

# Stochastic Models and Optimization: Problem Set 1

Roger Garriga Calleja, José Fernando Moreno Gutiérrez, David Rosenfeld, Katrina Walker

March 2, 2017

**Problem 1 (Machine Maintenance):** Suppose that we have a machine that is either running or is broken down. If it runs throughout one week it makes a gross profit of \$100. If it fails during the week, the gross profit is zero. If it is running and at the start of the week and we perform preventive maintenance, the probability that it will fail during the week is 0.4. If we do not perform such maintenance, the probability of failure is 0.7. However, maintenance will cost \$20. When the machine is broken down at the start of the week, it may either be repaired at a cost of \$40, in which case it will fail during the week with probability 0.4, or it may be replaced at a cost of \$90 by a new machine that is guaranteed to work properly through its first week of operation. Reformulate the problem within the DP framework, and find the optimal repair-replacement-maintenance policy that maximizes the total expected profit over four weeks, assuming a new machine at the start of the first week.

## Primitives:

State:  $x_k$  is the state of the machine at the beginning of the week  $k$ .  $x_k$  can be either 1 (running) or 0 (broken).

Control:  $u_k \in U_k$ . If  $x_k = 1$  then  $U_k = \{M, NM\}$  where  $M$  means maintenance and  $NM$  means not maintenance. If  $x_k = 0$  then  $U_k = \{F, R\}$  where  $F$  mean repair (Fix) and  $R$  means replace.

Uncertainty:  $w_k$  is the state at the end of the week, that is if it will be broken or not.  $w_k$  can be either 1 (running at the end of week  $k$ ) or 0 (broken at the end of week  $k$ ).

Dynamics: The state at the beginning of week  $k + 1$  will be the state at the end of the week  $k$ , so  $x_{k+1} = w_k$ .

Profit: The profit of week  $k$  will be the 100 if it is running at the end minus the cost of the policy  $u_k$ . So,

$$g_k(x_k, u_k, w_k) = 100w_k - \begin{cases} 0 & \text{if } u_k = NM \\ 20 & \text{if } u_k = M \\ 40 & \text{if } u_k = F \\ 90 & \text{if } u_k = R \end{cases}.$$

The terminal cost will be 0, since when the 4 weeks have finished we will not need the machine,  $g_N(x_N) = 0$ .

DP algorithm: Since the terminal profit is 0,  $J_N(x_N) = 0$ . Then, the profit-to-go of the  $N - k$  tail problem will be

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

$$j_k(x_k) = \begin{cases} \max_{u_k \in \{M, NM\}} \mathbb{E} \left[ 100w_k - \begin{cases} 0 & \text{if } u_k = NM \\ 20 & \text{if } u_k = M \end{cases} + J_{k+1}(w_k) \right] & \text{if } x_k = 1 \\ \max_{u_k \in \{M, NM\}} \mathbb{E} \left[ 100w_k - \begin{cases} 40 & \text{if } u_k = F \\ 90 & \text{if } u_k = R \end{cases} + J_{k+1}(w_k) \right] & \text{if } x_k = 0 \end{cases}$$

Solving by backward induction:  $J_4(x_4) = 0$  independent of  $x_4$ . Now, for the next profit-to-go we distinguish between the case  $x_3 = 1$  and case  $x_3 = 0$ ,

$$J_3(1) = \max_{u_3 \in \{M, NM\}} \{\mathbb{E}[100w_3|u_3 = NM], \mathbb{E}[100w_3 - 20|u_3 = M]\} = \quad (1)$$

$$= \max\{100\mathbb{P}(w_3 = 1|u_3 = NM), 100\mathbb{P}(w_3 = 1|u_3 = M) - 20\} = \quad (2)$$

$$= \max\{100 \cdot 0.3, 100 \cdot 0.6 - 20\} = \max\{30, 40\} = 40, \implies u_3 = M, \quad (3)$$

$$J_3(0) = \max_{u_3 \in \{F, R\}} \{\mathbb{E}[100w_3 - 40|u_3 = F], \mathbb{E}[100w_3 - 90|u_3 = R]\} = \quad (4)$$

$$= \max\{100\mathbb{P}(w_3 = 1|u_3 = F) - 40, 100\mathbb{P}(w_3 = 1|u_3 = R) - 90\} = \quad (5)$$

$$= \max\{100 \cdot 0.6 - 40, 100 - 90\} = \max\{20, 10\} = 20, \implies u_3 = F. \quad (6)$$

