1 Problem Description

You are designing a control software for a robot moving packages that arive via automated lines. The number of packages that arrive in a period $t = 1, \ldots, N$ is a random variable and the number of packages arriving in each period is independent of the number of those arriving in other periods. Suppose that p_i denotes the probability that i packages arrive in a period ($i = 0, 1, \ldots, m$), where $\sum_{i=0}^{m} p_i = 1$. The expected number of arriving packages in a period is positive.

At the end of each period, the robot may move packages to the shipping dock. If the robot takes packages for moving, then it takes all packages that have arrived so far and have not yet been moved. The maximal number of packages that the robot can move in one load is $M > 1$. The cost of moving is $K > 0$ and the moving time is negligible. The cost incurred in each period is $w > 0$ for each package that has not been moved. All packages, if any, must be moved to the shipping dock in period N.

The problem is to determine an optimal moving policy that minimizes the combined expected costs of moving and waiting over N periods.

2 Technique Details and Theoretical Analysis

For simplicity, we say the cost of moving K is independent of the number of packages robot can move in one load. And the maximal number of packages capacity M is much larger than m so that there is no need to worry that robot cannot even possibly handle the first-period arriving packages.

If the robot will automatically load on all the new arrival packages, then this is a typical inventory problem. However here M is not infinite, so we made a prior policy: if current number of packages meets $M - m$, which means it's possible that next-period arriving packages may let the number of packages exceed the maximal capability of the robot, then we move the robot.

2.1 Formulate a dynamic programming problem for determining the best moving policy. Clearly describe the state and control spaces, the cost functions, transition probabilities, and write the dynamic programming equations.

- State Space:

  $$\mathcal{X} = \{0, 1, \ldots, M\}$$

  represents the total number of packages waiting for moving at the end of each period before making decision.

- Control Space:

  $$\mathcal{U} = \{0 : \text{not move}, 1 : \text{move}\}.$$

- Feasible Mapping:

  $$U(x) = \begin{cases} \{0\}, & x = 0, \\ \{0, 1\}, & 0 < x \leq M - m, \\ \{1\}, & x > M - m. \end{cases}$$

- Cost Function:

$$c(x,u) = \begin{cases} wx, & u = 0, \\ K, & u = 1. \end{cases}$$

- Transition Probability:

$$\begin{aligned} P(X_{t+1} = x_t + i | X_t = x, u = 0) &= \begin{cases} p_i, & 0 \le x \le M - m, \\ 0, & \text{o.w..} \end{cases} \\ P(X_{t+1} = i | X_t = x, u = 1) &= p_i. \end{aligned}$$

  All other transition probabilities are zero.

- State Equation:

$$x_{t+1} = \begin{cases} x_t + i, u = 0, \\ i, u = 1. \end{cases}$$

- Dynamic Equation:

$$\begin{aligned} v_t(x_t) &= \begin{cases} \min_{u \in \mathcal{U}} \{c(x_t, u) + \sum_{j=0}^{m} p_j v_{t+1}(x_{t+1})\}, & 0 \le x_t \le M - m, \\ K + \sum_{j=0}^{m} p_j v_{t+1}(j)\}, & x_t > M - m. \end{cases} \qquad t = 1, \ldots, N-1, \\ v_N(x_N) &= \begin{cases} K & 0 < x_N \le M, \\ 0 & x_N = 0. \end{cases} \end{aligned} \tag{2.1}$$

Since all packages must be shipped at the end of period N, that is to say at the end of period N, we have no choice but to move the robot if there comes new packages. And if no packages comes during period N and there is nothing left on robot, then we'll cost nothing. That's how we set up (2.1).

2.2 Show that an optimal moving policy for the problem above has the following threshold decision rule in period t: denoting the number of packages waiting to be moved at the end of period t by s, move in period t if $s > s_t^*$ and wait (do not move) if $s \le s_t^*$.

Proof. Since $K, w > 0$, we are sure that $c(x,u)$ is non-decreasing.

In accordance with (2.1), we can verify that $v_N(\cdot)$ is also non-decreasing. Denote

$$\begin{aligned} h(s) &= c(s,0) + \sum_{j=0}^{m} p_j v_{t+1}(j+s), \\ g(s) &= c(s,1) + \sum_{j=0}^{m} p_j v_{t+1}(j). \end{aligned}$$

Then, it is plain that

$$v_t(x) = \begin{cases} \min\{h(x), g(x)\}, & 0 \le x \le M - m, \\ g(x), & x > M - m. \end{cases}$$

Assume that $v_{t+1}(\cdot)$ is non-decreasing. Obviously both $h(\cdot)$ and $g(\cdot)$ are non-decreasing because they are sum of non-decreasing functions. Then for any $0 \leq x < y \leq M - m$, we can obtain

$$h(x) \leq h(y), \quad g(x) \leq g(y).$$

and furthermore

$$v_t(x) = \min\{h(x), g(x)\} \leq \min\{h(y), g(y)\} = v_t(y).$$

For $M - n < x < y$, then

$$v_t(x) = g(x) = g(y) = v_t(y).$$

And it's also easy to verify that for any $0 \leq x \leq M - m < y$, we have

$$v_t(x) = \min\{h(x), g(x)\} = \begin{cases} h(x), & h(x) \leq g(x), \\ g(x), & h(x) > g(x), \end{cases} \leq g(y) = v_t(y).$$

That implies $v_t(\cdot)$ is also non-decreasing. Therefore we can conclude $v_t(\cdot)$ is non-decreasing inductively for any $t = 1, \ldots, N$.

It's plain that $s + j \geq j$. That is to say

$$\sum_{j=0}^{m} p_j v_{t+1}(j+s) \geq \sum_{j=0}^{m} p_j v_{t+1}(j).$$

Then a sufficient condition on $h(s) \geq g(s)$ will be

$$ws \geq K.$$

It can be derived

$$s \geq \frac{K}{w},$$

Therefore, we can obtain $s_t^* = \frac{K}{w}$.

□

2.3 Consider $N = \infty$ and formulate a discounted infinite-horizon dynamic programming problem for determining the best moving policy and write the dynamic programming equations with discount factor $\alpha = 0.9$.

The formulation of discounted infinite-horizon dynamic programming problem is the same like the finite-horizon problem we formulated in section 2.1 except for the dynamic programming equation.

In infinite-horizon problem, we think about the optimal value function $v^*(\cdot)$ instead of $v_t(\cdot)$.

$$v^*(x) = \begin{cases} \min_{u \in \mathcal{U}}\{c(x,u) + \sum_{j=0}^{m} p_j v_{t+1}((1-u)x + j)\}, & 0 \leq x \leq M - m, \\ K + \sum_{j=0}^{m} p_j v_{t+1}(j)\}, & x > M - m. \end{cases} \tag{2.2}$$

2.4 What can you say about the structure of the optimal policy for the infinite-horizon problem?

The optimal policy is stationary. That is to say the optimal policy at a state s is the same action at all times. And also by the similar way in (b), the optimal policy should be a threshold policy.