EE266 Homework 5 Solutions

1. A refined inventory model. In this problem we consider an inventory model that is more refined than the one you’ve seen in the lectures. The amount of inventory at time $t$ is denoted by $q_t \in \{0, 1, \ldots, C\}$, where $C > 0$ is the maximum capacity. New stock ordered at time $t$ is $u_t \in \{0, 1, \ldots, C - q_t\}$. This new stock is added to the inventory instantaneously. The demand that arrives after the new inventory, before the next time period, is $d_t \in \{0, 1, \ldots, D\}$. The dynamics are

$$q_{t+1} = (q_t + u_t - d_t)_{+}.$$ 

The unmet demand is $(q_t + u_t - d_t)_- = \max(-q_t - u_t + d_t, 0)$. The demands $d_0, d_1, \ldots$ are IID.

We now describe the stage cost, which does not depend on $t$, as a sum of terms.

The **ordering cost** is

$$g_{\text{order}}(u) = \begin{cases} 0 & u = 0 \\ p_{\text{fixed}} + p_{\text{whole}} u & 1 \leq u \leq u_{\text{disc}} \\ p_{\text{fixed}} + p_{\text{whole}} u_{\text{disc}} + p_{\text{disc}} (u - u_{\text{disc}}) & u > u_{\text{disc}} \end{cases}$$

where $p_{\text{fixed}}$ is the fixed ordering price, $p_{\text{whole}}$ is the wholesale price for ordering between one and $u_{\text{disc}}$ units, and $p_{\text{disc}} < p_{\text{whole}}$ is the discount price for any amount ordered above $u_{\text{disc}}$.

The **storage cost** is $g_{\text{store}}(q) = s_{\text{lin}} q + s_{\text{quad}} q^2$, where $s_{\text{lin}}$ and $s_{\text{quad}}$ are positive.

The **negative revenue** is $g_{\text{rev}}(q, u, d) = -p_{\text{rev}} \min(q + u, d)$, where $p_{\text{rev}} > 0$ is the retail price.

The **cost for unmet demand** is $g_{\text{unmet}}(q, u, d) = p_{\text{unmet}} (q + u - d)_-$, where $p_{\text{unmet}} > 0$.

The terminal cost is a **salvage cost**, which is $g_{\text{sal}}(q) = -p_{\text{sal}} q$ where $p_{\text{sal}} > 0$ is the salvage price.

We now consider a specific problem with

$$T = 50, \quad C = 20, \quad D = 4, \quad q_0 = 10,$$

and demand distribution

$$\text{Prob}(d_t = 0, 1, \ldots, 4) = (0.2, 0.25, 0.25, 0.2, 0.1).$$

The cost function parameters are

$$p_{\text{fixed}} = 4, \quad p_{\text{whole}} = 2, \quad p_{\text{disc}} = 1.6, \quad u_{\text{disc}} = 6, \quad s_{\text{lin}} = 0.1, \quad s_{\text{quad}} = 0.05, \quad p_{\text{rev}} = 3, \quad p_{\text{unmet}} = 3, \quad p_{\text{sal}} = 1.5$$

(a) Solve the MDP and report $J^*$.  

1
(b) Plot the optimal policy for several interesting values of $t$, and describe what you see. Does the optimal policy converge as $t$ goes to zero? If so, give the steady state optimal policy.

(c) Plot $E \, g_{\text{order}}(u_t)$, $E \, g_{\text{store}}(q_t)$, $E \, g_{\text{rev}}(q_t, u_t, d_t)$, $E \, g_{\text{unmet}}(q_t, u_t, d_t)$, and $E \, g_{\text{sal}}(q_t)$, versus $t$, all on the same plot.

Solution:

(a) We solve the MDP using value iteration, which results in $J^* = V_0(10) = 14.39$. We notice that the value function converges in shape at around $t = 40$ (with an offset of around 0.62 per time period), and the policy converges at around $t = 47$.

