\alpha}F(\bar{\alpha}) + \int_{\bar{\alpha}}^{\infty} w\ \mathrm{d}F(w)$$

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 (4.34). 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.

4.4.3 Scheduling problem

- Consider a given 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 4.8 (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

  $$L \triangleq (i_0, i_1, \ldots, i_{k-1}, i, j, i_{k+2}, \ldots, i_{N-1})$$

  We have

  $$\begin{aligned}&\mathbb{E}[\text{Reward}(L)]\\ =&\mathbb{E}\left[\text{Reward}\left(i_0, \ldots, i_{k-1}\right)\right] + p_{i_0} \cdots p_{i_1} \cdots p_{i_{k-1}} \left(p_i R_i + p_i p_j R_j\right) + \mathbb{E}\left[\text{Reward}\left(i_{k+2}, \ldots, i_{N-1}\right)\right]\end{aligned}$$

Consider the list L, now with i and j interchanged, and let:

$$L' \triangleq (i_0, \ldots, i_{k-1}, j, i, i_{k+2}, \ldots, i_{N-1})$$

Since L is optimal,

$$\mathbb{E}[\text{Reward}(L)] \geq \mathbb{E}\left[\text{Reward}\left(L'\right)\right]$$

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 $p_i R_i / (1 - p_i)$.

5 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.

5.1 Reduction to the perfect information case

5.1.1 Basic problem with imperfect state information

- Suppose that the controller has access to observations z_k of the form

$$z_0 = h_0(x_0, v_0), \quad z_k = h_k(x_k, u_{k-1}, v_k), \quad k = 1, 2, \dots, N-1$$

where

$$\begin{aligned} z_k &\in Z_k \quad \text{(observation space)} \\ v_k &\in V_k \quad \text{(random observation disturbances)} \end{aligned}$$

The random observation disturbance v_k is characterized by a probability distribution

$$P_{v_k}(. \mid x_k, \dots, x_0, u_{k-1}, \dots, u_0, w_{k-1}, \dots, w_0, v_{k-1}, \dots, v_0)$$

- 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

$$I_k = (z_0, z_1, \dots, z_k, u_0, u_1, \dots u_{k-1})$$

Consider a class of policies consisting of a sequence of functions $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$ where $\mu_k(I_k) \in U_k$ for all $I_k, k = 0, 1, \dots, N-1$.

Objective: Find an admissible policy $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$ that minimizes the cost function

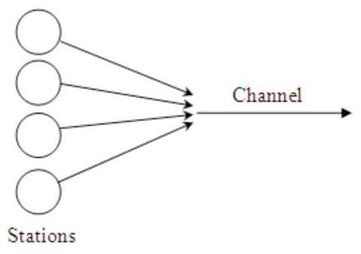
$$J_\pi = \mathbb{E}_{\substack{x_0, w_k, v_k \\ k=0,1,\dots,N-1}} \left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(I_k), w_k) \right]$$

subject to the system equation

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

and the measurement equation

$$\begin{aligned} z_0 &= h_0(x_0, v_0) \\ z_k &= h_k(x_k, \mu_{k-1}(I_{k-1}), v_k), \quad k = 1, 2, \dots, N-1 \end{aligned}$$



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 5.1 (Multiaccess Communication)**.**

- Consider a group of transmitting stations sharing a common channel
- Stations are synchronized to transmit packets of data at integer times
- Each packet requires one slot (time unit) for transmission

  Let a_k = Number of packet arrivals during slot k (with a given probability distribution)

  x_k = Number of packet waiting to be transmitted at the beginning of slot k (backlog)
- 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 $g_k(\cdot)$ 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

$$t_k = \begin{cases} 1 \quad \text{(success)} & \text{w.p. } x_k u_k (1-u_k)^{x_k-1} \text{ (i.e., } \mathbb{P}(\text{one Tx}, x_k - 1 \text{ do not Tx})) \\ 0 \quad \text{(failure)} & \text{w.p. } 1 - x_k u_k (1-u_k)^{x_k-1} \end{cases}$$

- Measurement equation:

$$z_{k+1} = v_{k+1} = \begin{cases} \text{“idle”} & \text{w.p.} \quad (1-u_k)^{x_k} \\ \text{“succes”} & \text{w.p.} \quad x_k u_k (1-u_k)^{x_k-1} \\ \text{“collision”} & \text{w.p.} \quad 1-(1-u_k)^{x_k} - x_k u_k (1-u_k)^{x_k-1} \end{cases}$$

where z_{k+1} is the observation obtained at the end of the k th slot.

5.1.2 Reformulated as a perfect information problem

Candidate for state is the information vector I_k

$$I_{k+1} = (I_k, z_{k+1}, u_k), k = 0, 1, \ldots, N-2, I_0 = z_0$$

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

$$\mathbb{P}(z_{k+1} \mid I_k, u_k) = \mathbb{P}(z_{k+1} \mid I_k, u_k, z_0, z_1, \ldots, z_k)$$

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:

$$\tilde{g}_k(I_k, u_k) = \mathbb{E}_{x_k, w_k}[g_k(x_k, u_k, w_k) \mid I_k, u_k]$$

Note that the new formulation is focused on past info and controls rather than on original system disturbances w_k.

- DP algorithm:

$$J_{N-1}(I_{N-1}) = \min_{u_{N-1} \in U_{N-1}} \{\mathbb{E}_{x_{N-1}, w_{N-1}}[g_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1})) \\ + g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) \mid I_{N-1}, u_{N-1}]\}$$

and for $k = 0, 1, \ldots, N-2$,

$$J_k(I_k) = \min_{u_k \in U_k} \left\{ \mathbb{E}_{x_k, w_k, z_{k+1}} \left[g_k(x_k, u_k, w_k) + J_{k+1}(\underbrace{I_k, z_{k+1}, u_k}_{I_{k+1}}) \mid I_k, u_k \right] \right\}$$

We minimize the RHS for every possible value of the vector I_{k+1} to obtain $\mu_{k+1}^*(I_{k+1})$. The optimal cost is given by $J_0^* = \mathbb{E}_{z_0}[J_0(z_0)]$.

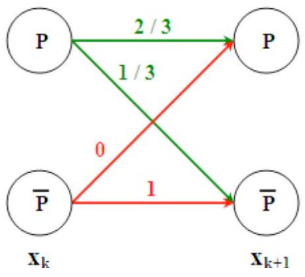
Example 5.2 (Machine repair).

- A machine can be in one of two unobservable states: P (good state) and $\bar{P}$ (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: $\mathbb{P}(x_0 = P) = \frac{2}{3}, \mathbb{P}(x_0 = \bar{P}) = \frac{1}{3}$
- Assume that we start with a machine in good state, i.e., $x_{-1} = P$
- System equation:

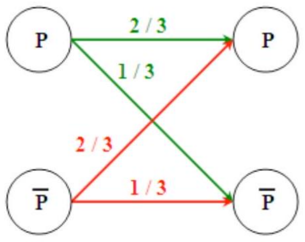
  $$x_{k+1} = w_k, \quad k = 0, 1$$

  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.

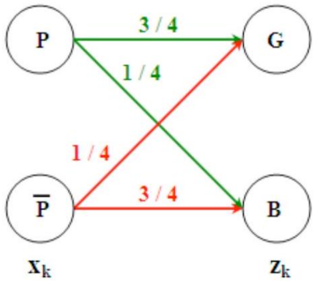
  Result of inspections: $z_k = v_k, k = 0, 1; v_k \in \{B, G\}$



(a) Control: C



(b) Control: S



- 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

$$\mathbb{E}_{\substack{x_0,w_0,w_1\\ v_0,v_1}}\left[g\left(x_0, \mu_0(\underbrace{z_0}_{I_0})\right) + g\left(x_1, \mu_1(\underbrace{z_0, z_1, \mu_0(z_0)}_{I_1})\right)\right]$$

