PROBLEM1,

Name: Wanwen Hu 100376272 Assignment 2

Problem 1

a. set x_{ij} from i to j

$$\max z = x_{8,11} + x_{9,11} + x_{10,11}$$

s.t.

$$\left.\begin{aligned}
x_{1,3} &= x_{3,6} + x_{3,7} \\
x_{1,4} + x_{2,4} &= x_{4,6} + x_{4,7} \\
x_{1,5} + x_{2,5} &= x_{5,6} + x_{5,7} \\
x_{3,6} + x_{4,6} + x_{5,6} &= x_{6,8} + x_{6,9} \\
x_{4,7} + x_{5,7} &= x_{7,8} + x_{7,9} + x_{7,10} \\
x_{6,8} + x_{7,8} &= x_{8,11} \\
x_{6,9} + x_{7,9} &= x_{9,11} \\
x_{7,10} &- x_{10,11}
\end{aligned}\right\} \Rightarrow
\begin{aligned}
x_{1,3} - x_{3,6} - x_{3,7} &= 0 && y_1 \\
x_{1,4} + x_{2,4} - x_{4,6} - x_{4,7} &= 0 && y_2 \\
x_{1,5} + x_{2,5} - x_{5,6} - x_{5,7} &= 0 && y_3 \\
x_{3,6} + x_{4,6} + x_{5,6} - x_{6,8} - x_{6,9} &= 0 && y_4 \\
x_{4,7} + x_{5,7} - x_{7,8} - x_{7,9} - x_{7,10} &= 0 && y_5 \\
x_{6,8} + x_{7,8} - x_{8,11} &= 0 && y_6 \\
x_{6,9} + x_{7,9} - x_{9,11} &= 0 && y_7 \\
x_{7,10} - x_{10,11} &= 0 && y_8
\end{aligned}$$

at each arcs the capacity is limited by

$x_{1,3} \le 83$	$y_{1,3}$
$x_{1,4} \le 72$	$y_{1,4}$
$x_{1,5} \le 65$	$y_{1,5}$
$x_{2,4} \le 84$	$y_{2,4}$
$x_{2,5} \le 77$	$y_{2,5}$
$x_{3,6} \le 69$	$y_{3,6}$
$x_{3,7} \le 45$	$y_{3,7}$
$x_{4,6} \le 39$	$y_{4,6}$
$x_{4,7} \le 83$	$y_{4,7}$
$x_{5,6} \le 91$	$y_{5,6}$
$x_{5,7} \le 68$	$y_{5,7}$
$x_{6,8} \le 49$	$y_{6,8}$
$x_{6,9} \le 52$	$y_{6,9}$
$x_{7,8} \le 72$	$y_{7,8}$
$x_{7,9} \le 37$	$y_{7,9}$
$x_{7,10} \le 81$	$y_{7,10}$
$x_{8,11} \le 82$	$y_{8,11}$
$x_{9,11} \le 103$	$y_{9,11}$
$x_{10,11} \le 92$	$y_{10,11}$