So,  $u_3(x_3 = 1) = M$  and  $u_3(x_3 = 0) = F$ . Then,

$$J_2(1) = \max_{u_2 \in \{M, NM\}} \{\mathbb{E}[100w_2 + J_3(w_2)|u_2 = NM], \mathbb{E}[100w_2 - 20 + J_3(w_2)|u_2 = M]\} = \quad (7)$$

$$= \max\{(100 + 40)\mathbb{P}(w_2 = 1|u_2 = NM) + 20\mathbb{P}(w_2 = 0|u_2 = NM), \quad (8)$$

$$(100 + 40)\mathbb{P}(w_2 = 1|u_2 = M) + 20\mathbb{P}(w_2 = 0|u_2 = M) - 20\} = \quad (9)$$

$$= \max\{(100 + 40) \cdot 0.3 + 20 \cdot 0.7, (100 + 40) \cdot 0.6 + 20 \cdot 0.4 - 20\} = \quad (10)$$

$$= \max\{56, 72\} = 72, \implies u_2 = M, \quad (11)$$

$$J_2(0) = \max_{u_2 \in \{F, R\}} \{\mathbb{E}[100w_2 - 40 + J_3(w_2)|u_2 = F], \mathbb{E}[100w_2 - 90 + J_3(w_2)|u_2 = R]\} = \quad (12)$$

$$= \max\{(100 + 40)\mathbb{P}(w_2 = 1|u_2 = F) + 20\mathbb{P}(w_2 = 0|u_2 = F) - 40, \quad (13)$$

$$(100 + 40)\mathbb{P}(w_2 = 1|u_2 = R) + 20\mathbb{P}(w_2 = 0|u_2 = R) - 90\} = \quad (14)$$

$$= \max\{(100 + 40) \cdot 0.6 + 20 \cdot 0.4 - 40, (100 + 40) - 90\} = \quad (15)$$

$$= \max\{52, 50\} = 52, \implies u_2 = F, \quad (16)$$

So,  $u_2(x_2 = 1) = M$  and  $u_2(x_2 = 0) = F$ . Then,

$$J_1(1) = \max_{u_1 \in \{M, NM\}} \{\mathbb{E}[100w_1 + J_2(w_1)|u_1 = NM], \mathbb{E}[100w_1 - 20 + J_2(w_1)|u_1 = M]\} = \quad (17)$$

$$= \max\{(100 + 72)\mathbb{P}(w_1 = 1|u_1 = NM) + 52\mathbb{P}(w_1 = 0|u_1 = NM), \quad (18)$$

$$(100 + 72)\mathbb{P}(w_1 = 1|u_1 = M) + 52\mathbb{P}(w_1 = 0|u_1 = M) - 20\} = \quad (19)$$

$$= \max\{(100 + 72) \cdot 0.3 + 52 \cdot 0.7, (100 + 72) \cdot 0.6 + 52 \cdot 0.4 - 20\} = \quad (20)$$

$$= \max\{88, 104\} = 104, \implies u_1 = M, \quad (21)$$

$$J_1(0) = \max_{u_1 \in \{F, R\}} \{\mathbb{E}[100w_1 - 40 + J_2(w_1)|u_1 = F], \mathbb{E}[100w_1 - 90 + J_2(w_1)|u_1 = R]\} = \quad (22)$$

$$= \max\{(100 + 72)\mathbb{P}(w_1 = 1|u_1 = F) + 52\mathbb{P}(w_1 = 0|u_1 = F) - 40, \quad (23)$$

$$(100 + 72)\mathbb{P}(w_1 = 1|u_1 = R) + 52\mathbb{P}(w_1 = 0|u_1 = R) - 90\} = \quad (24)$$

$$= \max\{(100 + 72) \cdot 0.6 + 52 \cdot 0.4 - 40, (100 + 72) - 90\} = \quad (25)$$

$$= \max\{84, 82\} = 84, \implies u_1 = F. \quad (26)$$

So,  $u_1(x_1 = 1) = M$  and  $u_1(x_1 = 0) = F$ . Since the machine is new at the start, during the first week it is guaranteed to work, so  $J_0(x_0) = \max \mathbb{E}[100w_0 + J_1(w_0)] = 100 + J_1(1) = 100 + 104 = 204$ .

**Problem 3 (Multiplicative cost):** In the framework of the basic problem, consider the case where the cost has the multiplicative form

$$\mathbb{E}_{\{w_k\}} \{g_N(x_N) \cdot g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) \cdots g_0(x_0, u_0, w_0)\}$$

Develop a DP-like algorithm for this problem assuming that  $g_k(x_k, u_k, w_k) > 0$ , for all  $x_k, u_k, w_k$  and  $k$ .

Primitives:

State: The state of the system at the beginning of the period  $k$  is  $x_k$ .

Control: The control or decision at period  $k$  is  $u_k$ .

Uncertainty: The uncertainty at period  $k$  is  $w_k$ .

Dynamics: The dynamics of the problem at period  $k$  is  $x_{k+1} = f_k(x_k, u_k, w_k)$ , for a certain  $f_k$ .

