ISYE 6669 Deterministic Optimization

Homework 5

February 20th, 2021

1.

Decision variables for our problem is the following:

i: Represents the employee number assignment

j: Represents the task number assignment

x_{ij} for $i = 1,\ldots,n$

$\quad\quad j = 1,\ldots,n$

c_{ij} for $i = 1,\ldots,n$

$\quad\quad j = 1,\ldots,n$

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij}$$

s.t. $\sum_{i=1}^{n} x_i = 1$ (Meaning each row sums to 1 for employee to task assignment)

s.t. $\sum_{j=1}^{n} x_j = 1$ (Meaning each column sums to 1 for task to employee assignment)

s.t. $\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \in [0, 1]$ Boolean observations for each x_{ij}

```
In [132]: import numpy as np
          import cvxpy as cp

          c = np.matrix([[3, 0, 0, 0], [7, 4, 4, 0], [11, 4, 10, 6], [8, 6, 9, 5]])
          x = cp.Variable(c.shape, boolean=True)
          constraints = [x>=0,x==1]
          objective = cp.Minimize(cp.sum(cp.sum(cp.multiply(c,x))))
          prob = cp.Problem(objective,constraints)

          prob.solve()
          print(prob.value)
          print(prob.status)

          77.0
          optimal

          c:\users\mjpearl\desktop\omsa\isye-6669-oan\env_isye6669\lib\site-packages\ipykernel_launcher.py:4: PendingDeprecationWarning:
          the matrix subclass is not the recommended way to represent matrices or deal with linear algebra (see https://docs.scipy.org/do
          c/numpy/user/numpy-for-matlab-users.html). Please adjust your code to use regular ndarray.
            after removing the cwd from sys.path.
```

Figure 1: CVXPY Output

2.

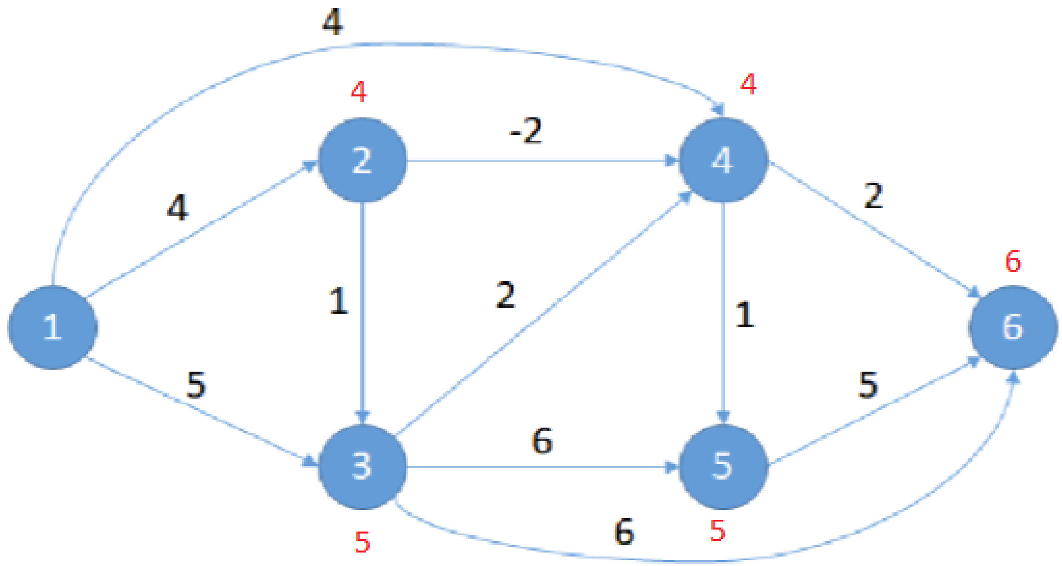

Figure 2: Shortest Path Bellman's Output

In our case the Bellman's output is in red for each corresponding node.

3.
a)

$$min\ c_1p_1 + c_2p_2 + c_3p_3 = 10p_1 + 6p_2 + 5c_3$$

$s.t$

$$-f_{61} + f_{12} = p_1$$
$$-f_{23} + f_{34} = p_2$$
$$-f_{45} + f_{56} = p_3$$
$$f_{12} - f_{23} = d_1 = 10$$
$$f_{34} - f_{45} = d_2 = 120$$
$$f_{56} - f_{61} = d_3 = 90$$

$$\begin{aligned}
f_{12} &= B_{12}(\theta_1 - \theta_2) = 11.6(\theta_1 - \theta_2)\\
f_{23} &= B_{23}(\theta_2 - \theta_3) = 5.9(\theta_2 - \theta_3)\\
f_{34} &= B_{34}(\theta_3 - \theta_4) = 13.7(\theta_3 - \theta_4)\\
f_{45} &= B_{45}(\theta_4 - \theta_5) = 9.8(\theta_4 - \theta_5)\\
f_{56} &= B_{56}(\theta_5 - \theta_6) = 5.6(\theta_5 - \theta_6)\\
f_{61} &= B_{61}(\theta_6 - \theta_1) = 10.5(\theta_6 - \theta_1)
\end{aligned}$$

$$\begin{aligned}
-f_{12}^{max} &= -100 \le f_{12} \le f_{12}^{max} = 100\\
-f_{23}^{max} &= -120 \le f_{23} \le f_{23}^{max} = 120\\
-f_{34}^{max} &= -50 \le f_{34} \le f_{34}^{max} = 50\\
-f_{45}^{max} &= -90 \le f_{45} \le f_{45}^{max} = 90\\
-f_{56}^{max} &= -60 \le f_{56} \le f_{56}^{max} = 60\\
-f_{61}^{max} &= -50 \le f_{61} \le f_{61}^{max} = 50
\end{aligned}$$

$$\begin{aligned}
p_1^{min} &= 20 \le p_1 \le p_1^{max} = 70\\
p_2^{min} &= 20 \le p_2 \le p_2^{max} = 150\\
p_3^{min} &= 10 \le p_3 \le p_3^{max} = 150
\end{aligned}$$

b-c)

Optimal value is in first part of the image, last cell contains the values at Nodes 2, 4 and 6.

```
Optimal Value: 1249.9999997631494
With the following variables: 19.999999928688347 50.00000011972041 149.99999995158868 59.85392656833289 63.21238250844543 71.51
03636788361 71.43431176537509 59.7820689114058 54.238870488904865 -38.95808890530549 -48.958088905305054 1.041911214415802 -11
8.95808878558378 31.041911166005285 -58.95808883399427
```

```
In [129]: print(constraints[15].dual_value)
          print(constraints[16].dual_value)
          print(constraints[17].dual_value)

-5.9999999735722565
-5.999999975933699
-5.999999973803812
```