1

1b.

```
from gurobipy import *
  # Model data
nodes,supply=multidict({3:0,4:0,5:0,6:0,7:0,8:0,9:0,10:0})
arcs, upcap = multidict({ (1, 3):    83, (1, 4):    72, (1, 5):    65, (2, 4):    84, (2, 5):    77,
```

```
(3, 6):    69, (3, 7):    45,(4, 6):    39, (4, 7):    83, (5, 6):    91, (5, 7):    68, (6, 8):    49,
(6, 9):    52, (7, 8):    72, (7, 9):    37, (7, 10): 81, (8, 11): 82, (9, 11): 103, (10, 11):97})
 # Create optimization model
m = Model('MaxSupplyNetwork')
# Create variables
flow = m.addVars(arcs, name="flow")
# Flow balance constraints
m.addConstrs(
            (flow.sum(i, '*') - flow.sum('*', i) == 0 for i in nodes[2:10]), "supply")
# Upper arc capacity constraints
m.addConstrs(
            (flow[i, j] <= upcap[i, j] for i, j in arcs), "upCap")
#Setting objective function
#m.setObjective
m.setObjective(flow[8,11]+flow[9,11]+flow[10,11], GRB.MAXIMIZE)
m.ModelSense=GRB.MAXIMIZE
# Compute optimal solution
m.optimize()
# Print solution
print('\nVariable Information Including Sensitivity Information:\n')
for v in m.getVars():
    print("%s %s %8.2f %s %8.2f %s %8.2f %s %8.2f" %
                (v.Varname, "=", v.X, ", reduced cost = ", abs(v.RC), ", from coeff = ", v.SAObjLow,
"to coeff = ", v.SAObjUp))
    print(" ")
print('\nOptimal objective value: %g' % m.objVal)
print('\nOptimal shadow prices:\n')
for c in m.getConstrs():
        print("%s %s %8.2f %s %8.2f %s %8.2f" % (c.ConstrName, ": shadow price = ", c.Pi, ",
from RHS = ", c.SARHSLow, "to RHS = ", c.SARHSUp))
        print(" ")
```

```
Solved in 0 iterations and 0.16 seconds
Optimal objective  2.520000000e+02

Variable Information Including Sensitivity

flow[1,3] =     0.00 , reduced cost =
```

...

```
Optimal objective value: 252

Optimal shadow prices:

supply[5] : shadow price =      0.00 , f

supply[6] : shadow price =      0.00 , f

supply[7] : shadow price =      0.00 , f

supply[8] : shadow price =      0.00 , f

supply[9] : shadow price =      1.00 , f

supply[10] : shadow price =      1.00 ,
```

```
upCap[8,11] : shadow price =      1.00 , from RHS =     82.00 to RHS =    121.00

upCap[9,11] : shadow price =      0.00 , from RHS =     89.00 to RHS =
1000000000000000015902891109759918046836080856394528138978132755774783877217038106

upCap[10,11] : shadow price =      0.00 , from RHS =     81.00 to RHS =
1000000000000000015902891109759918046836080856394528138978132755774783877217038106
```

(b). optimal: 252 where $x_{1,3}=0$, $x_{1,4}=0$, $x_{1,5}=0$, $x_{2,4}=0$ $x_{2,5}=68$

$x_{3,6}=62$ $x_{3,7}=0$ $x_{4,6}=39$ $x_{4,7}=83$, $x_{5,6}=0$

$x_{5,7}=68$ $x_{6,8}=49$, $x_{6,9}=52$ $x_{7,8}=33$ $x_{7,9}=37$

$x_{7,10}=37$ $x_{7,10}=81$ $x_{8,11}=82$, $x_{9,11}=89$ $x_{10,11}=81$

c. Dual problem

$$\min d = 83y_{1,3}+72y_{1,4}+65y_{1,5}+84y_{2,4}+77y_{2,5}+69y_{3,6}+45y_{3,7}+39y_{4,6}+83y_{4,7}+91y_{5,6}+88y_{5,}$$
$$+49y_{6,8}+52y_{6,9}+72y_{7,8}+37y_{7,9}+81y_{7,10}+82y_{8,11}+103y_{9,11}+92y_{10,11}+0\sum_{i=1}^{8}y_i$$

