

Macro Topics: Introduction to Matlab

Fall semester 2016

Lecture notes (November 8)

Olga Bychkova

Topics Covered Today

Discrete time dynamic programming: Finite horizon

- ▶ Problem formulation
- ▶ Examples
- ▶ The basic problem
- ▶ Principle of optimality
- ▶ DP example: Deterministic problem
- ▶ DP example: Stochastic problem
- ▶ The general DP algorithm
- ▶ State augmentation

This lecture is based on Dimitri Bertsekas' book, Ch. 1 of V. 1.

Dynamic Programming as an Optimization Methodology

Basic optimization problem

$$\min_{u \in U} g(u),$$

where

- ▶ u is the optimization/decision variable,
- ▶ $g(u)$ is the cost function, and
- ▶ U is the constraint set.

Categories of problems:

- ▶ discrete (U is finite) or continuous;
- ▶ linear (g is linear and U is polyhedral) or nonlinear;
- ▶ stochastic or deterministic:
 - ▶ In stochastic problems the cost involves a stochastic parameter w , which is averaged, i.e., it has the form

$$g(u) = \mathbb{E}_w \{ G(u, w) \},$$

where w is a random parameter.

Basic Structure of Stochastic Dynamic Programming I

Discrete-time system

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

- ▶ k : **discrete time**
- ▶ x_k : **state** (summarizes past information that is relevant for future optimization)
- ▶ u_k : **control** (decision to be selected at time k from a given set)
- ▶ w_k : **random parameter** (also called disturbance or noise depending on the context)
- ▶ N : **horizon** (or number of times control is applied)
- ▶ f_k is a function that describes the system and, in particular, the mechanism by which the state is updated

Basic Structure of Stochastic Dynamic Programming II

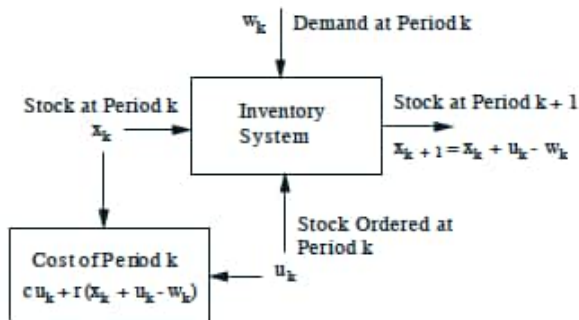
Cost function that is additive over time

$$\mathbb{E} \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right\},$$

where $g_N(x_N)$ is a terminal cost incurred at the end of the process.

The optimization is over the controls u_0, u_1, \dots, u_{N-1} .

Inventory Control Example I



- ▶ x_k — stock available at the beginning of the k th period,
- ▶ u_k — stock ordered (and immediately delivered) at the beginning of the k th period,
- ▶ w_k — demand during the k th period with given probability distribution.

Inventory Control Example II

Discrete-time system

$$x_{k+1} = f_k(x_k, u_k, w_k) = x_k + u_k - w_k$$

Cost function that is additive over time

$$\mathbb{E} \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right\} = \mathbb{E} \left\{ R(x_N) + \sum_{k=0}^{N-1} (cu_k + r(x_{k+1})) \right\},$$

- ▶ $r(x_k)$ — a cost that represents a penalty for either positive stock x_k (holding cost for excess inventory) or negative stock x_k (shortage cost for unfilled demand),
- ▶ cu_k — the purchasing cost, where c is cost per unit ordered,
- ▶ $R(x_N)$ — a terminal cost for being left with inventory x_N at the end of N periods.

Optimization over policies:

rules/functions $u_k = \mu_k(x_k)$ that map states to controls.

Additional Assumptions

The set of values that the control u_k can take depend at most on x_k and not on prior x or u .

Probability distribution of w_k does not depend on past values w_{k-1}, \dots, w_0 , but may depend on x_k and u_k .

- ▶ Otherwise past values of w or x would be useful for future optimization.

