Homework 8

Kaiyang Huang, Zhenping Guo, Jingzi Liu

Variables:

x_1 : productions of product 1 at plant 1
x_2 : productions of product 2 at plant 1
x_3 : productions of product 3 at plant 1
x_4 : productions of product 1 at plant 2
x_5 : productions of product 2 at plant 2
x_6 : productions of product 3 at plant 2

Parameter

$d = [6000 \quad 8000 \quad 5000]^T$: demand for different units

$m = [10000 \quad 10000]^T$: capacity of different plants

$$\begin{aligned} \min \quad & c^T x \\ s.t. \quad & x \geq 0 \\ & A\mathbf{x} \geq b \end{aligned}$$

Where

$$c = \begin{pmatrix} 5 & 6 & 8 & 8 & 7 & 10 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 \end{pmatrix}$$

$$b = \begin{pmatrix} d \\ -m \end{pmatrix} = \begin{pmatrix} 6000 \\ 8000 \\ 5000 \\ -10000 \\ -10000 \end{pmatrix}$$

Then the dual problem is:

$$\begin{aligned} \max \quad & b^T y \\ s.t. \quad & y \geq 0 \\ & A^T y \leq c \end{aligned}$$

Where *y* are dual variables.

The annual profit is 128000.

The solution of primal problem is:

```
Presolve time: 0.00s
Presolved: 5 rows, 6 columns, 12 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    0.0000000e+00   1.900000e+04   0.000000e+00      0s
       5    1.2800000e+05   0.000000e+00   0.000000e+00      0s

Solved in 5 iterations and 0.00 seconds (0.00 work units)
Optimal objective  1.280000000e+05
Obj: 128000
solution,  reduced cost,  ranges
x[0] 6000 5.0 0.0 6.0 -2.0
x[1] 0 6.0 1.0 inf 5.0
x[2] 4000 8.0 0.0 9.0 7.0
x[3] 0 8.0 1.0 inf 7.0
x[4] 8000 7.0 0.0 8.0 0.0
x[5] 1000 10.0 0.0 11.0 9.0

Shadow prices and ranges
Primal[0] 6000.0 7.0 6000.0 7000.0 5000.0
Primal[1] 8000.0 7.0 8000.0 9000.0 -0.0
Primal[2] 5000.0 10.0 5000.0 6000.0 4000.0
Primal[3] -10000.0 2.0 -10000.0 -9000.0 -11000.0
Primal[4] -9000.0 0.0 -10000.0 -9000.0 -inf
```

The solution of dual problem is:

```
Presolve time: 0.00s
Presolved: 6 rows, 5 columns, 12 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    1.9000000e+34   6.000000e+30   1.900000e+04      0s
       5    1.2800000e+05   0.000000e+00   0.000000e+00      0s

Solved in 5 iterations and 0.00 seconds (0.00 work units)
Optimal objective  1.280000000e+05
Obj: 128000
solution,  reduced cost,  ranges
y[0] 7 6000.0 0.0 7000.0 5000.0
y[1] 7 8000.0 0.0 9000.0 -0.0
y[2] 10 5000.0 0.0 6000.0 4000.0
y[3] 2 -10000.0 0.0 -9000.0 -11000.0
y[4] 0 -10000.0 -1000.0 -9000.0 -inf

Shadow prices and ranges
Dual  [0] 5.0 6000.0 5.0 6.0 -2.0
Dual  [1] 5.0 0.0 6.0 inf 5.0
Dual  [2] 8.0 4000.0 8.0 9.0 7.0
Dual  [3] 7.0 0.0 8.0 inf 7.0
Dual  [4] 7.0 8000.0 7.0 8.0 0.0
Dual  [5] 10.0 1000.0 10.0 11.0 9.0
```

The shadow prices of primal problem constraints are identical of optimal y values:

$$A^{\text{shadow price}} = \begin{pmatrix} 7 \\ 7 \\ 10 \\ 2 \\ 0 \end{pmatrix} = y^*$$

Also, for dual problem, the shadow prices of dual problem constraints are identical of optimal x values:

$$A^{T\,\text{shadow price}} = \begin{pmatrix} 6000 \\ 0 \\ 4000 \\ 0 \\ 8000 \\ 1000 \end{pmatrix} = x^*$$

And the optimal values are identical as 12800. There is no gap between the dual and primal optimal value, as it is linear programming.

```python
Product = np.array([5,6,8,8,7,10])
# demand  = np.array([6000,8000,5000])
m       = gp.Model("HW8")
A       = np.array([[1,0,0,1,0,0],[0,1,0,0,1,0],[0,0,1,0,0,1],[-1,-1,-1,0,0,0],[0,0,0,-1,-1,-1]])
b       = np.array([6000,8000,5000,-10000,-10000])

# # #create primal variables

x       = m.addMVar((6,),lb=[0,0,0,0,0,0],name = 'x')
m.setObjective(Product@x,GRB.MINIMIZE)

m.addConstr(A@x>=b,name = 'Primal')

m.optimize()
print('Obj: %g' % m.objVal)
print('solution,  reduced cost,  ranges')
for v in m.getVars():
    print('%s %g' % (v.varName, v.x), v.obj,v.RC,v.SAObjUp,v.SAObjLow)
print('\nShadow prices and ranges')
for c in m.getConstrs():
    lhsVal = m.getRow(c).getValue()
    print(c.ConstrName,lhsVal,c.Pi,c.RHS,c.SARHSUp,c.SARHSLow)
# dual problem
#create dual variables

y       = m.addMVar((5,),lb=[0,0,0,0,0],name = 'y')
m.setObjective(b@y,GRB.MAXIMIZE)

m.addConstr(A.transpose()@y<=Product,name = 'Dual  ')

m.optimize()
print('Obj: %g' % m.objVal)
print('solution,  reduced cost,  ranges')
for v in m.getVars():
    print('%s %g' % (v.varName, v.x), v.obj,v.RC,v.SAObjUp,v.SAObjLow)
print('\nShadow prices and ranges')
for c in m.getConstrs():
    lhsVal = m.getRow(c).getValue()
    print(c.ConstrName,lhsVal,c.Pi,c.RHS,c.SARHSUp,c.SARHSLow)
```