Student ID: [redacted] Student name: [redacted]

INDE2000 Assignment

[redacted]

Question 1. There are three sub-questions. Total marks are 15.

1.1 Solve the unconstraint optimisation problem below to determine the optimal value, the optimal set:

$$\min_{\mathbf{x},} \frac{1}{2}\left(x_1^2 - x_2^2\right) + 2x_1 - x_2 \,. \qquad \text{(5 marks)}$$

Solution:

$$\text{since } \nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \end{bmatrix} = \begin{bmatrix} x_1 + 2 \\ -x_2 - 1 \end{bmatrix} = 0$$

$$\text{critical point } (x_1^*, x_2^*) \text{ is } (-2, -1)$$

$$\text{then, we can compute the Hessian of } f(x_1, x_2)$$

$$H(x_1, x_2) = \nabla^2 f(x_1, x_2) = \begin{bmatrix} \frac{\partial^2 f}{\partial {x_1}^2}, & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2}, & \frac{\partial^2 f}{\partial {x_2}^2} \end{bmatrix} = \begin{bmatrix} 1, 0 \\ 0, -1 \end{bmatrix}$$

$$\text{since, } x^{*^T} H x^* = (-2, -1) \begin{bmatrix} 1, 0 \\ 0, -1 \end{bmatrix} \begin{pmatrix} -2 \\ -1 \end{pmatrix} = 3 > 0$$

$$(-2, -1) \text{ is a local minimum of } f(x).$$

So, the minimal value is 1.5.

1.2 Solve the constraint optimisation problem below to determine the optimal value, the optimal set:

$$\min_{\mathbf{x},} 2x_1^2 + x_2^2 \text{ subject to } x_1 x_2 = 1. \qquad \text{(5 marks)}$$

Solution:

$$\text{Since: } L(x_1, x_2, \lambda) = 2x_1^2 + x_2^2 - \lambda(x_1 x_2 - 1)$$

$$\frac{\partial L}{\partial x_1} = 4x_1 - x_2\lambda$$

$$\frac{\partial L}{\partial x_2} = 2x_2 - x_1\lambda$$

$$\frac{\partial L}{\partial \lambda} = 1 - x_1 x_2$$

$$\text{Then, we can find the values of } x_1 \text{ and } x_2$$

$$4x_1 - x_2\lambda = 0$$

$$2x_2 - x_1\lambda = 0$$

$$1 - x_1 x_2 = 0$$

$$\text{Thus, we can get the outcomes}$$

$$x_1 = -2^{-\frac{1}{4}}, x_2 = -2^{\frac{1}{4}}, \lambda = 2^{\frac{3}{2}}$$

$$x_1 = 2^{-\frac{1}{4}}, x_2 = 2^{\frac{1}{4}}, \lambda = 2^{\frac{3}{2}}$$

Student ID: [redacted] Student name: [redacted]

When $x_1 = 2^{-\frac{1}{4}}$, $x_2 = 2^{\frac{1}{4}}$, the function have the minimum value, is $2\sqrt{2}$.

1.3 Write python code to verify the solutions in 1.1 and 1.2. (5 marks)

(i)

```
from scipy import optimize
import cvxopt
import matplotlib.pyplot as plt
import numpy as np
import sympy
sympy.init_printing()
x1, x2 = sympy.symbols("x_1, x_2")
f_sym = 1/2*((x1**2)-(x2**2))+(2*x1)-x2
fprime_sym = [f_sym.diff(x_) for x_ in (x1, x2)]
sympy.Matrix(fprime_sym)
```

$$\begin{bmatrix} 1.0x_1 + 2 \\ -1.0x_2 - 1 \end{bmatrix}$$

```
fhess_sym = [[f_sym.diff(x1_, x2_) for x1_ in (x1, x2)] for x2_ in (x1, x2)]
sympy.Matrix(fhess_sym)
```

$$\begin{bmatrix} 1.0 & 0 \\ 0 & -1.0 \end{bmatrix}$$

```
cri_point = np.array([-2,1])
Sol= cri_point * fhess_sym * cri_point.transpose()
Sol
```

(ii)

```
x = x1, x2, l = sympy.symbols("x_1, x_2, lambda")
f = 2*(x1**2) + (x2**2)
g = (x1*x2)-1
L = f - l * g
grad_L = [sympy.diff(L, x_) for x_ in x]
```

sols = sympy.solve(grad_L)