Sequence of events envisioned in period k :

- ▶ x_k occurs according to

$$x_k = f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1})$$

- ▶ u_k is selected with knowledge of x_k , i.e.,

$$u_k \in U(x_k)$$

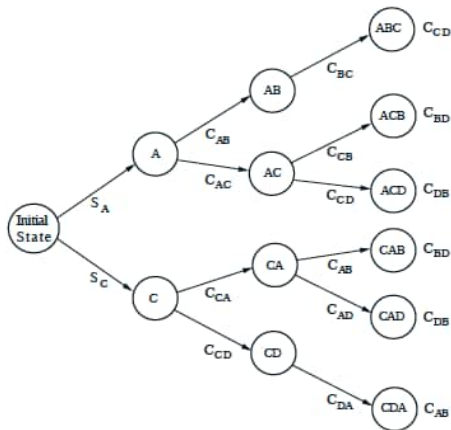
- ▶ w_k is random and generated according to a distribution

$$P_{w_k}(x_k, u_k)$$

Deterministic Finite-State Problems

Scheduling example:

- ▶ Find optimal sequence of operations A, B, C, D
- ▶ A must precede B, and C must precede D
- ▶ Given startup cost S_A and S_C , and setup transition cost C_{mn} from operation m to operation n

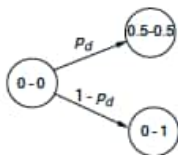


Stochastic Finite-State Problems I

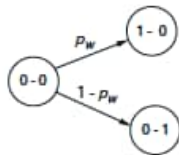
Example: Find two-game chess match strategy.

- ▶ **Timid** player draws with probability $p_d > 0$ and loses with probability $1 - p_d$.
- ▶ **Bold** player wins with probability $p_w < 1/2$ and loses with probability $1 - p_w$.

Assumption: $p_d > p_w$.

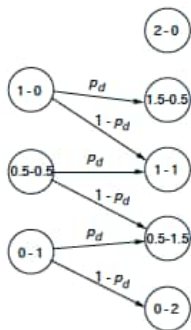


1st Game / Timid Play

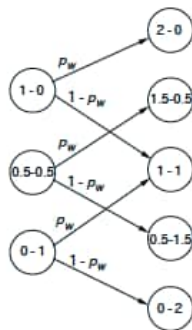


1st Game / Bold Play

Stochastic Finite-State Problems II



2nd Game / Timid Play



2nd Game / Bold Play

Stochastic Finite-State Problems III

2nd stage:

1 - 0 — **Bold:** $p_w + (1 - p_w)p_w$ vs. **Timid:** $p_d + (1 - p_d)p_w$
 $p_w + (1 - p_w)p_w < p_d + (1 - p_d)p_w \Rightarrow$ **Timid**

0.5 - 0.5 — **Bold:** p_w vs. **Timid:** $p_d p_w$
 $p_w > p_d p_w \Rightarrow$ **Bold**

0 - 1 — **Bold:** p_w^2 vs. **Timid:** 0
 $p_w^2 > 0 \Rightarrow$ **Bold**

1st stage:

Bold: $p_w(p_d + (1 - p_d)p_w) + (1 - p_w)p_w^2$ vs.

Timid: $p_d p_w + (1 - p_d)p_w^2$

$p_w(p_d + (1 - p_d)p_w) + (1 - p_w)p_w^2 > p_d p_w + (1 - p_d)p_w^2 \Rightarrow$ **Bold**

Answer: $\{ \text{Bold}, [\text{Timid}/\text{Bold}/\text{Bold}, 1 - 0/0.5 - 0.5/0 - 1] \} \Rightarrow$
 $\{ \text{Bold}, [\text{Timid}/\text{Bold}, 1 - 0/0 - 1] \}$