(b) The policy converges at time $t = 47$ to that of ordering 7 units when the inventory is empty, and 5 units when we only have 1 unit left in the inventory. We have plotted the optimal policy for $t = 49$, $t = 48$, $t = 47$ and $t = 0$ below.

(c) We formed the closed loop Markov chain under the optimal policy $\mu^*_t$, and evaluated the expected cost using distribution propagation. The plot is shown below.
2. **Exiting the market with incomplete fulfillment.** You have $S$ identical stocks that you can sell in one of $T$ periods. The price fluctuates randomly; we model this as the prices being IID from a known distribution. At each period, you are told the price, and then decide whether to sell one stock or wait. However, the market is not necessarily liquid, so if you decide to sell a stock, there is some chance that no buyer is willing to buy your stock. This random event is modeled as a Bernoulli random variable with probability 0.6 that no buyer is willing to buy your stock, and a probability of 0.4 that you sell it. Note that you can decide to sell a maximum of one stock per period. Your goal is to maximize the expected revenue from the sales.

(a) Model this problem as a Markov decision problem that you can use to find the optimal selling policy: that is, give the set of states $X$, the set of actions $U$, and the set of disturbances $W$. Give the dynamics function $f$, and reward function $g$. There is no terminal reward. What is the information pattern of the policy?

(b) The stock prices are independent random variables following a discretized log-normal distribution with the log-prices having mean 0 and standard deviation 0.2. In particular, the prices take 15 values, ranging from 0.6 to 2.0 in increments of 0.1, and the probability mass function of $p$ is proportional to the probability density function of the stated log-normal distribution. Compute the optimal policy and the optimal expected revenue for $T = 50$ periods and $S = 10$. Plot value functions at times $t = 0, 45, 48, 49, 50$, and policies at times $t = 0, 20, 40, 45$. What do you observe?

(c) Let us define a threshold policy in which you decide to sell only when the stock’s price is greater than the expected value of the prices. With the same parameters as in question (b), compute the expected revenue of this threshold policy. What
can you notice in comparison to question (b)?

(d) For the optimal policy and the threshold policy described in (c), compute the probability that we have unsold stocks after the final time $T$.

(e) Find a policy for which the probability of having unsold stock after the final time $T$ is less than 0.005. Compute the expected revenue of this policy, and compare it to the expected revenues of the two previous policies you calculated in this question.

Solution.

(a) The set of states $\mathcal{X}$ for this Markov decision problem represents the number of stocks we can own at each time step $t$. So $\mathcal{X} = \{0, 1, 2, \ldots, S - 1, S\}$ and $x_t \in \mathcal{X}$ for $t = 0, \ldots, T$. Then, the stock’s owner can only take two types of actions: the owner can decide to sell one stock, i.e. $u_t = 1$, or he/she can decide to wait i.e. $u_t = 0$. Therefore $\mathcal{U} = \{0, 1\}$. The information pattern of the policy is a ”split w” pattern where we measure $x$ and part of $w$ before determining the chosen action $u$. Thus $\mathcal{W} = \mathcal{W}^1 \times \mathcal{W}^2$ where $\mathcal{W}^1$ contains all the possible prices that a stock can take (we know this information before taking action $u$), and $\mathcal{W}^2 = \{0, 1\}$. Indeed, if $w^2_t = 0$ then no buyer is willing to buy the stock, and if $w^2_t = 1$, a buyer is willing to buy a stock at the current price $w^1_t$. This $w^2$ value is only known after the seller decides to sell or wait i.e. after the policy is chosen.

Thus, the dynamics $f$ of this problem are:

$$x_{t+1} = f_t(x_t, u_t, w^2_t) = \begin{cases} x_t - w^2_t & x_t > 0, u_t = 1, \\ x_t & \text{otherwise}. \end{cases}$$

And the reward function $g$ for $t = 0, \ldots, T - 1$ is

$$g_t(x_t, u_t, w^1_t, w^2_t) = \begin{cases} w^1_t & x_t > 0, u_t = 1, w^2_t = 1, \\ -\infty & x_t = 0, u_t = 1, \\ 0 & \text{otherwise}. \end{cases}$$

Finally, $g_T = 0$ as there is no terminal reward.