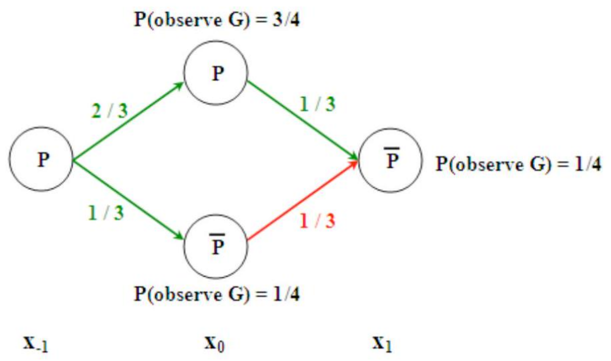
DP algorithm. Terminal condition: $J_2(I_2) = 0$ for all I_2
For $k = 0, 1$:

$$J_k(I_k) = \min\{\overbrace{\mathbb{P}(x_k = P \mid I_k, C)\underbrace{g(P,C)}_{0} + \mathbb{P}(x_k = \bar{P} \mid I_k, C)\underbrace{g(\bar{P},C)}_{2} + \mathbb{E}_{z_{k+1}}[J_{k+1}(I_k, z_{k+1}, C) \mid I_k, C]}^{\text{cost if } u_k = C},$$

$$\overbrace{\mathbb{P}(x_k = P \mid I_k, S)\underbrace{g(P,S)}_{1} + \mathbb{P}(x_k = \bar{P} \mid I_k, S)\underbrace{g(\bar{P},S)}_{1} + \mathbb{E}_{z_{k+1}}[J_{k+1}(I_k, z_{k+1}, S) \mid I_k, S]}^{\text{cost if } u_k = S}\}$$

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

$$\text{cost of } C = 2\mathbb{P}\left(x_1 = \bar{P} \mid I_1\right), \quad \text{cost of } S = 1$$



and therefore $J_1(I_1) = \min\{2\mathbb{P}(x_1 = \bar{P} \mid I_1), 1\}$. Compute probability $\mathbb{P}(x_1 = \bar{P} \mid I_1)$ for all possible realizations of $I_1 = (z_0, z_1, u_0)$ by using the conditional probability formula:

$$\mathbb{P}(X \mid A, B) = \frac{\mathbb{P}(X, A \mid B)}{\mathbb{P}(A \mid B)}$$

There are 8 cases to consider. We describe here 3 of them.

(1) For $I_1 = (G, G, S)$

$$\mathbb{P}(x_1 = \bar{P} \mid G, G, S) = \frac{\mathbb{P}(x_1 = \bar{P}, G, G \mid S)}{\mathbb{P}(G, G \mid S)} = \frac{1}{7}$$

Numerator:

$$\mathbb{P}(x_1 = \bar{P}, G, G \mid S) = \left(\frac{2}{3} \times \frac{3}{4} \times \frac{1}{3} \times \frac{1}{4}\right) + \left(\frac{1}{3} \times \frac{1}{4} \times \frac{1}{3} \times \frac{1}{4}\right) = \frac{7}{144}$$

Denominator:

$$\mathbb{P}(G, G \mid S) = \left(\frac{2}{3} \times \frac{3}{4} \times \frac{2}{3} \times \frac{3}{4}\right) + \left(\frac{2}{3} \times \frac{3}{4} \times \frac{1}{3} \times \frac{1}{4}\right) + \left(\frac{1}{3} \times \frac{1}{4} \times \frac{2}{3} \times \frac{3}{4}\right) + \left(\frac{1}{3} \times \frac{1}{4} \times \frac{1}{3} \times \frac{1}{4}\right) = \frac{49}{144}$$

Hence,

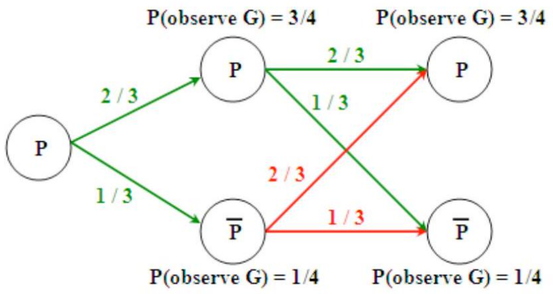
$$J_1(G, G, S) = 2\mathbb{P}(x_1 = \bar{P} \mid G, G, S) = \frac{2}{7} < 1, \mu_1^*(G, G, S) = C$$

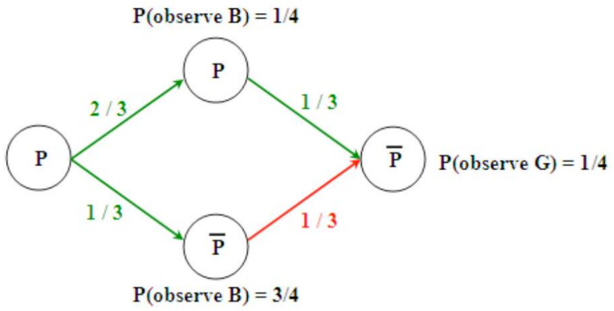
(2) For $I_1 = (B, G, S)$

$$\mathbb{P}(x_1 = \bar{P} \mid B, G, S) = \frac{1}{7}$$

Numerator:

$$\mathbb{P}(x_1 = \bar{P}, B, G \mid S) = \frac{1}{4} \times \frac{1}{3} \times \left(\frac{1}{4} \times \frac{2}{3} + \frac{3}{4} \times \frac{1}{3}\right) = \frac{5}{144}$$





Denominator:

$$\mathbb{P}\left(x_1 = \bar{P}, B, G \mid S\right) = \frac{2}{3} \times \frac{1}{4} \times \left(\frac{2}{3} \times \frac{3}{4} \times + \frac{1}{3} \times \frac{1}{4}\right) + \frac{1}{3} \times \frac{3}{4} \times \left(\frac{2}{3} \times \frac{3}{4} + \frac{1}{3} \times \frac{1}{4}\right)$$

Hence,

$$J_1(B, G, S) = \frac{2}{7}, \mu_1^*(B, G, S) = C$$

(3) For $I_1 = (G, B, S)$

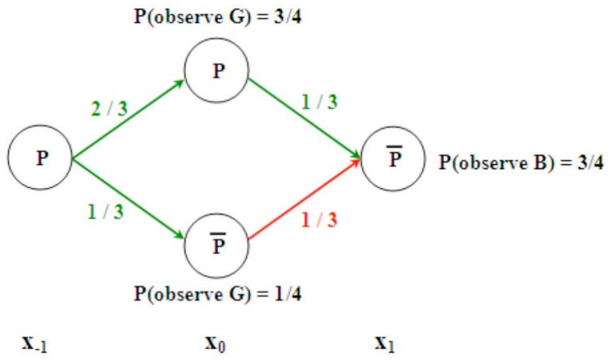
$$\mathbb{P}\left(x_1 = \bar{P} \mid G, B, S\right) = \frac{\mathbb{P}\left(x_1 = \bar{P}, G, B \mid S\right)}{\mathbb{P}(G, B \mid S)} = \frac{3}{5}$$

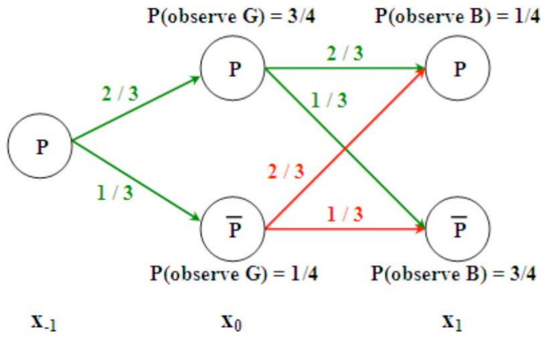
Numerator:

$$\mathbb{P}\left(x_1 = \bar{P}, G, B \mid S\right) = \frac{3}{4} \times \frac{1}{3} \times \left(\frac{3}{4} \times \frac{2}{3} + \frac{1}{4} \times \frac{1}{3}\right) = \frac{7}{48}$$

Denominator:

$$\mathbb{P}(G, B \mid S) = \frac{1}{4} \times \frac{2}{3} \times \left(\frac{3}{4} \times \frac{2}{3} + \frac{1}{4} \times \frac{1}{3}\right) + \frac{3}{4} \times \frac{1}{3} \times \left(\frac{3}{4} \times \frac{2}{3} + \frac{1}{4} \times \frac{1}{3}\right) = \frac{35}{144}$$