Basic Problem

- ▶ **System** $x_{k+1} = f_k(x_k, u_k, w_k)$, $k = 0, \dots, N - 1$
- ▶ **Control constraints** $u_k \in U(x_k)$
- ▶ **Probability distribution** $P_k(\cdot | x_k, u_k)$ of w_k
- ▶ **Policies** $\pi = \{\mu_0, \dots, \mu_{N-1}\}$, where μ_k maps states x_k into controls $u_k = \mu_k(x_k)$ and is such that $\mu_k(x_k) \in U_k(x_k)$ for all x_k
- ▶ **Expected cost** of π starting at x_0 is

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

- ▶ **Optimal cost function**

$$J^*(x_0) = \min_{\pi} J_\pi(x_0)$$

- ▶ **Optimal policy** π^* satisfies

$$J_{\pi^*}(x_0) = J^*(x_0)$$

When produced by DP, π^* is independent of x_0 .

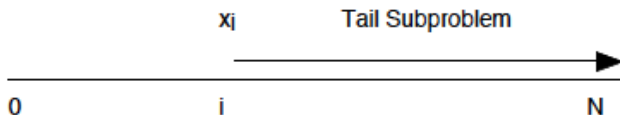
Principle of Optimality I

Let $\pi^* = \{\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*\}$ be an optimal policy.

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

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

and the “tail policy” $\{\mu_i^*, \mu_{i+1}^*, \dots, \mu_{N-1}^*\}$.



Principle of Optimality II

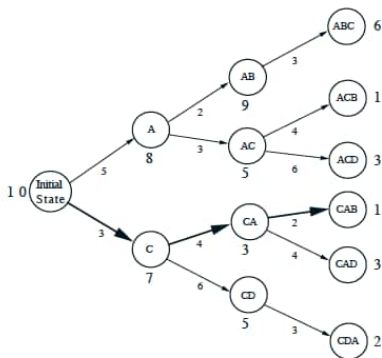
Principle of optimality: The tail policy is optimal for the tail subproblem.

DP first solves ALL tail subproblems of final stage.

At the generic step, it solves ALL tail subproblems of a given time length, using the solution of the tail subproblems of shorter time length.

Deterministic Scheduling Example

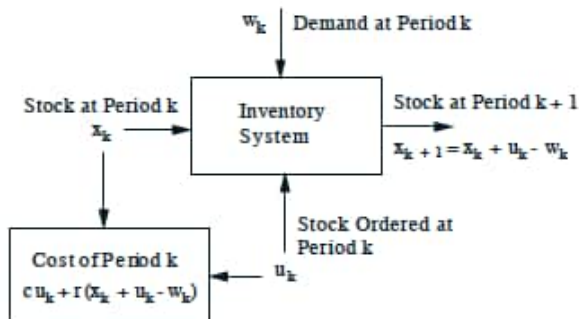
Find optimal sequence of operations A, B, C, D (A must precede B and C must precede D).



Start from the last tail subproblem and go backwards.

At each state-time pair, we record the optimal cost-to-go and the optimal decision.

Stochastic Inventory Example



Tail Subproblems of Length 1:

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

Tail Subproblems of Length $N - k$:

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

DP Algorithm I

Start with

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

and go backwards using, $\forall k = 0, 1, \dots, N - 1$,

$$J_k(x_k) = \min_{u_k \in U_k(x_k)} \mathbb{E}_{w_k} \{g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k))\}.$$

Then $J_0(x_0)$, generated at the last step, is equal to the optimal cost $J^*(x_0)$. Also, the policy

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

where $\mu_k^*(x_k)$ minimizes the right side above for each x_k and k , is optimal.

DP Algorithm II

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

Note that ALL the tail subproblems are solved in addition to the original problem, and the intensive computational requirements.

