Report

DNSC_6307 - Optimization I
SeungHeon Han

i. provide mathematical formulation for the production/stock policy problem with the goal of minimizing the total cost.

- **Decision variables:**

x1_1 = Regular Production in the 1st month
x1_2 = Extra Production in the 1st month
x2_1 = Regular Production in the 2nd month
x2_2 = Extra Production in the 2nd month
x3_1 = Regular Production in the 3rd month
x3_2 = Extra Production in the 3rd month
x4_1 = Regular Production in the 4th month
x4_2 = Extra Production in the 4th month
s1 = Stock produced in the 1st month
s2 = Stock produced in the 2nd month
s3 = Stock produced in the 3rd month

- **Objective Function:**

Minimizing $440*(x1_1+x2_1+x3_1+x4_1) + ($440+$260)*(x1_2+x2_2+x3_2+x4_2) + $5*(s1+s2+s3)

- **Constraints:**

[1] Regular production <= max regular production

1) x1_1 <= 140
2) x2_1 <= 150
3) x3_1 <= 140
4) x4_1 <= 160

[2] Extra production <= max extra production

5) x1_2 <= 50
6) x2_2 <= 75
7) x3_2 <= 70
8) x4_2 <= 80

[3] stock(i-1) + production(i) - stock(i) = demand(i)

9) x1_1 + x1_2 - s1 = 120
10) s1 + x2_1 + x2_2 - s2 = 160
11) s2 + x3_1 + x3_2 - s3 = 300
12) s3 + x4_1 + x4_2 = 200

[4] Max capacity of warehouse

13) s1 <= 100
14) s2 <= 100
15) s3 <= 100

[5] Regular production in each month >= 10% of the total production of the first three months

16) x1_1 >= 0.1*(x1_1+x1_2+x2_1+x2_2+x3_1+x3_2)
17) x2_1 >= 0.1*(x1_1+x1_2+x2_1+x2_2+x3_1+x3_2)
18) x3_1 >= 0.1*(x1_1+x1_2+x2_1+x2_2+x3_1+x3_2)
19) x4_1 >= 0.1*(x1_1+x1_2+x2_1+x2_2+x3_1+x3_2)

[6] Non-negativity constraints

20) x1_1 >= 0
21) x1_2 >= 0
22) x2_1 >= 0
23) x2_2 >= 0
24) x3_1 >= 0
25) x3_2 >= 0
26) x4_1 >= 0
27) x4_2 >= 0
28) s1 >= 0
29) s2 >= 0
30) s3 >= 0

ii. Code the problem using Python and Gurobi and include the code in your answer.