Sols

$$\left[\left\{\lambda : -2\sqrt{2},\ x_1 : -\frac{2^{\frac{3}{4}} i}{2},\ x_2 : \sqrt[4]{2} i\right\},\ \left\{\lambda : -2\sqrt{2},\ x_1 : \frac{2^{\frac{3}{4}} i}{2},\ x_2 : -\sqrt[4]{2} i\right\},\ \left\{\lambda : 2\sqrt{2},\ x_1 : -\frac{2^{\frac{3}{4}}}{2},\ x_2 : -\sqrt[4]{2}\right\},\ \left\{\lambda : 2\sqrt{2},\ x_1 : \frac{2^{\frac{3}{4}}}{2},\ x_2 : \sqrt[4]{2}\right\}\right]$$

g.subs(sols[0])

$$0$$

f.subs(sols[0])

$$-2\sqrt{2}$$

*source:Topic 2: Supply Chain Optimisation workshop--Lab2

Question 2 (15 marks)

SUNOIL, a manufacturer, has five manufacturing plants worldwide. Their locations and capacities are shown in Table 1 along with the fixed cost of producing 1 million units at each facility. The major markets are North America, South America, Europe, Asia, and Africa. Demand at each market, and production/transportation costs from each plant to each market in U.S. dollars are shown in the table below. Management must come up with a production plan for next year. How much should each plant produce, and which markets should each plant supply?

	Production/transportation cost per million units					Fixed Cost($)	Low capacity (million units)	Fixed cost($)	High capacity (million units)
Suppy/Market	N.America	S.America	Europe	Asia	Africa				
N.America	99	77	81	92	105	4,700	10	5,900	25
S.America	100	88	105	100	110	4,600	10	5,700	25
Europe	105	100	77	120	100	4,500	15	5,500	20
Asia	120	130	90	66	60	4,400	15	5,300	20
Africa	150	100	99	110	55	4,300	15	5,200	20
Demand million units	15	10	15	20	10				

a) Formulate an LP to minimize total costs. (10 marks)

b) Set up and solve the problem SciPy.Optimize in Python. What is the optimal solution? Explain the rationale for the answer. (5 marks)

Solution:

a) Decisions: y_i 1 if plant is ipen; 0 otherwise

x_{ij} $quantity\ shipped\ from\ plant\ i\ to\ market\ j$

Student ID: [redacted] Student name: [redacted]

$$\text{Min} \sum_{i=1}^{5} f_i * y_i + \sum_{i=1}^{5} \sum_{i=1}^{5} c_{i,j} * x_{i,j}$$

f_i annualized fixed cost of keeping plant i open

c_{ij} cost of producing and shipping from i to j

D_j annual demand from market j

K_i potential capacity of plant i

$$\sum_{i=1}^{5} x_{i,j} = D_j \qquad j = 1,..m$$

$$\sum_{j=1}^{5} x_{i,j} \le K_i y_i \qquad j = 1,..n, \quad y_i \in \{0,1\}$$

✓

b) Python code:

```
 from scipy import optimize
import sys
import cvxopt
import matplotlib.pyplot as plt
import numpy as np
import sympy
sympy.init_printing()
np.set_printoptions(threshold = np.inf)
cost = np.array([[99,77,81,92,105],
            [100,88,105,100,110,],
            [105,100,77,120,100],
            [120,130,90,66,60],
            [150,100,99,110,55]])
Fl = np.array([4700,4600,4500,4400,4300])
Fh = np.array([5900,5700,5500,5300,5200])
D = np.array([15,10,15,20,10])
Kl = np.array([10,10,15,15,15])
Kh = np.array([25,25,20,20,20])
cFlat = cost.flatten('F')
c = np append(cFlat, Fh)
c = c.astype(float)
```

Only Fh is used! How's about Fl

Student ID: [redacted] Student name: [redacted]

print(np.rint(c.reshape((5,6), order = 'F')))

```
[[  99.    77.    81.    92.   105.  5900.]
 [ 100.    88.   105.   100.   110.  5700.]
 [ 105.   100.    77.   120.   100.  5500.]
 [ 120.   130.    90.    66.    60.  5300.]
 [ 150.   100.    99.   110.    55.  5200.]]
```

c

```
array([  99.,  100.,  105.,  120.,  150.,   77.,   88.,  100.,  130.,
        100.,   81.,  105.,   77.,   90.,   99.,   92.,  100.,  120.,
         66.,  110.,  105.,  110.,  100.,   60.,   55., 5900., 5700.,
       5500., 5300., 5200.])
```

len(c)

```
30
```

Au = np.kron(np.eye(5), np.ones((1,5)))

Ad = np.kron(np.ones((1,5)), np.eye(5))

Khigh = Kh*np.ones(5)

from scipy.optimize import linprog

res = linprog(c, A_ub=A, b_ub=b, A_eq=Aeq, b_eq=beq, method='revised simplex')

Does Khigh equal to Kh? Why does Kl never be used?