Hence,

$$J_1(G,B,S)=1, \mu_1^*(G,B,S)=S$$

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 $\overline{(u_1=C)}$ 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

$$\begin{aligned}
\text{cost of } C &= 2\mathbb{P}\left(x_0=\bar{P} \mid I_0, C\right)+\mathbb{E}_{z_1}\left\{J_1\left(I_0, z_1, C\right) \mid I_0, C\right\} \\
&= 2\mathbb{P}\left(x_0=\bar{P} \mid I_0, C\right)+\mathbb{P}\left(z_1=G \mid I_0, C\right) J_1\left(I_0, G, C\right)+\mathbb{P}\left(z_1=B \mid I_0, C\right) J_1\left(I_0, B, C\right) \\
\text{cost of } S &= 1+\mathbb{E}_{z_1}\left\{J_1\left(I_0, z_1, S\right) \mid I_0, S\right\} \\
&= 1+\mathbb{P}\left(z_1=G \mid I_0, S\right) J_1\left(I_0, G, S\right)+\mathbb{P}\left(z_1=B \mid I_0, S\right) J_1\left(I_0, B, S\right)
\end{aligned}$$

using the values of J_1 from previous stage. Thus, we have

$$J_0\left(I_0\right)=\min \{\text{ cost of } C, \text{ cost of } S\}$$

The optimal cost is

$$J^*=\mathbb{P}(G) J_0(G)+\mathbb{P}(B) J_0(B)$$

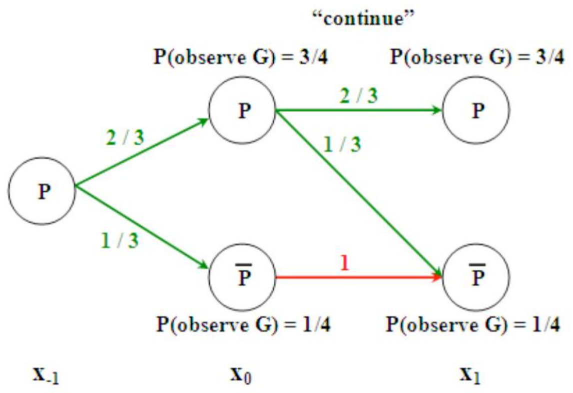
For illustration, we compute one of the values. For example, for $I_0 = G$

$$\mathbb{P}\left(z_1 = G \mid G, C\right) = \frac{\mathbb{P}(z_1 = G, \overbrace{G}^{z_0} \mid \overbrace{C}^{u_0})}{\mathbb{P}(\underbrace{G}_{z_0} \mid C)} = \frac{\mathbb{P}\left(z_1 = G, G \mid C\right)}{\mathbb{P}(G)} = \frac{\frac{15}{48}}{\frac{7}{12}} = \frac{15}{28}$$

Note that the $\mathbb{P}(G \mid C) = \mathbb{P}(G)$ follows since $z_0 = G$ is independent of the control $u_0 = C$ or $u_0 = S$

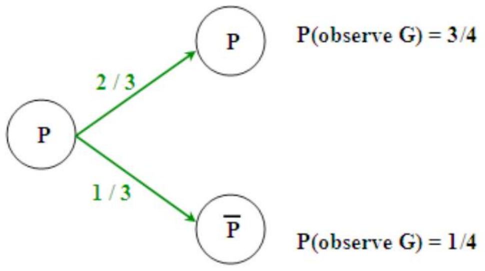
Numerator:

$$\mathbb{P}\left(z_1 = G, G \mid C\right) = \frac{2}{3} \times \frac{3}{4} \times \left(\frac{2}{3} \times \frac{3}{4} + \frac{1}{3} \times \frac{1}{4}\right) + \frac{1}{3} \times \frac{1}{4} \times 1 \times \frac{1}{4} = \frac{15}{48}$$

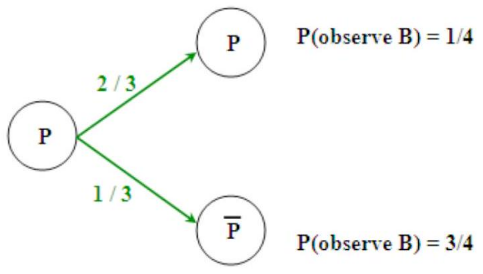


Denominator:

$$\mathbb{P}(G) = \frac{2}{3} \times \frac{3}{4} + \frac{1}{3} \times \frac{1}{4} = \frac{7}{12}$$



Similarly, we can compute $\mathbb{P}(B) = \frac{2}{3} \times \frac{1}{4} + \frac{1}{3} \times \frac{3}{4} = \frac{5}{12}$



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.

5.2 Linear-Quadratic Systems and Sufficient Statistics

In this section, we consider again the problem studied in Section **??**, 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.

5.2.1 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:

$$\mathbb{E}_{x_0, w_0, \dots, w_{N-1}} \left\{ x_N' Q_N x_N + \sum_{k=0}^{N-1} \left(x_k' Q_k x_k + u_k' R_k u_k \right) \right\}$$

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 $\mathbb{E}[w_k] = 0$, 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:

$$\underbrace{z_k}_{\in \mathbb{R}^s} = \underbrace{C_k}_{\in \mathbb{R}^{s \times n}} x_k + \underbrace{v_k}_{\in \mathbb{R}^s}, \quad k = 0, 1, \dots, N-1$$

where v_k 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 $\left\{\mu_0^*, \ldots, \mu_{N-1}^*\right\}$ is of the form

$$\mu_k^*\left(I_k\right) = L_k \mathbb{E}\left[x_k \mid I_k\right]$$

where

- L_k is the same as for the perfect state info case, and solves the "control problem".
- $\mathbb{E}\left[x_k \mid I_k\right]$ solves the "estimation problem".

This means that the control and estimation problems can be solved separately.
DP algorithm The DP algorithm becomes:
At stage $N-1$,

$$\begin{aligned} & J_{N-1}\left(I_{N-1}\right) \\ = & \min_{u_{N-1}} \mathbb{E}_{x_{N-1}, w_{N-1}}\left[x_{N-1}' Q_{N-1} x_{N-1} + u_{N-1}' R_{N-1} u_{N-1}\right. \\ & \left. + \left(A_{N-1} x_{N-1} + B_{N-1} u_{N-1} + w_{N-1}\right)' Q_N \left(A_{N-1} x_{N-1} + B_{N-1} u_{N-1} + w_{N-1}\right) \mid I_{N-1}, u_{N-1}\right] \end{aligned} \tag{5.1}$$

- 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

$$\mathbb{E}\left[w_{N-1} \mid I_{N-1}, u_{N-1}\right] = \mathbb{E}\left[w_{N-1} \mid I_{N-1}\right] = \mathbb{E}\left[w_{N-1}\right] = 0,$$

then the minimization just involves

$$\min_{u_{N-1}} \left\{ u_{N-1}' (\underbrace{B_{N-1}' Q_N B_{N-1}}_{\geq 0} + \underbrace{R_{N-1}}_{>0}) u_{N-1} + 2\mathbb{E}\left[x_{N-1} \mid I_{N-1}\right]' A_{N-1}' Q_N B_{N-1} u_{N-1} \right\}$$

Taking derivative of the argument with respect to u_{N-1}, we have the first order condition:

$$2\left(B_{N-1}' Q_N B_{N-1} + R_{N-1}\right) u_{N-1} + 2\mathbb{E}\left[x_{N-1} \mid I_{N-1}\right]' A_{N-1}' Q_N B_{N-1} = 0$$

This yields the optimal u_{N-1}^* :

$$u_{N-1}^* = \mu_{N-1}^*\left(I_{N-1}\right) = L_{N-1} \mathbb{E}\left[x_{N-1} \mid I_{N-1}\right]$$

where

$$L_{N-1} = -\left(B_{N-1}' Q_N B_{N-1} + R_{N-1}\right)^{-1} B_{N-1}' Q_N A_{N-1}$$

Note that this is very similar to the perfect state info counterpart, except that now x_{N-1} is replaced by $\mathbb{E}\left[x_{N-1} \mid I_{N-1}\right]$.

- Substituting back in (5.1), we get:

$$\begin{aligned}
&J_{N-1}\left(I_{N-1}\right)\\
=&\mathbb{E}_{x_{N-1}}\left[x_{N-1}'K_{N-1}x_{N-1} \mid I_{N-1}\right] \qquad \text{(quadratic in } x_{N-1}\text{)}\\
&+\mathbb{E}_{x_{N-1}}\left[\left(x_{N-1}-\mathbb{E}\left[x_{N-1} \mid I_{N-1}\right]\right)' P_{N-1}\left(x_{N-1}-\mathbb{E}\left[x_{N-1} \mid I_{N-1}\right]\right) \mid I_{N-1}\right]\\
&\quad \text{(quadratic in estimation error } x_{N-1}-\mathbb{E}\left[x_{N-1} \mid I_{N-1}\right]\text{)}\\
&+\mathbb{E}_{w_{N-1}}\left[w_{N-1}'Q_N w_{N-1}\right] \qquad \text{(constant term)}
\end{aligned}$$

where the matrices K_{N-1} and P_{N-1} are given by

$$P_{N-1}=A_{N-1}'Q_N B_{N-1}\left(R_{N-1}+B_{N-1}'Q_N B_{N-1}\right)^{-1}B_{N-1}'Q_N A_{N-1},$$

and

$$K_{N-1}=A_{N-1}'Q_N A_{N-1}-P_{N-1}+Q_{N-1}$$

- 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

$$x_{N-1}-\mathbb{E}\left[x_{N-1} \mid I_{N-1}\right].$$

In words, the estimation error is penalized quadratically in the value function.

