1 Introduction

2 Q2. Inventory control with forecasts

(a): Single period

Primitives D: Demand, which is a random variable

$D = \begin{cases} P_L & \text{with probability q} \\ P_S & \text{with probability 1 - q} \end{cases}$

p: Sale price per unit

c: Ordering cost per unit

s: Salvage value per unit

$b = p - c$: Backorder cost per unit

$h = c - s$: Inventory holding cost per unit

Q: Inventory decision.

The profit for this problem can be written as:

$\pi(Q) = E[p.min\{D, Q\} - c.Q + s(Q - D)^+]$

$\pi(Q) = pE[D] - (p - c)E[(D - Q)^+] + (c - s)E[(Q - D)^+]$

$\pi(Q) = pE[D] - bE[(D - Q)^+] - hE[(Q - D)^+]$

We then expand the formula to account for the expectation of demand D which takes value P_L with probability q:

$$\pi(Q) = pE[D]-b[q.E_{P_L}[(D-Q)^+]+(1-q).E_{P_S}[(D-Q)^+]]-h[qE_{P_L}[(Q-D)^+]+(1-q)E_{P_S}[(Q-D)^+]]$$

We take the derivative of the profit with respect to quantity in order to minimise, and set it equal to zero (with P_L and P_S the probability distributions in the large and small demand cases respectively):

$$\pi'(Q^*) = -b[qP_L(D > Q^*) + (1 - q)P_S(D > Q^*)] + h[qP_L(Q^* > D) + (1 - q)P_S(Q^* > D) = 0$$

$$-bq(1 - P_L(D \leqslant Q^*) - b(1 - q)(1 - P_S(D \leqslant Q^*)) + hqP_L(D \leqslant Q^*) + h(1 - q)P_S(D \leqslant Q^*) = 0$$

We can then obtain that the optimal decision Q^* satisfies:

$$qP_L(D \leqslant Q^*) + (1 - q)P_S(D \leqslant Q^*) = \frac{b}{b + h}$$

(b) Multi-period problem

Primitives

x_k: Inventory level at period k

u_k: Decision variable over the number of units to order.

w_k: Demand at period k.

$y_{k+1} = \xi$ where ξ takes the value "large demand" with probability q:

$\xi_k = \begin{cases} L & \text{with probability q} \\ S & \text{with probability 1 - q} \end{cases}$

c: ordering cost per unit

p: Sale price per unit

s: Salvage value

$b = p - c$: Backorder cost per unit
$h = c - s$: Inventory holding cost per unit

Dynamics:
$x_{k+1} = x_k + u_k - w_k$
$y_{k+1} = \xi_k$
$g_N(x_N) = -s(x_N)$
$g_k(x_k, u_k, w_k) = cu_k - pw_k + bmax\{w_k - x_k - u_k, 0\} + hmax\{x_k + u_k - w_k\}$

DP algorithm: $J_N(x_N) = -s(x_N)$
$J_k(x_k) = \min_{u_k \geqslant 0} E_{w_k,\xi_k}[g_k(x_k, u_k, w_k) + J_{k+1}(x_k, u_k, w_k)]$

We can introduce the variable $z_k = x_k + u_k$, and rewrite our DP equation as follows:

$$J_k(x_k) = \min_{z_k \geqslant x_k} [G_k(z_k)] - cx_k$$

where $G_k(z_k) = cz_k + bE[max\{0, w_k - z_k\}] + hE[max\{0, z_k - w_k\}] + E[J_{k+1}(z_k - w_k)]$

Once again we can expand this expression to take into account the two different values of ξ for large and small demand:
$G_k(z_k) = cz_k + bqE[max\{0, w_k - z_k\}|\xi = L] + b(1-q)E[max\{0, w_k - z_k\}|\xi = S] + hqE[max\{0, z_k - w_k\}|\xi = L] + h(1-q)E[max\{0, z_k - w_k\}|\xi = S] + E[J_{k+1}(z_k - w_k)]$

$G_k(z_k)$ is a convex function for all k, and there will be a solution S_k which minimises $G(z)$ such that the optimal ordering policy becomes:
$u_k^* = \mu_k^* = S_k - x_k$ if $S_k > x_k$, and 0 otherwise.

