BGSE MSc in Data Science - Stochastic Models and Optimization
Instructor: Mihalis G. Markakis

Problem Set 1 - Solutions

Problem 1 (Machine Maintenance - [B05] Exercise 1.3)

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.

Solution: Except for the first week, there are only two permissible states: R (= running) and B (0 broken down). For $x_k = R$, there are only two permissible control values: m (= maintenance) and n (= no maintenance). For $x_k = B$, there are two permissible control values: r (= repair) and l (= replace). At the beginning of the first week, the machine is in state N (= new machine), guaranteed to run properly during its first week of operation. Number the weeks from 0 to 3. Using the data of the problem we have:

Week 3: For $x_3 = R$:

$$\begin{cases} \text{if } u_3 = m, & \text{expected profit} = 0.6 \cdot 100 - 20 = 40 \\ \text{if } u_3 = n, & \text{expected profit} = 0.3 \cdot 100 = 30. \end{cases}$$

Thus, $J_3(R) = \max\{40, 30\} = 40$ and $\mu_3^*(R) = m$. Now, for $x_3 = B$:

$$\begin{cases} \text{if } u_3 = r, & \text{expected profit} = 0.6 \cdot 100 - 40 = 20 \\ \text{if } u_3 = l, & \text{expected profit} = 100 - 90 = 10. \end{cases}$$

Thus, $J_3(B) = \max\{20, 10\} = 20$ and $\mu_3^*(B) = r$.

Week 2: For $x_2 = R$:

$$\begin{cases} \text{if } u_2 = m, & \text{expected profit} = 0.6\big(100 + J_3(R)\big) + 0.4 J_3(B) - 20 = 72 \\ \text{if } u_2 = n, & \text{expected profit} = 0.3\big(100 + J_3(R)\big) + 0.7 J_3(B) = 56. \end{cases}$$

Thus, $J_2(R) = \max\{72, 56\} = 72$ and $\mu_2^*(R) = m$. Now, for $x_2 = B$:

$$\begin{cases} \text{if } u_2 = r, & \text{expected profit} = 0.6\big(100 + J_3(R)\big) + 0.4J_3(B) - 40 = 52 \\ \text{if } u_2 = l, & \text{expected profit} = 100 + J_3(R) - 90 = 50. \end{cases}$$

Thus, $J_2(B) = \max\{52, 50\} = 52$ and $\mu_2^*(B) = r$.

Week 1: For $x_1 = R$:

$$\begin{cases} \text{if } u_1 = m, & \text{expected profit} = 0.6\big(100 + J_2(R)\big) + 0.4J_2(B) - 20 = 104 \\ \text{if } u_1 = n, & \text{expected profit} = 0.3\big(100 + J_2(R)\big) + 0.7J_2(B) = 88. \end{cases}$$

Thus, $J_1(R) = \max\{104, 88\} = 104$ and $\mu_1^*(R) = m$. Now, for $x_1 = B$:

$$\begin{cases} \text{if } u_1 = r, & \text{expected profit} = 0.6\big(100 + J_2(R)\big) + 0.4J_2(B) - 40 = 84 \\ \text{if } u_1 = l, & \text{expected profit} = 100 + J_2(R) - 90 = 82. \end{cases}$$

Thus, $J_1(B) = \max\{84, 82\} = 84$ and $\mu_1^*(B) = r$.

Week 0: We start with a new machine, which is guaranteed to run properly through its first week of operation. Thus, $J_0(N) = 100 + J_1(R) = 204$.

Summarizing, the profit-maximizing policy is always to maintain a running machine and always repair a broken one. The corresponding expected profit is \$204.

Problem 2 (Discounted Cost - [B05] Exercise 1.6)

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

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

where $\alpha \in (0,1)$ is a discount factor. Develop a DP-like algorithm for this problem.

Solution: The fact that $J_N(x_N) = \alpha^N g_N(x_N)$ motivates us to define the function:

$$V_k(x_k) = \frac{J_k(x_k)}{\alpha^k}.$$

Then, the first equation of the DP algorithm is the discounted final cost, i.e., $V_N(x_N) = g_N(x_N)$, while the second one takes the form:

$$\begin{aligned}\frac{J_k(x_k)}{\alpha^k} = V_k(x_k) &= \min_{u_k \in U(x_k)} \mathbb{E}_{w_k}\Big\{\frac{\alpha^k g_k(x_k, u_k, w_k)}{\alpha^k} + \frac{J_{k+1}\big(f(x_k, u_k, w_k)\big)}{\alpha^k}\Big\} \\ &= \min_{u_k \in U(x_k)} \mathbb{E}_{w_k}\Big\{g_k(x_k, u_k, w_k) + \alpha\frac{J_{k+1}\big(f(x_k, u_k, w_k)\big)}{\alpha^{k+1}}\Big\},\end{aligned}$$

which allows us to substitute for $V_{k+1}(\cdot)$. By doing so, we have that

$$V_k(x_k) = \min_{u_k \in U(x_k)} \mathbb{E}_{w_k}\Big\{g_k(x_k, u_k, w_k) + \alpha V_{k+1}\big(f(x_k, u_k, w_k)\big)\Big\}.$$

Problem 3 (**Multiplicative Cost - [B05] Exercise 1.9**)

In the framework of the basic problem, consider the case where the cost has the multiplicative form

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

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.