(b) The optimal expected revenue for $T = 50$ and $S = 10$ is 11.36.

The plots of the value function at time $t = 0, 45, 48, 49, 50$ are showed below.
The plots of the policies at time $t = 0, 20, 40, 45$ are showed below. Green means that the seller attempts to sell, and red means that he holds his stock.
We observe that the closer we get to $T$, the more the value function converges to a single value for every positive stock remaining to sell. Indeed, this seems logical as there is no terminal reward at time $T$ so the seller has no interest in having any stock left at the terminal time $T$. Similarly, we observe the same trend looking at the policy plots. In particular, we see that for a fixed number of stocks, the price threshold at which the policy is to sell decreases as the time $t$ increases. Along the same line, for a fixed price, we observe that the quantity threshold at which the policy is to sell decreases as the time $t$ increases.

(c) Implementing the described threshold policy with the same parameters as in question (b), the expected revenue is 9.39 which is less than the expected revenue found in question (b). Indeed, it is logical to find such a result, as the policy computed in question (b) is optimal whereas the policy used in this question follows a heuristic.

(d) The probability that the seller has unsold stocks after the final time $T$ is 0.083 with the optimal policy computed in question (b). This unsold stock probability is 0.722 for the threshold policy computed in question (c).

(e) Implementing another threshold policy for which the seller will always decide to sell one stock if the price $w_1^t$ is greater than 0.7 (second lowest price), the probability of having unsold stocks at the final time $T$ is 0.0027 < 0.005. The expected revenue of this new threshold policy is 10.46 which is between the expected revenues found in questions (b) and (c). It is always logical that this new policy will have a lower expected revenue than the one computed in question (b) because the policies implemented in question (b) are optimal. However, at first glance, it can be surprising that this risk-averse policy generates more revenue than the previous threshold policy, but looking at the relatively high probability of unsold stocks of 0.722 found in question (c) compared to 0.0027 in question (d), it makes sense that minimizing his/her risk of possessing unsold stocks at the final $T$, increases his/her expected revenue.

The matlab code is provided below.

```matlab
function [pol, v] = value2(f, g, gfinal, w1dist, w2dist, T);
% Dynamic programming using split w information pattern

[n,m,p2] = size(f);
p1 = size(w1dist,2);
pol = zeros(n,p1,T);
v = zeros(n,T+1);

v(:,T+1) = gfinal;

for t=T-1:-1:0
    Q = zeros(n,m,p1);
    for x=1:n
        for u=1:m
            ... (remaining code)
        end
    end
end
```
for w1=1:p1
    for w2=1:p2
        Q(x,u,w1) = Q(x,u,w1) + w2dist(w2)*g(x,u,w1,w2) + w2dist(w2)*v(x,t+2);
    end
end
end

 [~, min_ind] = min(Q, [], 2);
min_ind = reshape(min_ind, n, p1);
pol(:,:,t+1) = min_ind;
z = zeros(n, p1, T);
for x=1:n
    for w1=1:p1
        u = pol(x, w1, t+1);
        for w2=1:p2
            z(x, w1, t+1) = z(x, w1, t+1) + w2dist(w2)*g(x, u, w1, w2) + w2dist(w2)*v(x, t+2);
        end
        v(x, t+1) = v(x, t+1) + w1dist(w1)*z(x, w1, t+1);
    end
end
end

function [pol_threshold_mean, v_threshold_mean] = naive(f, g, gfinal, w1val, w1dist, T);
% threshold_mean policy