At stage $N-2$,

$$\begin{aligned}
&J_{N-2}\left(I_{N-2}\right)\\
=&\min_{u_{N-2}}\mathbb{E}_{x_{N-2},w_{N-2},z_{N-1}}\left[x_{N-2}'Q_{N-2}x_{N-2}+u_{N-2}'R_{N-2}u_{N-2}+J_{N-1}\left(I_{N-1}\right) \mid I_{N-2},u_{N-2}\right]\\
=&\mathbb{E}\left[x_{N-2}'Q_{N-2}x_{N-2} \mid I_{N-2}\right]+\min_{u_{N-2}}\left\{u_{N-2}'R_{N-2}u_{N-2}+\mathbb{E}\left[x_{N-1}'K_{N-1}x_{N-1} \mid I_{N-2},u_{N-2}\right]\right\}\\
&+\mathbb{E}\left[\left(x_{N-1}-\mathbb{E}\left[x_{N-1} \mid I_{N-1}\right]\right)' P_{N-1}\left(x_{N-1}-\mathbb{E}\left[x_{N-1} \mid I_{N-1}\right]\right) \mid I_{N-2},u_{N-2}\right] \qquad (5.2)\\
&+\mathbb{E}_{w_{N-1}}\left[w_{N-1}'Q_N w_{N-1}\right]
\end{aligned}$$

Key point (to be proved): The term (5.2) 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 $x_k-\mathbb{E}\left[x_k \mid I_k\right]$ 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 5.1 (Quality of Estimation)**.** For every stage k, there is a function $M_k(\cdot)$ such that

$$M_k\left(x_0,w_0,\ldots,w_{k-1},v_0,\ldots,v_k\right)=x_k-\mathbb{E}\left[x_k \mid I_k\right]$$

independently of the policy being used.

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

$$x_{k+1} = A_k x_k + B_k u_k + w_k, z_k = C_k x_k + v_k.$$

2. There is no control being applied, and the system evolves according to

$$\bar{x}_{k+1} = A_k \bar{x}_k + \bar{w}_k, \bar{z}_k = C_k \bar{x}_k + \bar{v}_k. \tag{5.3}$$

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:

$$w_k = \bar{w}_k, v_k = \bar{v}_k, k = 0, 1, \ldots, N-1.$$

Consider the vectors:

$$Z^k = (z_0, \ldots, z_k)', \bar{Z}^k = (\bar{z}_0, \ldots, \bar{z}_k)', W^k = (w_0, \ldots, w_k)'$$
$$V^k = (v_0, \ldots, v_k)', \text{ and } U^k = (u_0, \ldots, u_k)'$$

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:

$$x_k = F_k x_0 + G_k U^{k-1} + H_k W^{k-1} \tag{5.4}$$

$$\bar{x}_k = F_k x_0 + H_k W^{k-1} \tag{5.5}$$

Note that the vector $U^{k-1} = (u_0, \ldots, u_{k-1})'$ is part of the information vector I_k, as verified below:

$$I_k = (z_0, \ldots, z_k, \underbrace{u_0, \ldots, u_{k-1}}_{U^{k-1}}), \quad k = 1, \ldots, N-1,$$

$$I_0 = z_0.$$

Then, $U^{k-1} = \mathbb{E}\left[U^{k-1} \mid I_k\right]$, and conditioning with respect to I_k in (5.4) and (5.5):

$$\mathbb{E}[x_k \mid I_k] = F_k \mathbb{E}[x_0 \mid I_k] + G_k U^{k-1} + H_k \mathbb{E}\left[W^{k-1} \middle| I_k\right] \tag{5.6}$$

$$\mathbb{E}[\bar{x}_k \mid I_k] = F_k \mathbb{E}[x_0 \mid I_k] + H_k \mathbb{E}\left[W^{k-1} \middle| I_k\right] \tag{5.7}$$

Then,

$$\underbrace{x_k}_{\text{from (5.4)}} - \underbrace{\mathbb{E}[x_k \mid I_k]}_{\text{from (5.6)}} = \underbrace{\bar{x}_k}_{\text{from (5.5)}} - \underbrace{\mathbb{E}[\bar{x}_k \mid I_k]}_{\text{from (5.7)}}$$

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, $\mathbb{E}[x_k \mid I_k]$.

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 (5.3) 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, and D_k are matrices of appropriate dimension. Thus, the information provided by $I_k = \left(Z^k, U^{k-1}\right)$ regarding $\bar{x}_k$ is summarized in $\bar{Z}^k$, and we have

$$\mathbb{E}\left[\bar{x}_k \mid I_k\right] = \mathbb{E}\left[\bar{x}_k \middle| \bar{Z}^k\right]$$

so that

$$\begin{aligned} x_k - \mathbb{E}\left[x_k \mid I_k\right] &= \bar{x}_k - \mathbb{E}\left[\bar{x}_k | I_k\right] \\ &= \bar{x}_k - \mathbb{E}\left[\bar{x}_k \middle| \bar{Z}^k\right]. \end{aligned}$$

Therefore, the function M_k to use is

$$M_k\left(x_0, w_0, \ldots, w_{k-1}, v_0, \ldots, v_k\right) = \bar{x}_k - \mathbb{E}\left[\bar{x}_k \middle| \bar{Z}_k\right]$$

which does not depend on the controls $u_0, \ldots, u_{k-1}$. □

Going back to the DP equation $J_{N-2}\left(I_{N-2}\right)$, and using the Quality of Estimation Lemma, we get

$$\xi_{N-1} \triangleq M_{N-1}\left(x_0, w_0, \ldots, w_{N-2}, v_0, \ldots, v_{N-1}\right) = x_{N-1} - \mathbb{E}\left[x_{N-1} \mid I_{N-1}\right]$$

Since ξ_{N-1} is independent of u_{N-2}, we have

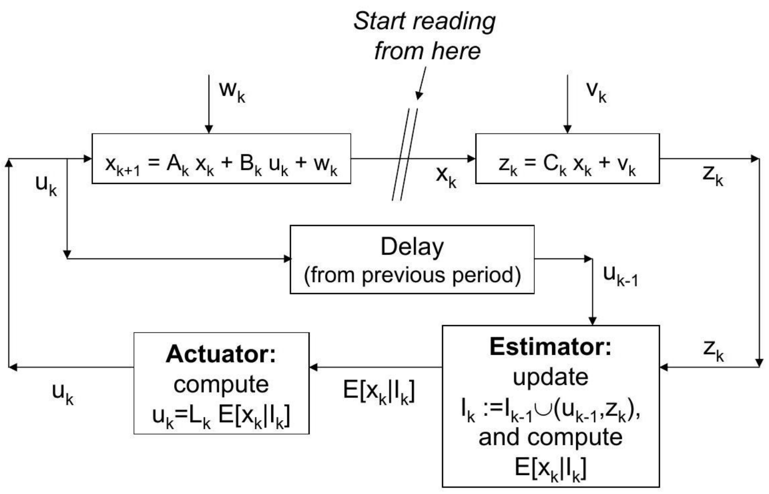
$$\mathbb{E}\left[\xi'_{N-1} P_{N-1} \xi_{N-1} \mid I_{N-2}, u_{N-2}\right] = \mathbb{E}\left[\xi'_{N-1} P_{N-1} \xi_{N-1} \mid I_{N-2}\right]$$

So, going back to the DP equation for $J_{N-2}\left(I_{N-2}\right)$, we can drop the term (5.2) to minimize over u_{N-2}, and similarly to stage $N-1$, the minimization yields

$$u^*_{N-2} = \mu^*_{N-2}\left(I_{N-2}\right) = L_{N-2}\mathbb{E}\left[x_{N-2} \mid I_{N-2}\right]$$

Continuing similarly (using also the Quality of Estimation Lemma), we have

$$\mu^*_k\left(I_k\right) = L_k \mathbb{E}\left[x_k \mid I_k\right]$$



Start reading from here
w_k
v_k
$x_{k+1} = A_k x_k + B_k u_k + w_k$
x_k
$z_k = C_k x_k + v_k$
z_k
u_k
Delay (from previous period)
u_{k-1}
Actuator: compute $u_k=L_k E[x_k|I_k]$
$E[x_k|I_k]$
Estimator: update $I_k := I_{k-1} \cup (u_{k-1}, z_k)$, and compute $E[x_k|I_k]$
z_k
u_k

Figure 5.1: Structure of the optimal controller for the L-Q problem.

where L_k is the same as for perfect state info:

$$L_k = -\left(R_k + B_k' K_{k+1} B_k\right)^{-1} B_k' K_{k+1} A_k$$

with K_k generated from $K_N = Q_N$, using

$$K_k = A_k' K_{k+1} A_k - P_k + Q_k$$
$$P_k = A_k' K_{k+1} B_k \left(R_k + B_k' K_{k+1} B_k\right)^{-1} B_k' K_{k+1} A_k$$

The optimal controller is represented in Figure 5.1.