st

$y_{1,3}+y_1 \geq 0$

$y_{1,4}+y_2 \geq 0$

$y_{1,5}+y_3 \geq 0$

$y_{2,4}+y_2 \geq 0$

$y_{2,5}+y_3 \geq 0$

$y_{3,6}-y_1+y_4 \geq 0 \Rightarrow y_{3,6} \geq y_1-y_4$

$y_{3,7}-y_1 \geq 0$

$y_{4,6}-y_2+y_4 \geq 0 \Rightarrow y_{4,6} \geq y_2-y_4$

$y_{4,7}-y_2+y_5 \geq 0$ $\quad y_{4,7} \geq y_2-y_5$

$y_{5,6}-y_3+y_4 \geq 0$ $\quad y_{5,6} \geq y_3-y_4$

$y_{5,7}-y_3+y_5 \geq 0$ $\quad y_{5,7} \geq y_3-y_5$

$y_{6,8}-y_4+y_6 \geq 0$ $\quad y_{6,8} \geq y_4-y_6$

$y_{6,9}-y_4+y_7 \geq 0$ $\quad y_{6,9} \geq y_4-y_7$

$y_{7,8}-y_5+y_6 \geq 0$ $\quad y_{7,8} \geq y_5-y_6$

$y_{7,9}-y_5+y_7 \geq 0$ $\quad y_{7,9} \geq y_5-y_7$

$y_{7,10}-y_5+y_8 \geq 0$ $\quad y_{7,10} \geq y_5-y_8$

$y_{8,11}-y_6 \geq 1$ $\quad y_{8,11} \geq y_6+1$

$y_{9,11}-y_7 \geq 1$ $\quad y_{9,11} \geq y_7+1$

$y_{10,11}-y_8 \geq 1$ $\quad y_{10,11} \geq y_8+1$

All $y \in [0,1]$

From the output in (b) we know that The optimal of (c)is also 252

d,I prioritize nodes8 to 11(8,11),since the shadow price no equal to 0,(8,11) increase the RHS the most.

Problem2:

a. x_i is the number of products. y_i: $\begin{cases}1\\0\end{cases}$ whether set up the production line or not.

$$\max z = 48x_1 + 55x_2 + 50x_3 + 52x_4 - 1000y_1 - 800y_2 - 900y_3 - 950y_4$$

s.t.

$$2x_1 + 3x_2 + 6x_3 + 5x_4 \le 600$$
$$6x_1 + 3x_2 + 4x_3 + 3x_4 \le 300$$
$$5x_1 + 6x_2 + 2x_3 + 2x_4 \le 400$$

to each product ~~$x_1 \ge \frac{1000}{48}$~~ when produce only 1 type product, the upper capacity

$$\begin{cases}2x_1 \le 600\\6x_1 \le 300\\5x_1 \le 400\end{cases} \Rightarrow \begin{cases}x_1 \le 300\\x_1 \le 50\\x_1 \le 80\end{cases} \Rightarrow x_1 \le 50$$

$$\begin{cases}3x_2 \le 600\\3x_2 \le 300\\6x_2 \le 400\end{cases} \Rightarrow \begin{cases}x_2 \le 200\\x_2 \le 100\\x_2 \le \frac{400}{6} \approx 66.7\end{cases} \Rightarrow x_2 \le \frac{400}{6}$$

$$\begin{cases}6x_3 \le 600\\4x_3 \le 300\\2x_3 \le 400\end{cases} \Rightarrow \begin{cases}x_3 \le 100\\x_1 \le \frac{300}{4} = 75\\x_3 \le 200\end{cases} \quad \{\, x_3 \le 75$$

$$\begin{cases}5x_4 \le 600\\3x_4 \le 300\\2x_4 \le 400\end{cases} \Rightarrow \begin{cases}x_4 \le 120\\x_4 \le 100\\x_4 \le 200\end{cases} \quad \{\, x_4 \le 100$$