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

Proof of the Induction Step

Assume that $J_{k+1}(x_{k+1}) = J_{k+1}^*(x_{k+1})$. Then

$$\begin{aligned} J_k^*(x_k) &= \min_{(\mu_k, \pi_{k+1})} \mathbb{E}_{w_k, \dots, w_{N-1}} \left\{ g_k(x_k, \mu_k(x_k), w_k) + g_N(x_N) + \right. \\ &+ \left. \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right\} = \min_{\mu_k} \mathbb{E}_{w_k} \left\{ g_k(x_k, \mu_k(x_k), w_k) + \right. \\ &+ \left. \min_{\pi_{k+1}} \left[\mathbb{E}_{w_{k+1}, \dots, w_{N-1}} \left\{ g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right\} \right] \right\} = \\ &= \min_{\mu_k} \mathbb{E}_{w_k} \left\{ g_k(x_k, \mu_k(x_k), w_k) + J_{k+1}^*(f_k(x_k, \mu_k(x_k), w_k)) \right\} = \\ &= \min_{\mu_k} \mathbb{E}_{w_k} \left\{ g_k(x_k, \mu_k(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k(x_k), w_k)) \right\} = \\ &= \min_{u_k \in U_k(x_k)} \mathbb{E}_{w_k} \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\} = J_k(x_k). \end{aligned}$$

Linear-Quadratic Analytical Example I



System

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

where a is given scalar from the interval $(0, 1)$.

Cost

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

where r is given positive scalar.

DP Algorithm:

$$J_2(x_2) = r(x_2 - T)^2$$
$$J_1(x_1) = \min_{u_1} \left[u_1^2 + r((1 - a)x_1 + au_1 - T)^2 \right]$$

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

Linear-Quadratic Analytical Example II

$$J_1(x_1) = \min_{u_1} \left[u_1^2 + r \left((1-a)x_1 + au_1 - T \right)^2 \right]$$

$$F.O.C. : [u_1] : 2u_1 + 2ra \left((1-a)x_1 + au_1 - T \right) = 0 \Rightarrow$$

$$\Rightarrow u_1 = \mu^*(x_1) = \frac{ra(T - (1-a)x_1)}{1 + ra^2} \Rightarrow$$

$$\Rightarrow J_1(x_1) = \frac{r^2 a^2 (T - (1-a)x_1)^2}{(1 + ra^2)^2} +$$

$$+ r \left((1-a)x_1 + \frac{ra^2 (T - (1-a)x_1)}{1 + ra^2} - T \right)^2 =$$

$$= \frac{r^2 a^2 (T - (1-a)x_1)^2}{(1 + ra^2)^2} + r \left(\frac{ra^2}{1 + ra^2} - 1 \right)^2 (T - (1-a)x_1)^2 =$$

$$= \frac{r}{1 + ra^2} (T - (1-a)x_1)^2$$

Linear-Quadratic Analytical Example III

$$\begin{aligned} J_0(x_0) &= \min_{u_0} \left[u_0^2 + J_1((1-a)x_0 + au_0) \right] = \\ &= \min_{u_0} \left[u_0^2 + \frac{r}{1+ra^2} (T - (1-a)((1-a)x_0 + au_0))^2 \right] \end{aligned}$$

$$F.O.C. : [u_0] : 2u_0 - \frac{2ra(1-a)}{1+ra^2} (T - (1-a)((1-a)x_0 + au_0)) = 0$$

$$\Rightarrow u_0 = \mu^*(x_0) = \frac{ra(1-a)(T - (1-a)^2x_0)}{1+ra^2(1+(1-a)^2)} \Rightarrow$$

$$\Rightarrow J_0(x_0) = \frac{r(T - (1-a)^2x_0)^2}{1+ra^2(1+(1-a)^2)}$$

State Augmentation

When assumptions of the basic problem are violated (e.g., disturbances are correlated, cost is nonadditive, etc.), reformulate/augment the state.

Example: Time lags

$$x_{k+1} = f_k(x_k, x_{k-1}, u_k, w_k)$$

Introduce additional state variable $y_k = x_{k-1}$. New system takes the form

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} f_k(x_k, y_k, u_k, w_k) \\ x_k \end{pmatrix}$$

View $\tilde{x}_k = (x_k, y_k)$ as the new state.

DP algorithm for the reformulated problem:

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