Separation interpretation:

1. The optimal controller can be decomposed into:
   - An estimator, which uses the data to generate the conditional expectation $\mathbb{E}\left[x_k \mid I_k\right]$.
   - An actuator, which multiplies $\mathbb{E}\left[x_k \mid I_k\right]$ by the gain matrix L_k and applies the control $u_k = L_k \mathbb{E}\left[x_k \mid I_k\right]$.
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

$$\mathbb{E}_x\left[\|x-\hat{x}\|^2 \mid I\right] = \mathbb{E}\left[\|x\|^2\right] - 2\mathbb{E}[x \mid I]\hat{x} + \|\hat{x}\|^2$$

   When we take derivative with respect to $\hat{x}$ and set it equal to zero:

$$2\hat{x} - 2\mathbb{E}[x \mid I] = 0 \Rightarrow \hat{x} = \mathbb{E}[x \mid I]$$

   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.

5.2.2 Implementation aspects - Steady-state controller

- In the imperfect info case, we need to compute an estimator $\hat{x}_k = \mathbb{E}[x_k \mid I_k]$, which is indeed the one that minimizes the mean squared error $\mathbb{E}_x\left[\|x - \hat{x}\|^2 \mid I\right]$.
- 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 $\hat{x}_k$.
Kalman filter recursion: For all $k = 0, 1, \dots, N-1$, compute

$$\hat{x}_{k+1} = A_k \hat{x}_k + B_k u_k + \Sigma_{k+1|k+1} C'_{k+1} N_{k+1}^{-1} \left(z_{k+1} - C_{k+1}\left(A_k \hat{x}_k + B_k u_k\right)\right),$$

and

$$\hat{x}_0 = \mathbb{E}[x_0] + \Sigma_{0|0} C'_0 N_0^{-1} \left(z_0 - C_0 \mathbb{E}[x_0]\right)$$

where the matrices $\Sigma_{k|k}$ are precomputable and are given recursively by

$$\Sigma_{k+1|k+1} = \Sigma_{k+1|k} - \Sigma_{k+1|k} C'_{k+1} \left(C_{k+1} \Sigma_{k+1|k} C'_{k+1} + N_{k+1}\right)^{-1} C_{k+1} \Sigma_{k+1|k},$$
$$\Sigma_{k+1|k} = A_k \Sigma_{k|k} A'_k + M_k, \quad k = 0, 1, \dots, N-1$$

with

$$\Sigma_{0|0} = S - S C'_0 \left(C_0 S C'_0 + N_0\right)^{-1} C_0 S$$

In these equations, M_k, N_k, and S are the covariance matrices[13] of w_k, v_k and x_0, respectively, and we assume that w_k and v_k have zero mean; that is

$$\mathbb{E}[w_k] = \mathbb{E}[v_k] = 0$$
$$M_k = \mathbb{E}\left[w_k w'_k\right], \quad N_k = \mathbb{E}\left[v_k v'_k\right]$$
$$S = \mathbb{E}\left[\left(x_0 - \mathbb{E}[x_0]\right)\left(x_0 - \mathbb{E}[x_0]\right)'\right]$$

Moreover, we are assuming that matrices N_k are positive definite (and hence, invertible).
Stationary case

[13] Recall that for a random vector X, its covariance matrix Σ is given by $\mathbb{E}[(X - \mathbb{E}[X])(X - \mathbb{E}[X])']$. Its entry (i,j) is given by $\Sigma_{ij} = \mathrm{Cov}(X_i, X_j) = \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])]$. The covariance matrix Σ is always positive semi-definite.

- 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

  $$L = -\left(R + B'KB\right)^{-1} B'KA,$$

  and where K is the unique ≥ 0 symmetric solution of the algebraic Riccati equation

  $$K = A'\left(K - KB\left(R + B'KB\right)^{-1} B'K\right) A + Q$$

- It can also be shown in the limit as $N \to \infty$, that

  $$\hat{x}_{k+1} = (A + BL)\hat{x}_k + \bar{\Sigma}C'N^{-1}\left(z_{k+1} - C(A + BL)\hat{x}_k\right),$$

  where $\bar{\Sigma}$ is given by

  $$\bar{\Sigma} = \Sigma - \Sigma C'\left(C\Sigma C' + N\right)^{-1} C\Sigma$$

  and Σ is the unique ≥ 0 symmetric solution of the Riccati equation

  $$\Sigma = A\left(\Sigma - \Sigma C'\left(C\Sigma C' + N\right)^{-1} C\Sigma\right) A' + M$$

  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 $\mathbb{E}[x_k \mid I_k]$ may be very difficult from a computational viewpoint. So, a suboptimal solution is typically used.

A common suboptimal controller is to replace $\mathbb{E}[x_k \mid I_k]$ 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.

[132] 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: $\left[B, AB, A^2B, \ldots, A^{n-1}B\right]$ 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.

5.2.3 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:

$$\begin{aligned} &J_{N-1}\left(I_{N-1}\right) \\ &= \min_{u_{N-1}\in U_{N-1}} \mathbb{E}_{x_{N-1},w_{N-1}}\left[g_N\left(f_{N-1}\left(x_{N-1},u_{N-1},w_{N-1}\right)\right) + g_{N-1}\left(x_{N-1},u_{N-1},w_{N-1}\right) \mid I_{N-1},u_{N-1}\right], \end{aligned} \tag{5.8}$$

$$J_k\left(I_k\right) = \min_{u_k\in U_k} \mathbb{E}_{x_k,w_k,z_{k+1}}\left[g_k\left(x_k,u_k,w_k\right) + J_{k+1}\left(I_k,z_{k+1},u_k\right) \mid I_k,u_k\right]. \tag{5.9}$$

- Suppose that we can find a function $S_k\left(I_k\right)$ such that the RHS of (5.8) and (5.9) can be written in terms of some function H_k as

$$\min_{u_k\in U_k} H_k\left(S_k\left(I_k\right),u_k\right)$$

  such that

$$J_k\left(I_k\right) = \min_{u_k\in U_k} H_k\left(S_k\left(I_k\right),u_k\right)$$

- Such a function S_k is called a sufficient statistic.
- An optimal policy obtained by the preceding minimization can be written as

$$\mu_k^*\left(I_k\right) = \bar{\mu}_k^*\left(S_k\left(I_k\right)\right)$$

  where $\bar{\mu}_k^*$ is an appropriate function.
- Example of a sufficient statistic: $S_k\left(I_k\right) = I_k$.
- Another important sufficient statistic is the conditional probability distribution of the state x_k given the information vector I_k, i.e.,

$$S_k\left(I_k\right) = P_{x_k|I_k}$$

  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
$$P_{x_{k+1}|I_{k+1}} = \Phi_k \left(P_{x_k|I_k}, u_k, z_{k+1} \right) \tag{5.10}$$
for a suitable function Φ_k determined from the data of the problem. (We will verify this later)

- Claim: Suppose for now that function Φ_k in equation (5.10) 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 (5.8)), 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

$$J_{N-1}\left(I_{N-1}\right) = \min_{u_{N-1}\in U_{N-1}} H_{N-1}\left(P_{x_{N-1}|I_{N-1}}, u_{N-1}\right) = \bar{J}_{N-1}\left(P_{x_{N-1}|I_{N-1}}\right)$$

for appropriate functions H_{N-1} and $\bar{J}_{N-1}$.

IH: Assume

$$J_{k+1}\left(I_{k+1}\right) = \min_{u_{k+1}\in U_{k+1}} H_{k+1}\left(P_{x_{k+1}|I_{k+1}}, u_{k+1}\right) = \bar{J}_{k+1}\left(P_{x_{k+1}|I_{k+1}}\right), \tag{5.11}$$

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

$$J_k\left(I_k\right) = \min_{u_k\in U_k} H_k\left(P_{x_k|I_k}, u_k\right) = \bar{J}_k\left(P_{x_k|I_k}\right)$$

Using equations (5.10) and (5.11), the DP in (5.9) can be written as

$$J_k\left(I_k\right) = \min_{u_k\in U_k} \mathbb{E}_{x_k,w_k,z_{k+1}}\left[g_k\left(x_k,u_k,w_k\right) + \bar{J}_{k+1}\left(\Phi_k\left(P_{x_k|I_k}, u_k, z_{k+1}\right)\right) \mid I_k, u_k\right] \tag{5.12}$$

To solve this problem, we also need the joint distribution $P\left(x_k, w_k, z_{k+1} \mid I_k, u_k\right)$, or equivalently, given that from the primitives of the system,

$$z_{k+1} = h_{k+1}\left(x_{k+1}, u_k, v_{k+1}\right), \text{ and } x_{k+1} = f_k\left(x_k, u_k, w_k\right)$$

we need
$$P\left(x_k, w_k, h_{k+1}\left(f_k\left(x_k,u_k,w_k\right), u_k, v_{k+1}\right) \mid I_k, u_k\right).$$

This distribution can be expressed in terms of $P_{x_k|I_k}$, the given distributions

$$P\left(w_k \mid x_k, u_k\right), P\left(v_{k+1} \mid f_k\left(x_k,u_k,w_k\right), u_k, w_k\right)$$

and the system equation $x_{k+1} = f_k(x_k, u_k, w_k)$.