print(res)

```
     message: Optimization terminated successfully.
     success: True
      status: 0
         fun: 22515.0
           x: [ 0.000e+00  1.500e+01 ...  0.000e+00  5.000e-01]
         nit: 12
```

print(np.rint(res.x.reshape((5,6), order = 'F')))

```
[[ 0.  10.  15.  20.   0.   2.]
 [15.   0.   0.   0.   0.   1.]
 [ 0.   0.   0.   0.   0.   0.]
 [ 0.   0.   0.   0.   0.   0.]
 [ 0.   0.   0.   0.  10.   0.]]
```

print("The optimal value of the objective function is ", res.fun)

```
The optimal value of the objective function is  22515.0
```

*source:Topic 2: Topic 3: Modelling Supply Chain and Logistics Network workshop--Lab3

Student ID: [redacted] Student name: [redacted]

Answer: from the solution of Python, the optimal solution is 22515.0. The lowest cost network will have facilities located in North America, South America, and Africa. The high-capacity plant should be planned in each region. Besides, the plant in north America should take care of the demand of South America, Europe and Asia. The plant in South America needs to meet the demand of North America, and those in Africa need to meet the local demand.

Question 3 (15 marks)

Moon Micro is a small manufacturer of servers that currently builds all of its products in Santa Clara, California. As the market for servers has grown dramatically, the Santa Clara plant has reached a capacity of 10,000 servers per year. Moon is considering two options to increase its capacity. The first option is to add 10,000 units of capacity to the Santa Clara plant at an annualized fixed cost of $10,000,000 plus $500 labour per server. The second option is to have Molectron, an independent assembler, manufacture servers for Moon at the cost of $2,000 for each server (excluding raw materials cost). Raw materials cost $8,000 per server, and Moon sells each server for $15,000.

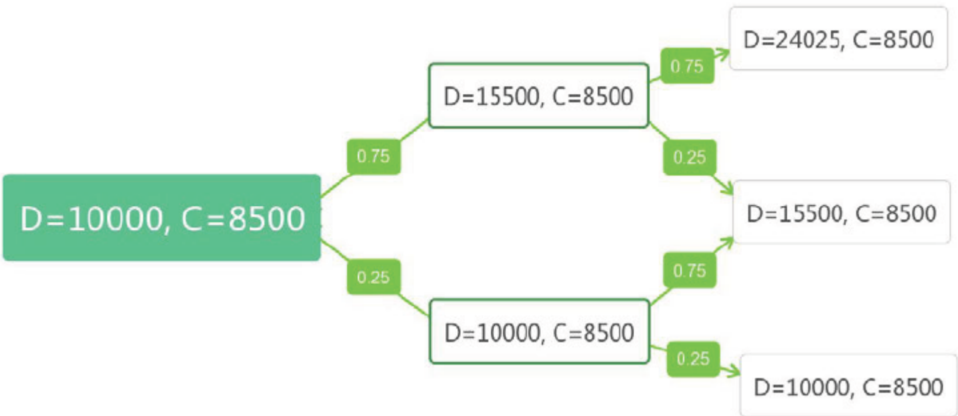
Moon must make this decision for a two-year time horizon. During each year, demand for Moon servers has a 75 per cent chance of increasing 55 per cent from the year before and a 25 percent chance of remaining the same as the year before. Molectron's prices may change as well.

a) Molectron's prices have a 60 per cent chance of increasing 25 per cent from the year before and a 40 per cent chance of remaining the same as the year before.
b) Molectron's prices are *fixed for the first year* but have a 60 per cent chance of increasing 25 per cent in the second year and a 40 per cent chance of remaining where they are.

Use a decision tree with a return rate of 0.1 to determine whether Moon should add capacity to its Santa Clara plant or if it should outsource to Molectron. What other factors would affect this decision that we have not discussed?

Solution:

OPTION 1:

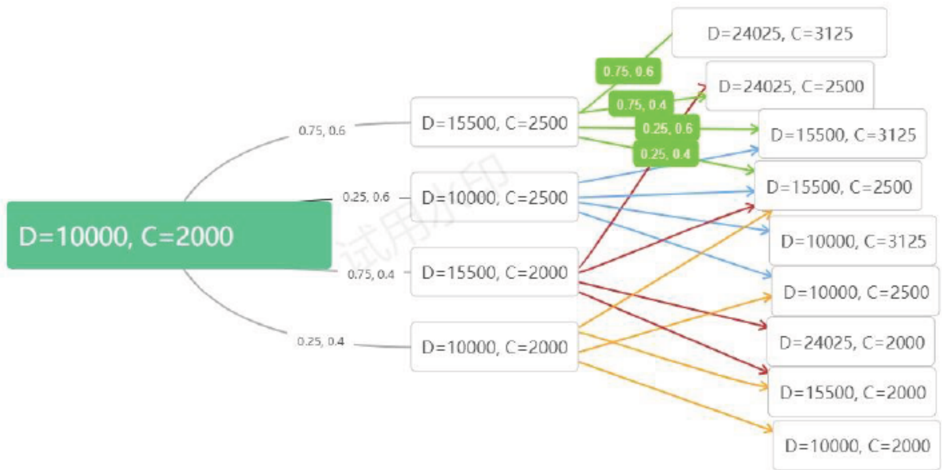


period 2					
D/Purchase	Labor and raw material per sever	revenue	Cost	FIxed Cost	Profit