 [n, m, p2] = size(f);
p1 = size(w1dist, 2);
pol_threshold_mean = zeros(n, p1, T);
v_threshold_mean = zeros(n, T+1);
lim = w1dist*(w1val'); % expected value of the prices
threshold = 0.7; % used for question (e)

v_threshold_mean(:, T+1) = gfinal;
for t=T-1:-1:0
    for x=1:n
        for w1=1:p1
            if w1val(w1) > lim && x>1
                pol_threshold_mean(x, w1, t+1)=2;
            else
                pol_threshold_mean(x, w1, t+1)=1;
            end
        end
    end
end
z = zeros(n,p1,T);
for x=1:n
    for w1=1:p1
        u = pol_threshold_mean(x,w1,t+1);
        for w2=1:p2
            z(x,w1,t+1) = z(x,w1,t+1) + w2dist(w2)*g(x,u,w1,w2) + w2dist(w2)*v_threshold_mean(f(x,u,w2),t+2);
        end
    v_threshold_mean(x,t+1) = v_threshold_mean(x,t+1) + w1dist(w1)*z(x,w1,t+1);
    end
end
end

clear all; close all; clc;

randn(’seed’,0);
sigma = 0.2;
Phi = @(x) 0.5 + 0.5 * erf(x/sqrt(2));
T = 10;
N = 1e5;

% optimal value recursion
V1 = [zeros(T-1,1) ; exp(sigma^2/2)];
V1_MC = [zeros(T-1,1) ; exp(sigma^2/2)];
for t = (T-1):(-1):1
    V1(t) = V1(t+1) * Phi(log(V1(t+1))/sigma) ...
        + exp(sigma^2/2) * (1 - Phi(log(V1(t+1))/sigma - sigma));
    V1_MC(t) = mean(max(exp(sigma * randn(N,1)) , V1_MC(t+1)));
end
V1(1)
V1_MC(1)

% simple policy
Rsimple = exp(sigma^2/2)
Rsimple_MC = mean(exp(sigma * randn(N,1)))

% prescient policy
Rprescient = quad(@a 1 - Phi(log(a)/sigma).^T , 0 , 100)
Rprescient_MC = mean(max(exp(sigma * randn(T,N))))

% plot the value function
figure();
plot(1:T , V1 , ’k’);
grid on;
xlabel(’t’);
ylabel('V_{t}^{*}(1)');
print(gcf, '-depsc', 'when_to_sell_value.eps');

close all;
clc;

S=10;  %maximum number of stocks
T=50;
x0=11;  %number of stocks owned at time 0

%w1
w1val = (0.6:.1:2);  %prices
mu=0;
sigma=0.2;
w1dist = lognpdf(w1val,mu,sigma);
w1dist = w1dist/sum(w1dist);  %price distribution
p1=size(w1dist,2);  %number of possible prices

%w2
w2val = [0 1];
w2dist = [0.6 0.4];
p2 = size(w2dist,2);

n=S+1;
m=2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%dynamics
f = zeros(n,m,p2);
for x=1:n
    for u=1:m
        for w2=1:p2
            xa = x - 1;
            ua = u - 1;
            w2a = w2 - 1;
            if ua == 1 && xa > 0
                xa_next = xa - w2a;
            else
                xa_next = xa;
            end
            f(x,u,w2) = xa_next + 1;
        end
    end
end
% costs
g = zeros(n,m,p1,p2);
for x=1:n
    for u=1:m
        for w1=1:p1
            for w2=1:p2
                xa = x - 1;
                ua = u - 1;
                w2a = w2 - 1;
                if ua == 1 && xa > 0 && w2a == 1
                    g(x,u,w1,w2) = -w1val(w1);
                elseif ua == 1 && xa == 0
                    g(x,u,w1,w2) = inf;
                else
                    g(x,u,w1,w2) = 0;
                end
            end
        end
    end
end

% final cost
gfinal = zeros(n,1);

% distribution propagation optimal policy
d = zeros(T+1, n);
d(1, x0) = 1;
for t=0:T-1
d(t+2,:) = d(t+1,:)*P(:,:,t+1);
end

% probability of unsold stock in optimal policy
p_unsold_stock = sum(d(end,2:end))

% distribution propagation threshold_mean policy
d_threshold_mean = zeros(T+1, n);
d_threshold_mean(1, x0) = 1;
for t=0:T-1
d_threshold_mean(t+2,:) = d_threshold_mean(t+1,:)*P_threshold_mean(:,:,t+1);
end