3 Q3. Inventory Pooling

Primitives
D = demands
Q = quantity ordered
P = price
h = inventory costs = c-s
b = backholding costs = p-c

First we will show that $Q_p^* = \sqrt{n}Q^* + \mu(n - \sqrt{n})$

$$G(Q) = hE[(Q-D)]^{+}bE[(D-Q)]^{+}$$
$$G'(Q^*) = hP(D \leqslant Q^*) - b(1 - P[(D \leqslant Q^*)]) = 0$$
$$P(D \leqslant Q^*) = \frac{b}{h+b}$$

Now, for the pooling we would obtain the same:

$$P(\sum_{L=1}^{n} D_i \leqslant Q_p^*) = \frac{b}{b+R}$$

Since this $\sum_{i=1}^{n} D_i = \sqrt{n}D_i + \mu(n - \sqrt{n})$,

$$P(\sqrt{n}D_1 + \mu(n - \sqrt{n}) \leqslant Q_p^*) = \frac{b}{b+R} \Longleftrightarrow P(D_1 \leqslant \frac{1}{\sqrt{n}}(Q_p^* - \mu(n - \sqrt{n}))) = \frac{b}{b+h}$$

which implies that $\frac{1}{\sqrt{n}}(Q_p^* - \mu(n - \sqrt{n}))) = Q^*$. Thus, $Q_P^* = \sqrt{n}Q^* + \mu(n - \sqrt{n})$. Next, we will apply the hint to prove the desired result:

$$\begin{aligned} nG(Q^*) &= n[hE[(Q^* - D)^+] + bE[(D - Q^*)^+]] \\ G(Q_p^*) &= [hE[(Q_p^* - \sum_{i=1}^{n} D_i)^+] + bE[(\sum_{i=1}^{n} D_i - Q_p^*)^+]] \end{aligned}$$

Since $\sum_{i=1}^{n} D_i = \sqrt{n}D_i + \mu(n - \sqrt{n})$ and $Q_P^* = \sqrt{n}Q^* + \mu(n - \sqrt{n})$, we can show that:

$$\begin{aligned} G(Q_P^*) &= [hE[(\sqrt{n}Q^* + \mu(n - \sqrt{n}) - \sqrt{n}D_i - \mu(n - \sqrt{n}))^+] \\ &+bE[\sqrt{n}D_i + \mu(n - \sqrt{n}) - \sqrt{n}Q^* + \mu(n - \sqrt{n})^+]] \\ &= \sqrt{n}hE[(Q^* - D_i)^+] + \sqrt{n}\ bE[(D_i - Q^*)^+] = \frac{nG(Q^*)}{\sqrt{n}} \end{aligned}$$

Q.E.D.

Problem 4 (An Investment Problem): An investor has the opportunity to make N sequential investments: at time k he may invest any amount $u_k \geqslant 0$ that does not exceed his current wealth x_k (does not exceed his current wealth, x_0, plus his gain or minus his loss thus far). He wins his investment back and as much more with probability p, where $\frac{1}{2} < p < 1$, and he loses his investment with probability $(1 - p)$. Find the optimal investment strategy.

Primitives:
x_k: wealth at the beginning of period k.
u_k: amount invested.
w_k: outcome (return).
$w_k = \begin{cases} 2u_k & \text{w.p. } p \\ 0 & \text{w.p. } 1-p \end{cases}$
where $\frac{1}{2} < p < 1$.

Constrains:
$u_k \geqslant 0$,
$u_k \leqslant x_k$.

Dynamics:
$x_{k+1} = x_k - u_k + w_k$,

Cost:

$g_N(x_N) = x_N$
$g_k(x_k, u_k, w_k) = 0.$

DP algorithm:
To simplify th $J_N(x_N) = \log(x_N)$.
$J_k(x_k) = \max_{u_k \in \mathcal{U}_k} \mathbb{E}_{w_k} \left[J_{k+1}(x_k - u_k + w_k)\right]$.
We will first prove that $J_k(x_k) = A_k + \ln(x_k)$ by induction.

$$J_{N-1}(x_{N-1}) = \max_{u_{n-1}} \mathbb{E}_{w_{N-1}} \left[\log(x_{N-1} - u_{N-1} + w_{N-1})\right] = \tag{1}$$

$$= \max_{u_{N-1}} \mathbb{E}_{w_{N-1}} [\log(x_{N-1} - u_{N-1} + w_{N-1})] = \tag{2}$$

$$= \max_{u_{N-1}} \left[p \log(x_{N-1} + u_{N-1}) + (1-p)\log(x_{N-1} - u_{N-1})\right]. \tag{3}$$

Now, we derive and apply FOC to find the maximum,

$$\frac{p}{x_{N-1} + u_{N-1}} - \frac{1-p}{x_{N-1} - u_{N-1}} = 0 \tag{4}$$

$$px_{N-1} - pu_{N-1} = x_{N-1} + u_{N-1} - px_{N-1} - pu_{N-1}, \tag{5}$$

solving we obtain $u_{N-1} = (2p-1)x_{N-1}$. Plugging into the expression of $J_{N-1}(x_{N-1})$

$$J_{N-1}(x_{N-1}) = \tag{6}$$