Therefore, the expression minimized over u_k in (5.12) can be written as a function of $P_{x_k|I_k}$ and u_k, and the DP equation (5.12) can be written as

$$J_k(I_k) = \min_{u_k \in U_k} H_k\left(P_{x_k|I_k}, u_k\right)$$

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.,

$$\mu_k(I_k) = \bar{\mu}_k\left(P_{x_k|I_k}\right), \quad k = 0, 1, \ldots, N-1$$

  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 5.2.

- This separation is the basis for various suboptimal control schemes that split the controller a priori into an estimator and an actuator.

- The controller $\bar{\mu}_k\left(P_{x_k|I_k}\right)$ 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.

5.2.4 The conditional state distribution recursion

We still need to justify the recursion

$$P_{x_{k+1}|I_{k+1}} = \Phi_k\left(P_{x_k|I_k}, u_k, z_{k+1}\right)$$

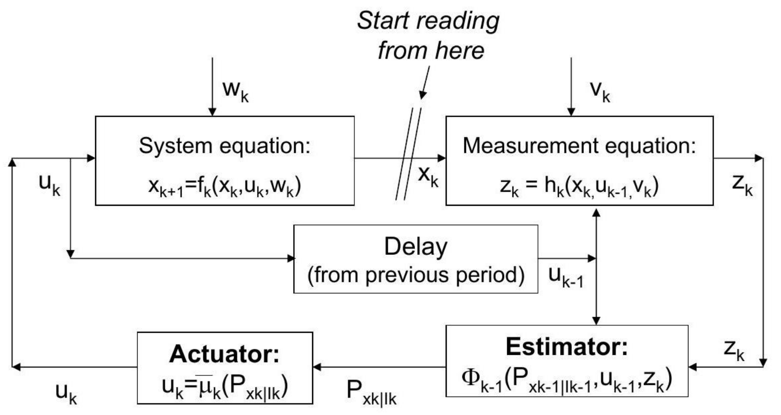


Figure 5.2: Conceptual separation of the optimal controller into an estimator and an actuator.

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 $p\left(x_{k+1} \mid I_{k+1}\right)$ is generated from $p\left(x_k \mid I_k\right), u_k$, and z_{k+1} by means of the equation:

$$\begin{aligned} p\left(x_{k+1} \mid I_{k+1}\right) &= p\left(x_{k+1} \mid I_k, u_k, z_{k+1}\right) \\ &= \frac{p\left(x_{k+1}, z_{k+1} \mid I_k, u_k\right)}{p\left(z_{k+1} \mid I_k, u_k\right)} \\ &= \frac{p\left(x_{k+1} \mid I_k, u_k\right) p\left(z_{k+1} \mid I_k, u_k, x_{k+1}\right)}{\int_{-\infty}^{\infty} p\left(x_{k+1} \mid I_k, u_k\right) p\left(z_{k+1} \mid I_k, u_k, x_{k+1}\right) \mathrm{d}x_{k+1}}. \end{aligned}$$

In this expression, all the probability densities appearing in the RHS may be expressed in terms of $p\left(x_k \mid I_k\right), u_k$, and z_{k+1}.

In particular:

- The density $p\left(x_{k+1} \mid I_k, u_k\right)$ may be expressed through $p\left(x_k \mid I_k\right), u_k$, and the system equation $x_{k+1} = f_k\left(x_k, u_k, w_k\right)$ using the given density $p\left(w_k \mid x_k, u_k\right)$ and the relation

$$p\left(w_k \mid x_k, u_k\right) = \int_{-\infty}^{\infty} p\left(x_k \mid I_k\right) p\left(w_k \mid x_k, u_k\right) \mathrm{d}x_k$$

- The density $p\left(z_{k+1} \mid I_k, u_k, x_{k+1}\right)$ is expressed through the measurement equation $z_{k+1} = h_{k+1}\left(x_{k+1}, u_k, v_{k+1}\right)$ using the densities

$$p\left(x_k \mid I_k\right), p\left(w_k \mid x_k, u_k\right), p\left(v_{k+1} \mid x_k, u_k, w_k\right)$$

Now, we give an example for the finite space set case.

Example 5.3 (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. β 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 (5.10).

5.3 Sufficient Statistics

In this section, we continue investigating the conditional state distribution as a sufficient statistic for problems with imperfect state information.

5.3.1 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.,

$$S_k\left(I_k\right) = P_{x_k \mid I_k}$$

  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

$$P_{x_{k+1}|I_{k+1}} = \Phi_k\left(P_{x_k|I_k}, u_k, z_{k+1}\right) \tag{5.13}$$

for a suitable function Φ_k determined from the data of the problem.

- We have already proven that if function Φ_k in equation (5.13) 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.,

$$\mu_k\left(I_k\right) = \bar{\mu}_k\left(P_{x_k|I_k}\right), \quad k = 0, 1, \ldots, N-1$$

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:

$$\bar{J}_{N-1}\left(P_{x_{N-1}|I_{N-1}}\right) \\ = \min_{u_{N-1}\in U_{N-1}} \mathbb{E}_{x_{N-1},w_{N-1}}\left[g_N\left(f_{N-1}\left(x_{N-1},u_{N-1},w_{N-1}\right)\right) + g_{N-1}\left(x_{N-1},u_{N-1},w_{N-1}\right) \mid I_{N-1},u_{N-1}\right] \tag{5.14}$$

and for $k = 0, 1, \ldots, N-2$,

$$\bar{J}_k\left(P_{x_k|I_k}\right) = \min_{u_k\in U_k} \mathbb{E}_{x_k,w_k,z_{k+1}}\left[g_k\left(x_k,u_k,w_k\right) + \bar{J}_{k+1}\left(\Phi_k\left(P_{x_k|I_k},u_k,z_{k+1}\right)\right) \mid I_k,u_k\right] \tag{5.15}$$

where $P_{x_k|I_k}$ plays the role of the state, and

$$P_{x_{k+1}|I_{k+1}} = \Phi_k\left(P_{x_k|I_k}, u_k, z_{k+1}\right) \tag{5.16}$$

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 5.4 (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. β 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 (5.16).
- Assume that once we decide not to search in a period, we cannot search at future times.
- The DP algorithm is

$$\bar{J}_N\left(p_N\right)=0$$

  and for $k=0,1,\ldots,N-1$,

$$\bar{J}_k\left(p_k\right)=\max \left\{0,-C+\underbrace{p_k \beta V}_{\substack{\text{reward} \\ \text{for search \& find}}}+\underbrace{\left(1-p_k \beta\right)}_{\substack{\text{prob. of} \\ \text{search \& not find}}} \bar{J}_{k+1}\left(\frac{p_k(1-\beta)}{p_k(1-\beta)+1-p_k}\right)\right\}$$

- It can be shown by induction that the functions $\bar{J}_k\left(p_k\right)$ satisfy

$$\bar{J}_k\left(p_k\right)=0, \quad \forall p_k \leq \frac{C}{\beta V}$$

Furthermore, it is optimal to search at period k if and only if

$$\underbrace{p_k \beta V}_{\substack{\text{expected reward}\\ \text{from search}}} \geq \underbrace{C.}_{\text{cost of search}}$$

5.3.2 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

$$\left(\mathbb{P}\left(x_k = 1 \mid I_k\right), \mathbb{P}\left(x_k = 2 \mid I_k\right), \ldots, \mathbb{P}\left(x_k = n \mid I_k\right)\right)$$

