



# Dynamic Programming and Optimal Control

## Study Note

**Author:** Jinyi Liu

**Institute:** Haslam Business School, UTK

**Date:** October 28, 2022

**Bio:** A first-year Ph.D. student in Business Analytics.

*Victory won't come to us unless we go to it.*

# Contents

# Chapter 1 The Dynamic Programming Algorithm

## 1.1 Introduction

### 1.1.1 General Structure of Finite Horizon Optimal Control Problems

Our finite horizon model has two principal features: (1) a *discrete-time dynamic system*, and (2) a *cost function that is additive over time*. The system has the form

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

where

$x_k$	state variable
$u_k$	control variable
$w_k$	random parameter,

and  $f_k$  is a function that describes the system.

The cost function is additive. The total cost is

$$g_N(x_N) + \sum_{i=0}^{N-1} g_i(x_i, u_i, w_i).$$

Since  $w_k$  is random, we formulate the problem as an optimization of the *expected cost*

$$E \left\{ g_N(x_N) + \sum_{i=0}^{N-1} g_i(x_i, u_i, w_i) \right\}.$$

## 1.2 The Basic Problem

### Basic Problem

We are given a discrete-time dynamic system

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

where the state  $x_k \in S_k$ , the control  $u_k \in C_k$  and the random "disturbance"  $w_k$  is an element of a space  $D_k$ .

The control  $u_k$  is constrained to be  $u_k \in U_k(x_k) \subset C_k$  for all  $x_k \in S_k$  and  $k$ .

$w_k$  is characterized by a probability distribution  $P_k(\cdot | x_k, u_k)$  that may explicitly on  $x_k$  and  $u_k$  but not on values of prior disturbances  $w_{k-1}, \dots, w_0$ .

We consider the class of policies

$$\pi = \{\mu_0, \dots, \mu_{N-1}\}$$

, where  $\mu_k$  maps  $x_k$  into controls  $u_k = \mu_k(x_k)$  and is such that  $\mu_k(x_k) \in U_k(x_k)$  for all  $x_k \in S_k$ . Such policies will be called *admissible*.

Given  $x_0$  and admissible  $\pi$ , we have

$$x_{k+1} = f_k(x_k, \mu_k(x_k), w_k), \quad k = 0, 1, \dots, N-1 \quad (1.1)$$

Thus, for given function  $g_k$ , we have the expected cost of  $\pi$  starting at  $x_0$ :

$$J_\pi(x_0) = E \left\{ g_N(x_N) + \sum_{i=0}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right\}$$

where the expectation is taken over  $x_k$  and  $w_k$ . An optimal policy  $\pi^*$  is one such that

$$J_{\pi^*}(x_0) = \min_{\pi \in \Pi} J_\pi(x_0).$$

## The Role and Value of Information

## Encoding Risk in the Cost Function

### 1.3 The Dynamic Programming Algorithm

The DP algorithm rests on the *principle of optimality*.

#### The DP Algorithm

##### Proposition 1.3.1

For every initial state  $x_0$ , the optimal cost  $J^*(x_0)$  of the basic problem is equal to  $J_0(x_0)$ , given by the last step of the following algorithm, which proceeds backward in time from period  $N-1$  to period 0 :

$$\begin{aligned} J_N(x_N) &= g_N(x_N), \\ J_k(x_k) &= \min_{u_k \in U_k(x_k)} E \{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \} \\ k &= 0, 1, \dots, N-1, \end{aligned}$$

where the expectation is taken with respect to the probability distribution of  $w_k$ , which depends on  $x_k$  and  $u_k$ . Furthermore, if  $u_k^* = \mu_k^*(x_k)$  minimizes the right side of Eq. (1.6) for each  $x_k$  and  $k$ , the policy  $\pi^* = \{\mu_0^*, \dots, \mu_{N-1}^*\}$  is optimal.



### 1.4 State Augmentation and Other Reformulations

The general guideline in *state augmentation* is to include in the enlarged state at time  $k$  all the information that is known to the controller at time  $k$  and can be used with advantage in selecting  $u_k$ .

## Time Delays

### 1.5 Some Mathematical Issues

Well-defined random variables.

### 1.6 Dynamic Programming and Minimax Control

Consider a triplet  $(\Pi, W, J)$ , where  $\Pi$  is the set of policies under consideration,  $W$  is the set in which the uncertain quantities are known to belong, and  $J : \Pi \times W \rightarrow [-\infty, +\infty]$  is a given cost function. The objective is to

$$\min_{\pi \in \Pi} \max_{w \in W} J(\pi, w)$$

over all  $\pi \in \Pi$ .

#### Lemma 1.6.1

Let  $f : W \rightarrow X$  be a function, and  $M$  be the set of all functions  $\mu : X \rightarrow U$ , where  $W, X$ , and  $U$  are some sets. Then for any functions  $G_0 : W \rightarrow (-\infty, \infty]$  and  $G_1 : X \times U \rightarrow (-\infty, \infty]$  such that

$$\min_{u \in U} G_1(f(w), u) > -\infty, \quad \text{for all } w \in W,$$

we have

$$\min_{\mu \in M} \max_{w \in W} [G_0(w) + G_1(f(w), \mu(f(w)))] = \max_{w \in W} \left[ G_0(w) + \min_{u \in U} G_1(f(w), u) \right].$$



# Chapter 2 Deterministic Systems and the Shortest Path Problem

In this chapter we focus on deterministic problems, i.e.,  $w_k$  can take only one value. In contrast with stochastic problems, *using feedback results in no advantage in terms of cost reduction.*

## 2.1 Finite-State Systems and Shortest Paths

The DP algorithm takes the form

$$J_N(i) = a_{it}^N, \quad i \in S_N, \quad (2.1)$$

$$J_k(i) = \min_{j \in S_{k+1}} [a_{ij}^k + J_{k+1}(j)], \quad i \in S_k, \quad k = 0, 1, \dots, N-1. \quad (2.2)$$

### A Forward DP Algorithm for Shortest Path Problems

An optimal path from  $s$  to  $t$  is also an optimal path from  $t$  to  $s$  in a "reverse" shortest path problem. It is given by

$$\tilde{J}_N(j) = a_{sj}^0, \quad j \in S_1, \quad (2.3)$$

$$\tilde{J}_k(j) = \min_{i \in S_{N-k}} [a_{ij}^{N-k} + \tilde{J}_{k+1}(i)], \quad j \in S_{N-k+1}, \quad k = 1, 2, \dots, N-1. \quad (2.4)$$

The optimal cost is

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