Solution: Since the distribution of w_k depends only on x_k and u_k, and, additionally, the cost functions $g_k(\cdot)$ are strictly positive, we can write:

$$\begin{aligned}J^*(x_0) = &\min_{\mu_0(\cdot)}\Big[\mathbb{E}_{w_0}\Big\{g_0(x_0,\mu_0(x_0),w_0)\cdot\min_{\mu_1(\cdot)}\Big[\mathbb{E}_{w_1}\Big\{g_1(x_1,\mu_1(x_1),w_1)\dots\\ &\dots\min_{\mu_{N-1}(\cdot)}\Big[\mathbb{E}_{w_{N-1}}\Big\{g_{N-1}(x_{N-1},\mu_{N-1}(x_{N-1}),w_{N-1})\cdot g_N(x_N)\Big\}\Big]\dots\Big\}\Big]\Big\}\Big].\end{aligned}$$

This motivates a DP-like algorithm with the following form:

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

and

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

Problem 4 (Knapsack Problem - [B05] Exercise 1.10)

Assume that we have a vessel whose maximum weight capacity is z and whose cargo is to consist of different quantities of N different items. Let v_i denote the value of the i^{th} type of item, w_i the weight of the i^{th} type of item, and x_i the number of items of type i that are loaded in the vessel. The problem is to find the most valuable cargo, i.e., to maximize $\sum_{i=1}^{N} x_i v_i$ subject to the constraints $\sum_{i=1}^{N} x_i w_i \leq z$ and $x_i \in \mathbb{N}$. Reformulate the problem within the DP framework.

Solution: Let us load the N items sequentially, and identify stages with items. We define as z_i, the remaining capacity before loading item i, to be the state of the system, and x_i to be the control at stage i, i.e., the number of type-i items that are loaded. Then, we have the following formulation:

Dynamics: $z_{i+1} = z_i - x_i w_i$

Constraints: $X_i(z_i) = \{x \in \mathbb{Z}_+ : x \leq z_i / w_i\}$

Reward: $g_i(z_i, x_i) = x_i v_i$.

We can use the DP recursion:

$$J_{N+1}(z_{N+1}) = 0,$$

and

$$J_i(z_i) = \max_{x_i \in X_i} \Big\{ x_i v_i + J_{i+1}(z_i - x_i w_i) \Big\},$$

to compute $J_1(z)$, which is the most valuable cargo of the vessel.

Problem 5 (Traveling Repairman Problem - [B05] Exercise 1.24)

A repairman must service N sites, which are located along a line and are sequentially numbered $1, 2, \ldots, N$. 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 $i, i+1, \ldots, j$, then he may service only site $i-1$ (assuming $1 < i$) or site $j+1$ (assuming $j < N$). There is a waiting cost of c_i for each time period that site i has remained unserviced and there is a travel cost t_{ij} for servicing site j right after site i. Reformulate the problem within the DP framework.

Solution: Let x_k denote the location of the traveling repairman at the beginning of stage k, with $x_1 = s$, and u_k the location he will visit during stage k. Let also $X_k = \{x_1, \ldots, x_k\}$ be the set of serviced stations at the beginning of stage k. Together x_k and X_k constitute the state of the system. Then, we have the following formulation:

Dynamics: $x_{k+1} = u_k,\ X_{k+1} = X_k \cup \{u_k\}$

Constraints:

$$U_k(X_k) = \begin{cases} \big\{ \max\{X_k\} + 1, \max\{X_k\} - 1 \big\}, & \text{if } 1, N \notin X_k \\ \big\{ \max\{X_k\} + 1 \big\}, & \text{if } 1 \in X_k, N \notin X_k \\ \big\{ \max\{X_k\} - 1 \big\}, & \text{if } N \in X_k, 1 \notin X_k \end{cases}$$

Cost: $g_k(x_k, X_k, u_k) = \sum_{i \notin X_k} c_i + t_{x_k u_k}$.

We can use the DP recursion:

$$J_N(x_{N+1}, X_{N+1}) = 0,$$

and

$$J_k(x_k, X_k) = \min_{u_k \in U_k(X_k)} \Big\{ \sum_{i \notin X_k} c_i + t_{x_k u_k} + J_{k+1}\big(u_k, X_k \cup \{u_k\}\big) \Big\},$$

to compute $J_1(s)$, which is the minimum cost service schedule.