Cost: The cost at period  $k$  is  $f_k(x_k, u_k, w_k)$  and has a multiplicative form (so the cost from the period 0 to the period  $k \neq N$  will be  $\prod_{i=0}^k g_i(x_i, u_i, w_i)$  and  $g_N(x_N) \prod_{i=0}^{k-1} g_i(x_i, u_i, w_i)$  if  $k = N$ ).

As in the DP problem, we can take  $J_k(x_k)$  as the cost-to-go of the  $N - k$  element. Then,

$$J_N(x_N) = g(x_N).$$

And from there on considering  $U_k$  the set of the possible decisions,

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

Let's prove that this makes sense, meaning that if the minimum of the expected cost-to-go in the  $(k + 1)$ th state is  $J_{k+1}(x_{k+1})$  then the minimum of the expected cost-to-go in the  $k$ th state will be  $J_k(x_k)$  as we have defined it. To do so, we will use induction.

Let us call  $J^*(x_k)$  the real minimum of the expected cost-to-go in the  $k$ th state. It is clear that  $J_N^*(x_N) = J_N(x_N) = g_N(x_N)$  (because it does not depend on any  $w$ ). Now, assume that  $J_{k+1}^*(x_{k+1}) = J_{k+1}(x_{k+1})$ . Then, considering  $J_k^*(x_k)$ ,

$$J_k^*(x_k) = \min_{u_k, \dots, u_{N-1}} \mathbb{E}_{w_k, \dots, w_{N-1}} \left\{ g_k(x_k, u_k, w_k) \prod_{i=k+1}^{N-1} (g_i(x_i, u_i, w_i)) g_N(x_N) \right\}.$$

Since  $g_k(x_k, u_k, w_k)$  only depends on period  $k$  we can put it out of the expectation and the minimum over the subsequent variables  $(u_{k+1}, \dots, u_{N-1}$  and  $w_{k+1}, \dots, w_{N-1})$ . So we get

$$\begin{aligned} J_k^*(x_k) &= \min_{u_k} \mathbb{E}_{w_k} \left\{ g_k(x_k, u_k, w_k) \min_{u_{k+1}, \dots, u_{N-1}} \mathbb{E}_{w_{k+1}, \dots, w_{N-1}} \left\{ \prod_{i=k+1}^{N-1} (g_i(x_i, u_i, w_i)) g_N(x_N) \right\} \right\} = \\ &= \min_{u_k} \mathbb{E}_{w_k} \{g_k(x_k, u_k, w_k) J_{k+1}^*(x_{k+1})\}, \end{aligned}$$

since  $J_{k+1}(x_{k+1}) = J_{k+1}^*(x_{k+1}) = \min_{u_{k+1}, \dots, u_{N-1}} \mathbb{E}_{w_{k+1}, \dots, w_{N-1}} \left\{ \prod_{i=k+1}^{N-1} (g_i(x_i, u_i, w_i)) g_N(x_N) \right\}$  by hypothesis of induction,

$$J_k^*(x_k) = \min_{u_k} \mathbb{E}_{w_k} \{g_k(x_k, u_k, w_k) J_{k+1}(x_{k+1})\}.$$

Q.E.D.

**Problem 5 (Traveling Repairman Problem):** *A repairman must service  $N$  sites, which are located along a line and are sequentially numbered  $1, 2, \dots, N$ . The repairman starts at a given site  $s$  with  $1 < s < N$ , and is constrained to service only sites that are adjacent to the ones serviced so far, i.e. if he has already serviced sites  $i, i+1, \dots, j$ , then he may service only site  $i-1$  (assuming  $1 < i$ ) or that site  $j+1$  (assuming  $j < N$ ). There is a waiting cost of  $c_i$  for each time period that site  $i$  has remained unserved and there is a travel cost  $t_{ij}$  for servicing site  $j$  right after site  $i$ . Reformulate the problem within the DP framework.*

Primitives:

S: the origin at period  $k$

T: the destination

$\alpha_{ij}$ :  $t_{ij} + \sum_{j=k+1}^N c_j$

$d_i$  : path of minimum total cost from  $s$  to  $i$

$P_i$  : parent of node

Open: set of nodes whose labels may need correction

Upper: lowest costs from  $s$  -  $t$

Initialization:

$$d_S = \sum_{j=1}^N c_j$$

$$d_i = \inf$$

$$\forall j \neq S$$

$$\text{Open} = S$$

$$\text{Upper} = \infty$$

Algorithm:

- 1) Remove  $i \in \text{Open}$ 
  - $\forall j$  child of  $i$  execute (2)
- 2) if  $d_i + \alpha_{ij} < \min \{d_j, \text{Upper}\}$ 
  - then  $d_j = d_i + \alpha_{ij}$
  - $P_j = i$
  - If  $j \neq t$ , put  $j$  in Open
  - If  $j = t$ ,  $\text{Upper} = d_i + \alpha_{ij}$
- 3) If  $\text{Open} \neq \emptyset$  terminate, else to go to (1)