Student ID: [redacted] Student name: [redacted]

24025	8500	300000000	230375000	10000000	59625000
15500	8500	232500000	131750000	10000000	90750000
10000	8500	150000000	85000000	10000000	55000000
period 1					
D	P	EP	PVEP	Profit	
15500	8500	67406250	61278409.09	1620 8409.1	
10000	8500	81812500	74375000	139375000	
period 0					
EP(D=10000, P=8500, 0) =	156365056.8				
PVEP(D=10000, P=8500, 0)=	142150051.7				
P(D=10000, P=8500, 0)=	207150051.7				

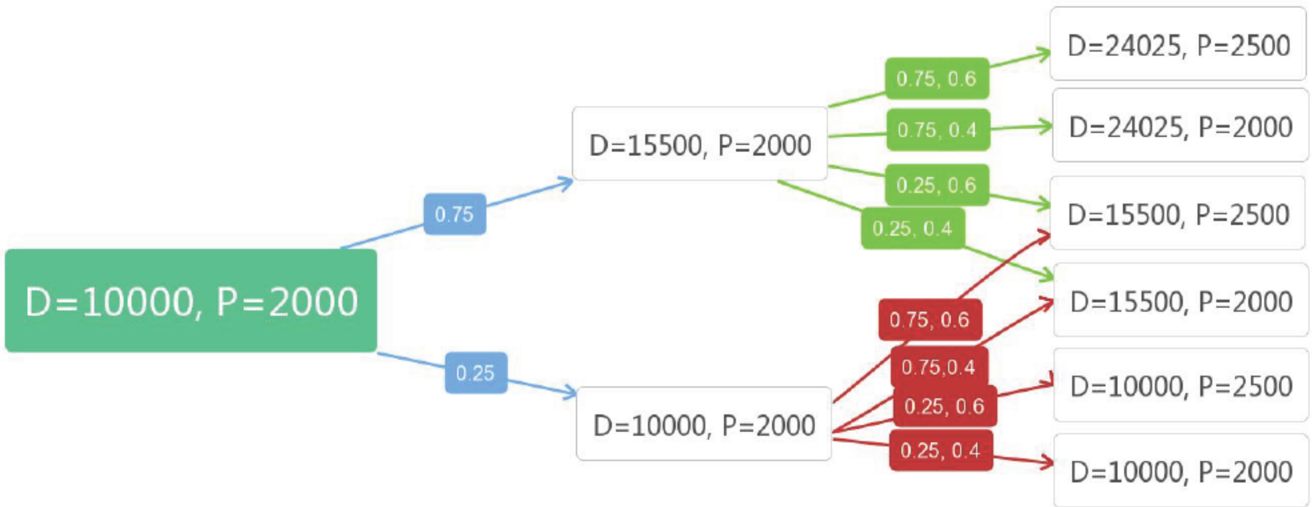
OPTION 2(a):



period 2				
D	P	revenue	Cost	Profit
24025	3125	360375000	267278125	93096875
24025	2500	360375000	252262500	108112500
15500	3125	232500000	172437500	60062500
15500	2500	232500000	162750000	69750000
10000	3125	150000000	111250000	38750000
10000	2500	150000000	105000000	45000000

24025	2000	360375000	240250000	120125000
15500	2000	232500000	155000000	77500000
10000	2000	150000000	100000000	50000000
period 1				
D	P	EP	PVEP	Profit
15500	2500	90311718.75	89417543.32	159167543.3
10000	2500	58265625	57688737.62	102688737.6
15500	2000	95344375	94400371.29	171900371.3
10000	2000	66387500	65730198.02	115730198
period 0				
EP(D=10000,P=10000,0)=	150171836.3			
PVEP(D=10000,P=10000,0)=	148684986.5			
P(D=10000,P=10000,0)=	198684986.5			

OPTION 2(b):



period 2				
D	P	revenue	Cost	Profit

24025	2500	360375000	252262500	108112500
24025	2000	360375000	240250000	120125000
15500	2500	232500000	162750000	69750000
15500	2000	232500000	155000000	77500000
10000	2500	150000000	105000000	45000000
10000	2000	150000000	100000000	50000000
period 1				
D	P	EP	PVEP ?	Profit
15500	2000	102900625	10188180 .9	179381806.9
10000	2000	66387500	65730198.02	115730198
period 0				
EP(D=10000, P=10000, 0)=	163468904.7			
PVEP(D=10000, P=10000, 0)=	161850400.7			
P(D=10000, P=10000, 0)=	211850400.7			

Question 4 (20 marks)

(a) Prefab, a furniture manufacturer, uses 20,000 square feet of plywood per month. Their trucking company charges Prefab $400 per shipment, independent of the quantity purchased. The manufacturer offers an all-unit quantity discount with a price of $1 per square foot for orders under 20,000 square feet, $0.98 per square foot for orders between 20,000 square feet and 40,000 square feet, and $0.96 per square foot for orders larger than 40,000 square feet. Prefab incurs a holding cost of 20 percent.