- 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^θ, given that the current state is $x_k = j$ and the preceding control was u_{k-1}, is denoted by $\mathbb{P}\left(z_k = z^\theta \mid u_{k-1}, x_k = j\right) \triangleq r_j\left(u_{k-1}, z^\theta\right), \theta = 1, \ldots, q$.
- The information available to the controller at stage k is

$$I_k\left(z_1, \ldots, z_k, u_0, \ldots, u_{k-1}\right)$$

- Following the observation z_k, a control u_k is applied, and a cost $g\left(x_k, u_k\right)$ 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

$$p_k = \left(p_k^1, \ldots, p_k^n\right)'$$

where

$$p_k^i = \mathbb{P}\left(x_k = i \mid I_k\right), \quad i = 1, 2, \ldots, n$$

We refer to p_k as the belief state. It evolves according to

$$p_{k+1} = \Phi_k\left(p_k, u_k, z_{k+1}\right)$$

where the function Φ_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

$$\begin{aligned}
p_{k+1}^j &= \mathbb{P}\left(x_{k+1}=j \mid I_{k+1}\right) \\
&= \mathbb{P}\left(x_{k+1}=j \mid z_0, \ldots, z_{k+1}, u_0, \ldots, u_k\right) \\
&= \frac{\mathbb{P}\left(x_{k+1}=j, z_{k+1} \mid I_k, u_k\right)}{\mathbb{P}\left(z_{k+1} \mid I_k, u_k\right)} \qquad \text{(because } \mathbb{P}(A \mid B, C)=\mathbb{P}(A, B \mid C) / \mathbb{P}(B \mid C)\text{)} \\
&= \frac{\sum_{i=1}^n \mathbb{P}\left(x_k=i \mid I_k\right) \mathbb{P}\left(x_{k+1}=j \mid x_k=i, u_k\right) \mathbb{P}\left(z_{k+1} \mid u_k, x_{k+1}=j\right)}{\sum_{s=1}^n \sum_{i=1}^n \mathbb{P}\left(x_k=i \mid I_k\right) \mathbb{P}\left(x_{k+1}=s \mid x_k=i, u_k\right) \mathbb{P}\left(z_{k+1} \mid u_k, x_{k+1}=s\right)} \\
&= \frac{\sum_{i=1}^n p_k^i p_{ij}\left(u_k\right) r_j\left(u_k, z_{k+1}\right)}{\sum_{s=1}^n \sum_{i=1}^n p_k^i p_{is}\left(u_k\right) r_s\left(u_k, z_{k+1}\right)}
\end{aligned}$$

In vector form, we have

$$p_{k+1}^j=\frac{r_j\left(u_k, z_{k+1}\right)\left[P\left(u_k\right)' p_k\right]_j}{\sum_{s=1}^n r_s\left(u_k, z_{k+1}\right)\left[P\left(u_k\right)' p_k\right]_s}, \quad j=1, \ldots, n \tag{5.17}$$

where $P\left(u_k\right)$ is the $n \times n$ transition probability matrix formed by $p_{ij}\left(u_k\right)$, and $\left[P\left(u_k\right)' p_k\right]_j$ is the j th component of vector $\left[P\left(u_k\right)' p_k\right]$.

The corresponding DP algorithm (5.14)-(5.15) has the specific form

$$\bar{J}_k\left(p_k\right)=\min_{u_k \in U}\left\{p_k' g\left(u_k\right)+\mathbb{E}_{z_{k+1}}\left[\bar{J}_{k+1}\left(\Phi\left(p_k, u_k, z_{k+1}\right)\right) \mid p_k, u_k\right]\right\}, \quad k=0, \ldots, N-1 \tag{5.18}$$

where $g\left(u_k\right)$ is the column vector with components $g\left(1, u_k\right), \ldots, g\left(n, u_k\right)$, and $p_k' g\left(u_k\right)$ is the expected stage cost.

The algorithm starts at stage N with

$$\bar{J}_N\left(p_N\right)=p_N' G$$

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 $\bar{J}_k$ in the DP algorithm are piecewise linear and concave. A consequence of this fact is that $\bar{J}_k$ 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 5.5 (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: $g(\bar{P}, C) = 2$; $g(P, C) = 0$; $g(\bar{P}, S) = 1$; $g(P, S) = 1$, or in vector form:

$$g(C) = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, g(S) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

- 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: $\mathbb{P}\left(x_0 = \bar{P}\right) = \frac{1}{3}$, $\mathbb{P}(x_0 = P) = \frac{2}{3}$
- Assume that we start with a machine in good state, i.e., $x_{-1} = P$
- System equation:

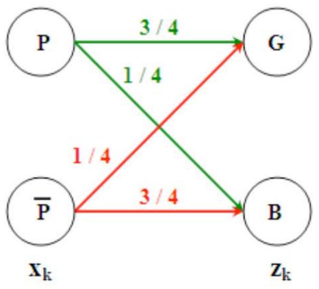
$$x_{k+1} = w_k, \quad k = 0, 1$$

  where the transition probabilities are given by the matrix form, following the aforementioned indexing of the states, transition probabilities can be expressed as

$$P(C) = \begin{pmatrix} 1 & 0 \\ 1/3 & 2/3 \end{pmatrix}; P(S) = \begin{pmatrix} 1/3 & 2/3 \\ 1/3 & 2/3 \end{pmatrix}$$

- 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: $z_k = v_k, \quad k = 0, 1; v_k \in \{B, G\}$



The inspection results can be described by the following definitions:

$$\begin{aligned}
r_1(S,G) &\triangleq \mathbb{P}\left(z_{k+1}=G \mid u_k=S, x_{k+1}=\bar{P}\right)=\frac{1}{4}=r_1(C,G),\\
r_1(S,B) &\triangleq \mathbb{P}\left(z_{k+1}=B \mid u_k=S, x_{k+1}=\bar{P}\right)=\frac{3}{4}=r_1(C,B),\\
r_2(S,G) &\triangleq \mathbb{P}\left(z_{k+1}=G \mid u_k=S, x_{k+1}=P\right)=\frac{3}{4}=r_2(C,G),\\
r_2(S,B) &\triangleq \mathbb{P}\left(z_{k+1}=B \mid u_k=S, x_{k+1}=P\right)=\frac{1}{4}=r_2(C,B).
\end{aligned}$$

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:

$$p_0^1 \triangleq \mathbb{P}\left(x_0=\bar{P} \mid I_0\right), \quad p_0^2 \triangleq \mathbb{P}\left(x_0=P \mid I_0\right)=1-p_0^1$$

Similarly, define the belief state p_1 with coordinates

$$p_1^1 \triangleq \mathbb{P}\left(x_1=\bar{P} \mid I_1\right), \quad p_1^2 \triangleq \mathbb{P}\left(x_1=P \mid I_1\right)=1-p_1^1$$

where the evolution of the beliefs is driven by the estimator

$$p_1=\Phi_0\left(p_0,u_0,z_1\right)$$

We will use equation (5.17) to compute p_1 given p_0, but first we calculate the matrix products $P\left(u_0\right)' p_0$, for $u_0 \in \{S,C\}$:

$$P(S)'p_0=\begin{pmatrix} 1/3 & 1/3 \\ 2/3 & 2/3 \end{pmatrix}\begin{pmatrix} p_0^1 \\ p_0^2 \end{pmatrix}=\begin{pmatrix} \frac{1}{3}\left(p_0^1+p_0^2\right) \\ \frac{2}{3}\left(p_0^1+p_0^2\right) \end{pmatrix}=\begin{pmatrix} \frac{1}{3} \\ \frac{2}{3} \end{pmatrix},$$

and

$$P(C)'p_0=\begin{pmatrix} 1 & 1/3 \\ 0 & 2/3 \end{pmatrix}\begin{pmatrix} p_0^1 \\ p_0^2 \end{pmatrix}=\begin{pmatrix} p_0^1+\frac{1}{3}p_0^2 \\ \frac{2}{3}p_0^2 \end{pmatrix}=\begin{pmatrix} \frac{1}{3}+\frac{2}{3}p_0^1 \\ \frac{2}{3}-\frac{2}{3}p_0^1 \end{pmatrix}.$$

Now, using equation (5.17) for state $j=1$ (i.e., for state $\bar{P}$), we get

- For $u_0=S, z_1=G$:

$$p_1^1=\frac{r_1(S,G)\left[P(S)'p_0\right]_1}{r_1(S,G)\left[P(S)'p_0\right]_1+r_2(S,G)\left[P(S)'p_0\right]_2}=\frac{\frac{1}{4}\times\frac{1}{3}}{\frac{1}{4}\times\frac{1}{3}+\frac{3}{4}\times\frac{2}{3}}=\frac{1}{7}$$

- For $u_0=S, z_1=B$:

$$p_1^1=\frac{r_1(S,B)\left[P(S)'p_0\right]_1}{r_1(S,B)\left[P(S)'p_0\right]_1+r_2(S,B)\left[P(S)'p_0\right]_2}=\frac{\frac{3}{4}\times\frac{1}{3}}{\frac{3}{4}\times\frac{1}{3}+\frac{1}{4}\times\frac{2}{3}}=\frac{3}{5}$$