% probability of unsold stock in threshold_mean policy
p_unsold_stock_threshold_mean = sum(d_threshold_mean(end,2:end))

% plots value function with optimal policy
count = 0;
for t=[0,45,48,49,50]
count = count + 1;
figure();
plot([0:n-1], -v(:,t+1),'.-');
s=sprintf('value function at t=%d',t);
title(s); ylabel('value'); xlabel('number of stocks');
end

%plots optimal policy
figure();
count = 0;
for t=[0,20,40,45]
count = count + 1;
subplot(2,2,count);
polplot=pol(:,:,t+1);
imagesc(polplot);
s=sprintf('policy at t=%d',t);
title(s);
xlabel('price'); ylabel('number of stocks');
end

% % plots value function with threshold_mean policy
% % for t=[0,45,48,49,50]
% % s=sprintf('value at t=%d',t);
% % figure('name',s);
% % clf;
3. **Appliance scheduling with fluctuating real-time prices.** An appliance has $C$ cycles, $c = 1, \ldots, C$, that must be run, in order, in $T \geq C$ time periods, $t = 0, \ldots, T - 1$. A schedule consists of a sequence $0 \leq t_1 < \cdots < t_C \leq T - 1$, where $t_c$ is the time period in which cycle $c$ is run. Each cycle $c$ uses a (known) amount of energy $e_c > 0$, $c = 1, \ldots, C$, and, in each period $t$, there is an energy price $p_t$. The total energy cost is then $J = \sum_{c=1}^{C} e_c p_{t_c}$. We assume that the prices are independent log-normal random variables, with known means, $\bar{p}_t$, and variances, $\sigma_t^2$, $t = 0, \ldots, T - 1$. You can think of $\bar{p}_t$ as the predicted energy price (say, from historical data), and $p_t$ as the actual realized real-time energy price.

The following questions pertain to the specific problem instance defined in `appliance_sched_data.m`.

(a) **Minimum mean cost schedule.** Find the schedule that minimizes $\mathbb{E} J$. Give the optimal value of $\mathbb{E} J$, and show a histogram of $J$ (using Monte Carlo simulation). Here you do not know the real-time prices; you only know their distributions.

(b) **Optimal policy with real-time prices.** Now suppose that right before each time period $t$, you are told the real-time price $p_t$, and then you can choose whether or not to run the next cycle in time period $t$. (If you have already run all cycles, there is nothing you can do.) Find the optimal policy, $\mu^*$. Find the optimal value of $\mathbb{E} J$, and compare it to the value found in part (a). Give a histogram of $J$.

You may use Monte Carlo (or simple numerical integration) to evaluate any integrals that appear in your calculations. For simulations, the following facts will be helpful: If...
$z \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2)$, then $w = \exp z$ is log-normal with mean $\mu$ and variance $\sigma^2$ given by

$$
\mu = e^{\tilde{\mu} + \tilde{\sigma}^2/2}, \quad \sigma^2 = \left( e^{\tilde{\sigma}^2} - 1 \right) e^{2\tilde{\mu} + \tilde{\sigma}^2}.
$$

We can solve these equations for

$$
\tilde{\mu} = \log \left( \frac{\mu^2}{\sqrt{\mu^2 + \sigma^2}} \right), \quad \tilde{\sigma}^2 = \log(1 + \sigma^2/\mu^2).
$$

**Solution.**

(a) We use state variable $x_t \in \{0, \ldots, C\}$, where $x_t$ is the number of cycles run prior to time period $t$, so we start with $x_0 = 0$ and we require $x_T = C$. The action $u_t \in \{0, 1\}$ indicates whether or not we run a cycle at time $t$; we require $u_t = 0$ when $x_t = C$. The state transition function is then $x_{t+1} = x_t + u_t$.

The stage cost is

$$
g_t(x_t, u_t) = \begin{cases} 
e x_{t+1} \tilde{p}_t & u_t = 1, \; x_t \neq C \\
\infty & u_t = 1, \; x_t = C \\
0 & \text{otherwise}, \end{cases}
$$

for $t = 0, \ldots, T - 1$, and the terminal cost is $g_T(x_T, u_T) = 0$ if $x_T = C$ and $\infty$ otherwise. This is a deterministic finite-state control problem.