- What is the optimal lot size for Prefab?
- What is the annual cost of such a policy?
- What is the cycle inventory of plywood at Prefab?
- How does it compare with the cycle inventory if the manufacturer does not offer a quantity discount but sells all plywood at $0.96 per square foot?

(b) Reconsider Question 4(a) about Prefab. However, the manufacturer now offers a marginal unit quantity discount for the plywood. The first 20,000 square feet of any order is sold at $1 per square foot, the next 20,000 square feet is sold at $0.98 per square foot, and any quantity over 40,000 square feet is sold for $0.96 per square foot.

- What is the optimal lot size for Prefab given this pricing structure?

- How much cycle inventory of plywood will Prefab carry given the ordering policy?

(b) Verify the solutions in a) and b) using python code. (5 marks)

Solution:

(a) In this case, D=20000×12=240000, A=400, i=0.2.

Let $q_1 = 0, q_2, q_3$ be the order quantities a=t which the purchasing cost changes.

Let the corresponding unit purchasing cost be denoted by $c_j, j = 1,2,3$.

Let h_j, j=1,2,3, be the holding cost of an item at different discount cost.

From the question, we can get the discount table:

Quantity purchaed	c_j
1-20000	1
20000-40000	0.98
40000 and higher)	0.96

Thus, we can get:

$$q_1 = 0, q_2 = 20000, q_3 = 40000$$

$$c_1 = 1, c_2 = 0.98, c_3 = 0.96$$

$$h_1 = 0.2, h_2 = 0.196, h_3 = 0.192,$$

Due to the formulation:

$$Q^*_{c_j} = \sqrt{\frac{2DA}{h_j}}$$

So

$$Q^*_{c_1} = \sqrt{\frac{2 * 240000 * 400}{0.2}} \approx 30984$$

$$Q^*_{c_2} = \sqrt{\frac{2 * 240000 * 400}{0.196}} \approx 31298$$

$$Q^*_{c_3} = \sqrt{\frac{2 * 240000 * 400}{0.192}} \approx 31623$$

Since only $Q^*_{c_2}$ is feasible, because it belongs to $[q_2, q_3]$.

Then, the total cost have below answers, based on the formulation:

$$\text{TC} = (\frac{\text{D}}{\text{Q}}) \times \text{A} + (\frac{Q}{2}) \times \text{i}c_j + c_j \times D$$

Hence,

$$TC_{Q=31298} = (\frac{240000}{31298}) \times 400 + (\frac{31298}{2}) \times 0.2 \times 0.98 + 0.98 \times 240000 \approx 241334$$

$$TC_{Q=40000} = (\frac{240000}{40000}) \times 400 + (\frac{40000}{2}) \times 0.2 \times 0.96 + 0.96 \times 240000 \approx 236640$$

In conclusion, $TC_{Q=40000}$ is lower, thus, they need to adjusted upward to 40000 to get the lower total cost that is \$236640.

$$\text{Cycle inventory} = \frac{\text{Lot size}}{2} = \frac{40000}{2} = 20000$$

So the cycle inventory of plywood is 20000.

If the price of plywood is \$0.96 per square without discounting, the optimal lot size is

$$Q^* = \sqrt{\frac{2DA}{h}} = \sqrt{\frac{2 \times 240000 \times 400}{0.2 \times 0.96}} \approx 31623$$

$$\text{Cycle inventory} = \frac{\text{Lot size}}{2} = \frac{31623}{2} \approx 15811$$

Consequently, the cycle inventory of plywood with discount is greater than that without discount.

(b)

$$D = 20000 \times 12 = 240000, m = 3, q_1 = 0, q_2 = 20000, q_3 = 40000$$

$$c_1 = 1, c_2 = 0.98, c_3 = 0.96$$

Based on the formulation:

$$R_j = c_1(q_2 - q_1) + \ldots + c_{j-1}(q_j - q_{j-1})$$

$$Q_j^* = \sqrt{\frac{2(R_j - c_j \times q_j + A)D}{i \times c_j}}$$

Then we can compute that:

$$R_1 = 0, R_2 = 1 \times (20000 - 0) = 20000, R_3 = R_2 + 0.98 \times (40000 - 20000) = 39600$$

$$Q_1^* = \sqrt{\frac{2(0 - 1 \times 0 + 400) \times 240000}{0.2 \times 1}} \approx 30984$$

$$Q_2^* = \sqrt{\frac{2(20000 - 0.98 \times 20000 + 400) \times 240000}{0.2 \times 0.98}} \approx 44263$$

$$Q_3^* = \sqrt{\frac{2(39600 - 0.96 \times 40000 + 400) \times 240000}{0.2 \times 0.96}} \approx 63246$$

Only the Q_3^* is feasible which lies in the interval $(40000, \infty)$, so 63246 is the optimal lot size.