- For $u_0 = C, z_1 = G$:

$$p_1^1 = \frac{r_1(C,G)\,[P(C)'p_0]_1}{r_1(C,G)\,[P(C)'p_0]_1 + r_2(C,G)\,[P(C)'p_0]_2} = \frac{\frac{1}{4}\times\left(\frac{1}{3}+\frac{2}{3}p_0^1\right)}{\frac{1}{4}\times\left(\frac{1}{3}+\frac{2}{3}p_0^1\right)+\frac{3}{4}\times\left(\frac{2}{3}-\frac{2}{3}p_0^1\right)} = \frac{1+2p_0^1}{7-4p_0^1}$$

- For $u_0 = C, z_1 = B$:

$$p_1^1 = \frac{r_1(C,B)\,[P(C)'p_0]_1}{r_1(C,B)\,[P(C)'p_0]_1 + r_2(C,B)\,[P(C)'p_0]_2} = \frac{\frac{3}{4}\times\left(\frac{1}{3}+\frac{2}{3}p_0^1\right)}{\frac{3}{4}\times\left(\frac{1}{3}+\frac{2}{3}p_0^1\right)+\frac{1}{4}\times\left(\frac{2}{3}-\frac{2}{3}p_0^1\right)} = \frac{3+6p_0^1}{5+4p_0^1}$$

In summary, we get

$$p_1^1 = \left[\Phi_0\left(p_0,u_0,z_1\right)\right]_1 = \begin{cases} \frac{1}{7} & \text{if } u_0 = S, z_1 = G \\ \frac{3}{5} & \text{if } u_0 = S, z_1 = B \\ \frac{1+2p_0^1}{7-4p_0^1} & \text{if } u_0 = C, z_1 = G \\ \frac{3+6p_0^1}{5+4p_0^1} & \text{if } u_0 = C, z_1 = B \end{cases}$$

where $p_0^2 = 1 - p_0^1$ and $p_1^2 = 1 - p_1^1$.

The DP algorithm (5.18) may be written as:

$$\bar{J}_2\left(p_2\right) = 0 \quad \text{(i.e., zero terminal cost)},$$

and

$$\bar{J}_1\left(p_1\right) = \min_{u_1\in\{S,C\}}\left\{p_1' g\left(u_1\right)\right\} = \min\left\{\underbrace{\left(p_1^1,p_1^2\right)\begin{pmatrix}1\\1\end{pmatrix}}_{u_1=S},\underbrace{\left(p_1^1,p_1^2\right)\begin{pmatrix}2\\0\end{pmatrix}}_{u_1=C}\right\}$$

$$= \min\{\underbrace{p_1^1+p_1^2}_{u_1=S},\ \underbrace{2p_1^1}_{u_1=C}\} = \min\{\underbrace{1}_{u_1=S},\ \underbrace{2p_1^1}_{u_1=C}\}.$$

This minimization yields

$$\bar{\mu}_1^*\left(p_1\right) = \begin{cases} C & \text{if } p_1^1 \le \frac{1}{2} \\ S & \text{if } p_1^1 > \frac{1}{2} \end{cases}$$

For stage $k = 0$, we have

$$\begin{aligned}\bar{J}_0\left(p_0\right) &= \min_{u_0\in\{C,S\}}\left\{p_0' g\left(u_0\right) + \mathbb{E}_{z_1}\left[\bar{J}_1\left(\Phi\left(p_0,u_0,z_1\right)\right) \mid p_0,u_0\right]\right\} \\ &= \min\{\underbrace{2p_0^1 + \mathbb{P}\left(z_1 = G \mid I_0, C\right)\bar{J}_1\left(\Phi_0\left(p_0,C,G\right)\right) + \mathbb{P}\left(z_1 = B \mid I_0, C\right)\bar{J}_1\left(\Phi_0\left(p_0,C,B\right)\right)}_{u_0=C}, \\ &\quad \underbrace{\underbrace{\left(p_0^1+p_0^2\right)}_{1} + \mathbb{P}\left(z_1 = G \mid I_0, S\right)\bar{J}_1\left(\Phi_0\left(p_0,S,G\right)\right) + \mathbb{P}\left(z_1 = B \mid I_0, S\right)\bar{J}_1\left(\Phi_0\left(p_0,S,B\right)\right)}_{u_0=S}\}\end{aligned}$$

The probabilities here may be expressed in terms of p_0 by using the expression in the denominator of (5.17); that is:

$$\begin{aligned}\mathbb{P}\left(z_{k+1} \mid I_k, u_k\right) &= \sum_{s=1}^{n}\sum_{i=1}^{n}\mathbb{P}\left(x_k = i \mid I_k\right)\mathbb{P}\left(x_{k+1} = s \mid x_k = i, u_k\right)\mathbb{P}\left(z_{k+1} \mid u_k, x_{k+1} = s\right)\\ &= \sum_{s=1}^{n}\sum_{i=1}^{n} p_k^i p_{is}\left(u_k\right) r_s\left(u_k, z_{k+1}\right)\\ &= \sum_{s=1}^{n} r_s\left(u_k, z_{k+1}\right)\left[P\left(u_k\right)' p_k\right]_s\end{aligned}$$

In our case:

$$\begin{aligned}\mathbb{P}\left(z_1 = G \mid I_0, u_0 = C\right) &= r_1(C,G)\left[P(C)'p_0\right]_1 + r_2(C,G)\left[P(C)'p_0\right]_2\\ &= \frac{1}{4}\times\left(\frac{1}{3}+\frac{2}{3}p_0^1\right)+\frac{3}{4}\times\left(\frac{2}{3}-\frac{2}{3}p_0^1\right)\\ &= \frac{7-4p_0^1}{12}\end{aligned}$$

Similarly, we obtain:

$$\mathbb{P}\left(z_1 = B \mid I_0, C\right) = \frac{5+4p_0^1}{12}, \quad \mathbb{P}\left(z_1 = G \mid I_0, S\right) = \frac{7}{12}, \quad \mathbb{P}\left(z_1 = B \mid I_0, S\right) = \frac{5}{12}$$

Using these values we have

$$\bar{J}_0\left(p_0\right) = \min\left\{\begin{aligned}&2p_0^1 + \frac{7-4p_0^1}{12}\bar{J}_1\left(\underbrace{\frac{1+2p_0^1}{7-4p_0^1}}_{p_1^1}, 1-p_1^1\right) + \frac{5+4p_0^1}{12}\bar{J}_1\left(\underbrace{\frac{3+6p_0^1}{5+4p_0^1}}_{p_1^1}, 1-p_1^1\right),\\ &1+\frac{7}{12}\bar{J}_1\left(\frac{1}{7},\frac{6}{7}\right)+\frac{5}{12}\bar{J}_1\left(\frac{3}{5},\frac{2}{5}\right)\end{aligned}\right\}.$$

By substitution of $\bar{J}_1\left(p_1\right)$ and after some algebra we obtain

$$\bar{J}_0\left(p_0\right) = \begin{cases}\frac{19}{12} & \text{if } \frac{3}{8} \le p_0^1 \le 1\\ \frac{7+32p_0^1}{12} & \text{if } 0 \le p_0^1 \le \frac{3}{8}\end{cases}$$

and an optimal control for the first stage

$$\bar{\mu}_0^*(p_0) = \begin{cases} C & \text{if } p_0^1 \le \frac{3}{8} \\ S & \text{if } p_0^1 > \frac{3}{8} \end{cases}$$

Also, we know that $\mathbb{P}(z_0 = G) = \frac{7}{12}$, and $\mathbb{P}(z_0 = B) = \frac{5}{12}$. In addition, we can establish the initial value for p_0^1 according to the value of I_0 (i.e., z_0):

$$\mathbb{P}\big(x_0 = \bar{P} \mid z_0 = G\big) = \frac{\mathbb{P}\big(x_0 = \bar{P}, z_0 = G\big)}{\mathbb{P}(z_0 = G)} = \frac{\frac{1}{3} \times \frac{1}{4}}{\frac{7}{12}} = \frac{1}{7},$$

and

$$\mathbb{P}\big(x_0 = \bar{P} \mid z_0 = B\big) = \frac{\mathbb{P}\big(x_0 = \bar{P}, z_0 = B\big)}{\mathbb{P}(z_0 = B)} = \frac{\frac{1}{3} \times \frac{3}{4}}{\frac{5}{12}} = \frac{3}{5}$$

so that the formula

$$J^* = \mathrm{E}_{z_0}\left[\bar{J}_0\left(P_{x_0|z_0}\right)\right] = \frac{7}{12}\bar{J}_0\left(\frac{1}{7}, \frac{6}{7}\right) + \frac{5}{12}\bar{J}_0\left(\frac{3}{5}, \frac{2}{5}\right) = \frac{176}{144}$$

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.