The dynamic programming iteration has the form

$$
V_t(x_t) = \min \left( g_t(x_t, 0) + V_{t+1}(x_t), \; g_t(x_t, 1) + V_{t+1}(x_t + 1) \right),
$$

which we can write as

$$
V_t(x_t) = \begin{cases} 
\min \left( V_{t+1}(x_t), \; e_{x_{t+1}} \tilde{p}_t + V_{t+1}(x_t + 1) \right) & x_t \neq C \\
V_{t+1}(x_t) & x_t = C.
\end{cases}
$$

The dynamic programming recursion is initialized with $V_T(x) = g_T(x)$.

For the given problem instance, we obtain an optimal value of $E J = 96.4$, which is achieved by the schedule

$$(1, 2, 3, 4, 5, 6, 7, 8, 23, 24).$$

A histogram of $J$ under this policy is shown below.

(b) Here we have a stochastic control problem, in which we know the disturbance before determining the action. The stage cost is identical to the cost in part (a), with the mean price $\tilde{p}_t$ replaced with the real-time price $p_t$. The DP iteration has the form

$$
V_t(x_t) = \begin{cases} 
E \min \left( V_{t+1}(x_t), \; e_{x_{t+1}} p_t + V_{t+1}(x_t + 1) \right) & x_t \neq C, \\
V_{t+1}(x_t) & x_t = C,
\end{cases}
$$
where the expectation is taken with respect to \( p_t \). The optimal policy has the form
\[
\mu^*_t(x_t, p_t) = \begin{cases} 
1 & e_{x_t+1}p_t + V_{t+1}(x_t + 1) \leq V_{t+1}(x_t) \\
0 & \text{otherwise}
\end{cases}
\]
This has a very nice interpretation: It says we should run the next cycle if the energy price is cheaper than
\[
p_t \leq \frac{V_{t+1}(x_t + 1) - V_{t+1}(x_t)}{e_{x_t+1}}.
\]
We can evaluate the expectation in the iteration for \( V_t \) analytically, using the formula for the CDF of a log-normal variable, or (more simply) by Monte Carlo simulation. (Our code does the latter.)
We obtain an optimal value of \( EJ = 67.1 \), which is less than the average cost found in part (a).
The following code computes the optimal policy and optimal cost.

clear all; close all; clc;

appliance_sched_data;
rand('state',1);
randn('state',1);

N_sim = 10000; % number of simulations for histograms
Np_sim = 10000; % Monte Carlo simulations to approx EV

% c = 1 indicates no cycles have been completed; cycles indices off by 1.
% u = 1 indicates cycle not completed; u = 2 indicates cycle completed

%% part a
V = [zeros(C+1,T) [Inf*ones(C,1); 0]];
mu = NaN*ones(C+1, T);

% stage cost
G = Inf*ones(C+1, 2,T);
for t = 1:T
    G(:,1,t) = 0;
    G(1:C,2,t) = e_c'*p_mu(t);
    G(end,2,t) = 0; % state C has no cost
end

% DP
for t = T:-1:1
    [V(:,t), mu(:,t)] = min([G(:,1,t) + V(:,t+1), ...
                              G(:,2,t) + [V(2:end,t+1); V(end,t+1)]],[],2);
end

% unroll the policy
u = zeros(1,T);
x = zeros(1,T+1);
x(1) = 1; J_a = V(1,1);
for t = 1:T
    u(t) = mu(x(t),t);
    x(t+1) = x(t) + u(t) - 1;
end

% find times when u = 1, i.e., optimal control sequence
u = find(u == 2)

% Expect(J)
Eja = p_mu(u)*e_c
% realizations (for histogram)
ln_var = log(1+p_var(u)./p_mu(u).^2); % parameters for lognorm dist
ln_mu = log(p_mu(u).^2./sqrt(p_mu(u).^2 + p_var(u)));