Then we can get the cylce inventory of this optimal lot size is $63246 \div 2 = 31623$

(c)

Python code:

(i)

```
import matplotlib as mpl

import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d.axes3d import Axes3D

import numpy as np

import sympy

from scipy import stats

import math

D = 240000

A = 400

I = 0.2

h = np.array([0.0, 0.0, 0.0])

C = np.array([1.00, 0.98, 0.96])

q  = np.array([0, 20000, 40000])

TCMin = 0.0

for i in (0, 1, 2):

  h[i] = I*C[i]

  Qcal = np.sqrt(2*D*A/h[i])

  if i == 0:

    if Qcal < q[i+1]:

      Qp = Qcal

    else:

      Qp = q[i+1] - 1

  elif i == 1:

    if Qcal < q[i]:

      Qp = q[i]

    elif Qcal > q[i+1]:

      Qp = q[i+1] - 1

  else:

    if Qcal < q[i]:
```

```
            Qp = q[i]
        else:
            Qp = Qcal
    TC = (D/Qp) * A + (Qp/2)*h[i] + C[i]*D
    if i == 0:
        TCMin = TC
        EOQ = Qp
        hi = h[i]
        Ci = C[i]
    else:
        if TC < TCMin:
            TCMin = TC
            EOQ = Qp
            hi = h[i]
            Ci = C[i]
n = D / EOQ
HC = (EOQ / 2) * hi
OC = n * A
PC = D * Ci
TotalC = (HC + OC + PC)
Cycleinventory = EOQ / 2
print('The optimal lost size =', EOQ)
print('Annual inventory cost   = ', TotalC)
print('Cycle inventory =', Cycleinventory)
```

```
The optimal lost size = 40000
Annual inventory cost    =  236640.0
Cycle inventory = 20000.0
```

(ii)

```
p = 0.96
H = I * p
Q_star = math.sqrt(2 * D * A / H)
```

Student ID: [redacted] Student name: [redacted]

```
cycin = Q_star/2
print('the optimal lot size =', Q_star)
print('Cycle inventory without discount at the price of 0.96 =',cycin)
```

```
the optimal lot size = 31622.776601683792
Cycle inventory without dicount at the price of 0.96 = 15811.388300841896
```

(iii)

```
def Rcal(j, q, C):
 sumR = 0.0
 if j == 0:
  sumR = 0
 else:
  for i in range(j):
   sumR += C[i]*(q[i+1]-q[i])
 return sumR
import numpy as np
D = 240000
A = 400
I = 0.2
C = np.array([1.00, 0.98, 0.96])
q  = np.array([0, 20000, 40000])
Q = []
R = []
for j in range(len(q)):
 Rj = Rcal(j, q, C)
 Qj = round(np.sqrt(2*(Rj - C[j]*q[j] + A) *D/(I*C[j])),2)
 if j==0:
  if Qj > 0 and Qj < q[j+1]:
   Q.append(Qj)
  else:
   Q.append(0)
 elif j > 0 and j < len(q)-1:
```

```
    if Qj >= q[j] and Qj < q[j+1]:
      Q.append(Qj)
    else:
      Q.append(0)
  else:
    if Qj >= q[j]:
        Q.append(Qj)
    else:
      Q.append(0)
  R.append(Rj)
print(R)
print('the optimal lot sizs=',Q[3])
cyin=Q[3]/2
print('cylce inventory=', cyin)
```

```
[0, 20000.0, 39600.0]
the optimal lot sizs= 63245.55
cylce inventory= 31622.775
```

*source: Topic 5: Deterministic inventory models: Extended EOQ and EPQ workshop--Lab5-2023

Question 5 (20 marks)

Sun.com sells three models of 6 × 250 W solar panels. These models are the polar panel, the PV polar panel, and the off-grid polar panel. Annual demands for the three products are 12,000 units for the polar panel, 1,200 units for the PV polar panel, and 120 units for the 0ff-grid polar panel. Each model costs Sun.com $500. A fixed transportation of $4,000 is incurred each time an order is delivered. For each model ordered and delivered on the same truck, an additional fixed cost of $1,000 is incurred for receiving and storage. Sun.com incurs a holding cost of 20 percent. Evaluate the optimal lot size for each model, and also evaluate the annual cost of such a policy. How many packed should Sun.com produce in each batch?