$y_i \in [0,1]$

x_i is integer

In order to get a prositive profit,to each products

$$48x_1 \ge 1000y_1, 55x_2 \ge 800y_2, 50x_3 \ge 900y_3, 52x_4 \ge 950y_4$$

Problem2:code

```
from gurobipy import *
m = 3 # number of operations
n = 4 # number of products
p = 4 #numeber of binary variables auxiliar
operations = range(1, m+1)    # list [1, ..., n]
products = range(1, n+1)    # list [1, ..., n]
auxiliar = range(1, p+1)    # list [1, ..., p]
# primal objective coefficients
r_coeff = [48, 55, 50, 52]
# primal objective coefficients for the auxiliar
```

```
aux_coeff = [-1000 , -800 ,-900,-950]
# left-hand side (LHS) coefficients (matrix A) for the operations table
A_coeff = [[2,3,6,5],[6,3,4,3],[5,6,2,7]]
# right-hand side (RHS) coefficients
b_coeff = [600, 300, 400]
# upper capacity
upcap = [50, 400/6, 75, 100]
r = {j : r_coeff[j-1] for j in products}
aux = {j : aux_coeff[j-1] for j in auxiliar}
A = {i : {j : A_coeff[i-1][j-1] for j in products}
    for i in operations}
b = {i : b_coeff[i-1] for i in operations}
model = Model('problem2')
x = model.addVars(products, name="x") # quantity produced
#x = model.addVars(products, name="x",vtype=GRB.INTEGER)
#now we define the binary ones:
# uncomment next lines for linear relaxation (continuous variables)
y = model.addVars(auxiliar, name="y") # quantity produced
model.addConstrs((y[j] <= 1 for j in products)) #we add
# uncomment next line for binary variables
#y = model.addVars(auxiliar, name="y", vtype=GRB.BINARY)
model.update()
# Capacity constraints
model.addConstrs((quicksum(A[i][j] * x[j] for j in products)
                           <= b[i]
                            for i in operations))
# Variable upper bound constraints
model.addConstrs((x[j] <= upcap[j-1]*y[j] for j in products))
# Objective
obj = quicksum(r[j] * x[j] for j in products)+quicksum(aux[j] * y[j] for j in products)
model.setObjective(obj, GRB.MAXIMIZE)
model.optimize()
# Display solution (print the name of each variable and the solution value)
 # Print solution
# Display solution (print the name of each variable and the solution value)
print('--------------------------------')
print('\nOptimal solution:\n')
print('Variable Information:')
for v in model.getVars():
    print("%s %s %8.2f" %
            (v.Varname, "=", v.X))
    print(" ")
print('\nOptimal objective value: %g' % model.objVal)
```

```
Solved in 4 iterations and 0.05 seconds
Optimal objective  3.655555556e+03
--------------------------------

Optimal solution:

Variable Information:
x[1] =     0.00

x[2] =    55.56

x[3] =    33.33

x[4] =     0.00

y[1] =     0.00

y[2] =     0.83

y[3] =     0.44

y[4] =     0.00

Optimal objective value: 3655.56
```

z^L=3655.56,x^L=[0,55.56,33.33,0],y^L=[0,0.83,0.44,0]

The largest limit of x2 and x3 is when x1=x4=0,the limit conditions change into:$X_2+2X_3<=200$ ①and $3X_2+4X_3<=300$②and $3X_2+X_3<=200$③,we get when $X_2=56$, $X_3<=32$; when $X_2=55$, $X_3<=33$,When $X_3=34$,$X_2<=54$,

Whenx$^{L'}$=[0,55,33,0],y$^{L'}$=[0,1,1,0]the optimal in integer.z'=2975.

Whenx$^{L''}$=[0,56,32,0],y$^{L''}$=[0,1,1,0]the optimal in integer.z''=2980.

So the optimal is 2980.

Problem 3

ⓐ $k=\lfloor \ln(2n)\rfloor = \lfloor 3.69\rfloor = 3$

$w_j = r_j = 2^{24} + 2^{3+j} + 1 \quad (j=1,2,\cdots,20)$

$b = \lfloor \sum_j w_j/2 \rfloor = \lfloor \frac{\sum_j w_j}{2} \rfloor = \lfloor \frac{20\times 2^{24} + \frac{2^4(1-2^{20})}{1-2} + 20}{2} \rfloor = 176160770$

$V_t^*(s) = \max r_{t+1} a_t + \cdots + r_n a_{n-1}$

st

$w_{t+1} a_t + \cdots + w_n a_{n-1} \le s \qquad s \in \{0, \cdots, b\} \quad t=0,\cdots, n-1$

$a_t + \cdots, a_{n-1} \geq 0$ and integer. $\qquad S_{t+1} = S_t - w_{t+1} a_t$

$V_n^*(s) = 0.$

$V_{20}^*(S_{20}) = r_{21}(S_{20}) = 0$

$V_{20}^*(S_{20}) = \max r_{20} a_{20} + V_{21}^*(S_{21}) = \max$

$0 \le a_{20} \le \lfloor \frac{S_{20}}{w_{20}} \rfloor =$