J_sim_a = zeros(1,N_sim);
for i = 1:N_sim
    J_sim_a(i) = exp(ln_mu + sqrt(ln_var).*randn(1, C))*e_c;
end

% part b
V = [zeros(C+1,T) [Inf*ones(C,1); 0]]; mu = NaN*ones(C+1, T);

% stage cost
G = Inf*ones(C+1, 2 ,T);
for t = 1:T
    G(:,1,t) = 0;
    G(1:C,2,t) = e_c*p_mu(t);
    G(end,2,t) = 0; % state C has no cost
end

% compute EV
for t = T:-1:1

% generate Np_sim random prices
ln_var = log(1+p_var(t)./p_mu(t).^2); % parameters for lognorm dist
ln_mu = log(p_mu(t).^2./sqrt(p_mu(t).^2 + p_var(t)));
p_t = exp(ln_mu + sqrt(ln_var).*randn(Np_sim,1));
for c = 1:C
    V(c,t) = mean(min(V(c,t+1), ...
p_t*e_c(c) + repmat(V(c+1,t+1),Np_sim,1)));)
end

% realizations (for histogram)
J_sim_b = zeros(1,N_sim);
for i = 1:N_sim
    J = zeros(1,T);
    x = zeros(1,T);
    x(1) = 1;
    for t = 1:T
        % observe random price

16
\[\ln \text{var} = \log(1 + \text{var}(t)/\text{mu}(t)^2)\] % parameters for lognorm dist
\[\ln \text{mu} = \log(\text{mu}(t)^2/\sqrt{\text{mu}(t)^2 + \text{var}(t)})\]
\[\text{p}_t = \exp(\ln \text{mu} + \sqrt{\ln \text{var}} \times \text{randn})\]
if \(x(t) == \text{C+1}\)
  \(x(t+1) = x(t)\);
elseif \(e_c(x(t)) \times p_t < V(x(t), t+1) - V(x(t)+1, t+1)\)
  \(x(t+1) = x(t) + 1\);
  \(J(t) = e_c(x(t)) \times p_t\);
else
  \(x(t+1) = x(t)\);
end
end

% Expect(E) for simulation
\(J\_\text{sim}_b(i) = \text{sum}(J)\);
end
\(EJb = \text{mean}(J\_\text{sim}_b)\)

%% plot
\(\text{xmin} = \text{min}(\text{min}([J\_\text{sim}_a];%, J\_\text{sim}_b)));\)
\(\text{xmax} = \text{max}(\text{max}([J\_\text{sim}_a]);%, J\_\text{sim}_b)));\)
\(\text{ymax} = 2500;\)

% histogram of price from (a)
% subplot(2,1,1)
\(\text{hist}(J\_\text{sim}_a, \text{xmin}:5:\text{xmax})\)
\(\text{axis}([\text{xmin} \text{xmax} 0 \text{ymax}]);\)
\(\text{title('Minimum mean cost')};\)
\(\text{xlabel('J')};\)
\(\text{set(gca, 'fontsize', 12);}\)
% histogram of prices from (b)
% subplot(2,1,2)
\(\text{hist}(J\_\text{sim}_b, \text{xmin}:5:\text{xmax})\)
\(\text{axis}([\text{xmin} \text{xmax} 0 \text{ymax}]);\)
\(\text{title('Real-time prices')};\)
\(\text{xlabel('J')};\)
\(\text{set(gca, 'fontsize', 12);}\)
% print('-depsc', 'appliance_sched_hist.eps')

% price statistics (not requested in problem statement)
figure;
hold on
\(\text{plot}(\text{p}_\text{mu}, 'k');\)
plot(p_mu + sqrt(p_var), 'b--')
plot(p_mu - sqrt(p_var), 'b--')
%errorbar(1:96, p_mu, sqrt(p_var))
hold off
legend('average price', '\pm one standard deviation', 'Location', 'SouthEast')
%set(gca, 'FontSize', 16)
print('-depsc', 'appliance_sched_prices.eps')