a) Three models of products are ordered and delivered independently. (5 marks)

b) The manager has decided to aggregate and order all three models together. (10 marks)

c) Verify the solutions in a) and b) using python code. (5 marks)

Solution:

a)

Annual demand of each product: $D_1 = 12000, D_2 = 1200, D_3 = 120$;

Student ID: [redacted] Student name: [redacted]

Unit cost: $C_1 = 500, C_2 = 500, C_3 = 500;$

Common order cost: $\mathrm{A} = 4000;$

Additional fixed cost: $A_1 = 1000, A_2 = 1000, A_3 = 1000;$

Holding Cost: $\mathrm{I} = 0.2$

$$\mathrm{A}^* = \mathrm{A} + \mathrm{A_i} = 4000 + 1000 = 5000$$

Based on the formulation, $Q^* = \sqrt{\frac{2DA}{IC}}$, we can get the optimal lot size for each model:

$$Q_1^* = \sqrt{\frac{2 \times 12000 \times 5000}{0.2 \times 500}} \approx 1095$$

$$Q_2^* = \sqrt{\frac{2 \times 1200 \times 5000}{0.2 \times 500}} \approx 346$$

$$Q_3^* = \sqrt{\frac{2 \times 120 \times 5000}{0.2 \times 500}} \approx 110$$

Therefore, the optimal lot size of the polar panel, the PV polar panel, and the off-grid polar panel are 1095, 346, 110 units respectively.

According to the formula, $\mathrm{TC} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \sqrt{2 \times \mathrm{D_i} \times \mathrm{A_i} \times \mathrm{h}}$, we can get the annual cost of each model:

$$\mathrm{TC} = \sqrt{2 \times 12000 \times 5000 \times 0.2 \times 500} + \sqrt{2 \times 1200 \times 5000 \times 0.2 \times 500} + \sqrt{2 \times 120 \times 5000 \times 0.2 \times 500} \approx 155140.$$

Hence, the annual cost of such policy is equal to 155140.

b)

The combined order cost is :

$$\mathrm{A}^* = \mathrm{A} + A_1 + A_2 + A_3 = 4000 + 1000 + 1000 + 1000 = 7000$$

The optimal order frequency is given by the formula

$$n^* = \sqrt{\frac{\sum_{j=1}^{k} D_j I C_j}{2A^*}}$$

The optimal lot size of each products is given by the formula

$$Q^* = \frac{D}{\mathrm{n}^*}$$

The total annual cost is given by :

$$\mathrm{TC} = \text{Annual order cost} + \text{total annual holding cost}$$

$$= \mathrm{nA}^* + \frac{\sum_{\mathrm{j}=1}^{\mathrm{k}} \mathrm{D_j} \mathrm{IC_j}}{2\mathrm{n}}$$

Then we can get the table:

	the polar panel	the PV polar panel	the off-grid polar panel
Demand per year(D)	12000	1200	120
Order frequency(n^*)	9.75	9.75	9.75
Optimal order size(Q^*)	1230	123	12.3
Annual order cost	68279		
Annual holding cost	61512	6151	615
Total annual cost	136558		

c)

(i)

Python code:

```
import matplotlib as mpl

import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d.axes3d import Axes3D

import numpy as np

import sympy

from scipy import stats

import math

def EOQ(D, A, h):

  TC = np.sqrt(2*D*A*h)

  Q  = np.sqrt(2*D*A/h)

  n  = D/Q

  T  = Q/D

  return TC, Q, n, T

D = np.array([12000, 1200, 120])

A = np.array([5000, 5000, 5000])

C = np.array([500, 500, 500])

I = 0.2

TC = np.array([0.00, 0.00, 0.00])

Q  = np.array([0.00, 0.00, 0.00])

n  = np.array([0.00, 0.00, 0.00])
```

Student ID: [redacted] Student name: [redacted]

```
T  = np.array([0.00, 0.00, 0.00])
HC  = np.array([0.00, 0.00, 0.00])
OC  = np.array([0.00, 0.00, 0.00])
TotalC = 0.0
for i in (0, 1, 2):
    h = I*C[i]
    TC[i], Q[i], n[i], T[i] = EOQ(D[i], A[i], h)
    HC[i] = Q[i]*h/2
    OC[i] = A[i]*n[i]
    TotalC = TotalC + TC[i]
    from tabulate import tabulate
    rnames = ["the polar panel", "the PV polar panel", "the off-grid polar panel"]
    headers = ["Optimal lot size", "Number of order", "Cycle length",
            "Holding cost", "Ordering cost", "Total variable cost"]
    table = zip(rnames, np.round(Q), np.round(n), T, HC, OC, TC)
    print(tabulate(table, headers=headers, floatfmt=".2f"))
```