- **Python code:**

```
# Packages

import gurobipy as gp
from gurobipy import GRB
import pandas as pd
from collections import OrderedDict

# Model

m = gp.Model("PB")

# Decision variables for the regular, extra production, and stock by month

x1_1 = m.addVar(name="Normal_Prod1")
x1_2 = m.addVar(name="Extra_Prod1")
x2_1 = m.addVar(name="Normal_Prod2")
x2_2 = m.addVar(name="Extra_Prod2")
x3_1 = m.addVar(name="Normal_Prod3")
x3_2 = m.addVar(name="Extra_Prod3")
```

```
x4_1 = m.addVar(name="Normal_Prod4")
x4_2 = m.addVar(name="Extra_Prod4")
s1 = m.addVar(name="stock1")
s2 = m.addVar(name="stock2")
s3 = m.addVar(name="stock3")

# Formulation

#### Objective Function ####
# The objective is to minimize the total cost

obj = 440*(x1_1+x2_1+x3_1+x4_1) + 700*(x1_2+x2_2+x3_2+x4_2)+5*(s1+s2+s3)
m.setObjective(obj, GRB.MINIMIZE)

#### Constraints ####
# Regular production <= max regular production

con1 = m.addConstr(x1_1 <= 140, name = "max_normProd1")
con2 = m.addConstr(x2_1 <= 150, name = "max_normProd2")
con3 = m.addConstr(x3_1 <= 140, name = "max_normProd3")
con4 = m.addConstr(x4_1 <= 160, name = "max_normProd4")

# Extra production <= max extra production

con5 = m.addConstr(x1_2 <= 50, name = "max_extraProd1")
con6 = m.addConstr(x2_2 <= 75, name = "max_extraProd2")
con7 = m.addConstr(x3_2 <= 70, name = "max_extraProd3")
con8 = m.addConstr(x4_2 <= 80, name = "max_extraProd4")

# stock(i-1) + production(i) - stock(i) = demand(i)

con9 = m.addConstr(x1_1+x1_2-s1 == 120, name = "meet_demand1")
con10 = m.addConstr(s1+x2_1+x2_2-s2 == 160, name = "meet_demand2")
con11 = m.addConstr(s2+x3_1+x3_2-s3 == 300, name = "meet_demand3")
con12 = m.addConstr(s3+x4_1+x4_2 == 200, name = "meet_demand4")

# Max capacity of warehouse

con13 = m.addConstr(s1 <= 100, name = "max_cap1")
con14 = m.addConstr(s2 <= 100, name = "max_cap2")
con15 = m.addConstr(s3 <= 100, name = "max_cap3")

# Normal production of each month >= 10% of the total production of the first three months

con16 = m.addConstr(x1_1 >= 0.1*(x1_1+x1_2+x2_1+x2_2+x3_1+x3_2), name = "balanced_prod1")
con17 = m.addConstr(x2_1 >= 0.1*(x1_1+x1_2+x2_1+x2_2+x3_1+x3_2), name = "balanced_prod2")
con18 = m.addConstr(x3_1 >= 0.1*(x1_1+x1_2+x2_1+x2_2+x3_1+x3_2), name = "balanced_prod3")
con19 = m.addConstr(x4_1 >= 0.1*(x1_1+x1_2+x2_1+x2_2+x3_1+x3_2), name = "balanced_prod4")

# Non-negativity constraints

con20 = m.addConstr(x1_1 >= 0, name = "NonNegative1_1")
con21 = m.addConstr(x1_2 >= 0, name = "NonNegative1_2")
con22 = m.addConstr(x2_1 >= 0, name = "NonNegative2_1")
con23 = m.addConstr(x2_2 >= 0, name = "NonNegative2_2")
```

```
con24 = m.addConstr(x3_1 >= 0, name = "NonNegative3_1")
con25 = m.addConstr(x3_2 >= 0, name = "NonNegative3_2")
con26 = m.addConstr(x4_1 >= 0, name = "NonNegative4_1")
con27 = m.addConstr(x4_2 >= 0, name = "NonNegative4_2")
con28 = m.addConstr(s1 >= 0, name = "NonNegative_s1")
con29 = m.addConstr(s2 >= 0, name = "NonNegative_s2")
con30 = m.addConstr(s3 >= 0, name = "NonNegative_s3")

# Solve

m.optimize()

# Print optimal value of the objective function

print('\nTotal cost to produce rice over 4 months: $ %g' % m.objVal)

# Print optimal values for the decision variables

print('\nDecision variables:')

for v in m.getVars():
    print('%s = %g' % (v.varName, v.x))
```

iii. solve the problem that you have coded and report (a) the optimal monthly production amounts and (b) the total cost.

- **Optimal variables:**
  (unit = ton)

  Regular Production in month 1 = 140
  Extra Production in month 1 = 5
  Stock in month 1= 25

  Regular Production in month 2 = 150
  Extra Production in month 2 = 75
  Stock in month 2 = 90

  Regular Production in month 3 = 140
  Extra Production in month 3= 70
  Stock in month 3 = 0

  Regular Production in month 4 = 160
  Extra Production in month 4 = 40

- **Total cost:**

$ 393,175