	Optimal lot size	Number of order	Cycle length	Holding cost	Ordering cost	Total variable cost
the polar panel	1095.00	11.00	0.09	54772.26	54772.26	109544.51
the PV polar panel	346.00	3.00	0.29	17320.51	17320.51	34641.02
the off-grid polar panel	110.00	1.00	0.91	5477.23	5477.23	10954.45

```
    print('Total inventory-related cost : ', round(TotalC, 2))
```

```
Total inventory-related cost :  155139.98
```

(ii)

```
D = np.array([12000, 1200, 120])
A = np.array([1000, 1000, 1000])
C = np.array([500, 500, 500])
I = 0.2
AC = 4000
Q  = np.array([0.00, 0.00, 0.00])
HC  = np.array([0.00, 0.00, 0.00])
Sum = 0.0
AStar = AC
```

```
for i in (0, 1, 2):
   Sum = Sum + D[i]*I*C[i]
   AStar = AStar + A[i]
print('A* = ', AStar)
nStar = np.sqrt(Sum/(2*AStar))
print('n* = ', nStar)
OC = AStar * nStar
Sum = 0
Qmax = 0
for i in (0, 1, 2):
   Q[i] = D[i]/nStar
   if Q[i] > Qmax:
      Qmax = Q[i]
      Dc = D[i]
   HC[i] = Q[i]*I*C[i]/2
   Sum = Sum + HC[i]
TotalC = Sum + OC
T = Qmax / Dc
```

```
A* =  7000
n* =  9.754120008635178
```

```
rnames = ["the polar panel", "the PV polar panel", "the off-grid polar panel"]
headers = ["Optimal lot size", "Holding cost"]
table = zip(rnames, np.round(Q), HC)
print(tabulate(table, headers=headers, floatfmt=".2f"))
```

```
                            Optimal lot size    Holding cost
--------------------------  ------------------  --------------
the polar panel                        1230.00        61512.47
the PV polar panel                      123.00         6151.25
the off-grid polar panel                 12.00          615.12
```

```
print('Number of order          : ', round(nStar))
print('Total Ordering cost      : ', round(OC,2))
print('Total Holding cost       : ', round(sum(HC),2))
```

Student ID: [redacted] Student name: [redacted]

```
print('Cycle length (year)          : ', round(T,2))

print('Total inventory-related cost : ', round(TotalC,2))
```

```
Number of order               :  10
Total Ordering cost           :  68278.84
Total Holding cost            :  68278.84
Cycle length (year)           :  0.1
Total inventory-related cost  :  136557.68
```

*source: Topic 6: Deterministic inventory models: EOQ model with price break; Multi Commodity EOQ Models workshop--Lab6-SOL

Question 6 (15 marks)

Swanson's Bakery is well known for producing the best fresh bread in the city, so the sales are very substantial. The daily demand for its fresh bread has a uniform distribution between 300 and 600 loaves. The bread is baked in the early morning, before the bakery opens for business, at a cost of $2 per loaf. It then is sold that day for $3 per loaf. Any bread not sold on the day it is baked is relabelled as day-old bread and sold subsequently at a discount price of $1.50 per loaf.

a) Apply the stochastic single-period model for perishable products to determine the optimal service level. (5 marks)
b) Given your answer in part (a), what is the probability of incurring a shortage of fresh bread on any given day? (5 marks)
c) Because the bakery's bread is so popular, its customers are quite disappointed when a shortage occurs. The owner of the bakery, Ken Swanson, places high priority on keeping his customers satisfied, so he doesn't like having shortages. He feels that the analysis also should consider the loss of customer goodwill due to shortages. Since this loss of goodwill can have a negative effect on future sales, he estimates that a cost of $1.50 per loaf should be assessed each time a customer cannot purchase fresh bread because of a shortage. Determine the new optimal number of loaves to bake each day with this change. What is the new probability of incurring a shortage of fresh bread on any given day? (5 marks)

Solution:

a)

Overage cost:

C_o 1

Underage cost:

C

Thus, we can get:

So, the optimal service level is:

$Q^* = a + (b -$ $\times 0.5 = 450$

b)

$$P(\text{stockout}) = P(\text{demand} \geq Q) = 1 - P(\text{demand} \leq Q) = 1 - 0.5 = 0.5$$

Consequently, the probability of incurring a shortage is 0.5.

Which cost is given?

c)

$$C_o = cost\ price - salvage\ value = 1.5 - 1 = 0.5$$

$$C_u = selling\ price - cost\ price = 3 - 1.5 = 1.5$$

$$P(\text{demand} \leq Q) = \frac{C_u}{C_o + C_u} = \frac{1.5}{1.5 + 0.5} = 0.75$$

$$P(\text{stockout}) = P(\text{demand} \geq Q) = 1 - P(\text{demand} \leq Q) = 1 - 0.75 = 0.25$$

At a cost of $1.5, the probability of incurring shortage of fresh bread is 0.25.