IE533 Homework 1

January 24, 2023

NAME: Duo Zhou

NETID: duozhou2

1 Question I

1.1 Basic Question

Company ZZ wants to assign its staff members efficiently among the jobs and wants to utilize their time. Based on people skills and job requirements, we have a utility of u_{ij} for assigning person i to job j. The manager of ZZ wants to assign people to jobs to maximize overall utility. How would you formulate this problem as a network flow problem?

1.1.1 Solution

Introduce the variable $x_{ij} \in \{0, 1\}^{n \times m}$, which indicate:

$$x_{ij} = \begin{cases} 1, & \textit{Assign the } i^{th} \textit{ person to complete the } j^{th} \textit{ job} \\ 0, & \textit{Otherwise} \end{cases}$$

Thus the mathematical model of the assignment problem is:

$$\begin{aligned} max \ & \sum_{i=1}^{n} \sum_{j=1}^{m} u_{ij} x_{ij} \\ s.t. & \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, ..., n \\ & \sum_{i=1}^{m} x_{ij} = 1, \quad j = 1, ..., m \\ & x_{ij} \in \{0, 1\} \end{aligned}$$

1.2 Bonus Question

Now assume that you have job assignments to be made for multiple time periods. How would you change your formulation to meet this new situation while still attempting to maximize the overall utility.

1.2.1 Solution

It can be assumed that each persion i could be assigned to each job j with a time cost c_{ij}, the total time of assigned jobs is at most t_i.

Introduce the variable $x_{ij} \in \{0,1\}^{n\times m}$ as well, which indicate:

$$x_{ij} = \begin{cases} 1, & Assign\ the\ i^{th}\ person\ to\ complete\ the\ j^{th}\ job \\ 0, & Otherwise \end{cases}$$

Thus the mathematical model of the assignment problem made for multiple time periods is:

$$\begin{aligned} max\ & \sum_{i=1}^{n}\sum_{j=1}^{m} u_{ij}x_{ij} \\ s.t. & \sum_{j=1}^{m} c_{ij}x_{ij} \le t_i, \quad i = 1, ..., n \\ & \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, ..., m \\ & x_{ij} \in \{0,1\} \end{aligned}$$

2 Question II

2.1 Basic Question

A global supply chain company has n factories (indexed by i) and m warehouses (indexed by j). Each factory has a certain supply available and each warehouse as a certain demand requirement. Assume that the total supply is greater than the total demand for sake of simplicity. The cost to ship one unit of item from factory i to warehouse j is c_{ij}. The supply chain manager wants to minimize total shipment cost. Formulate the problem as a network flow model.

2.1.1 Solution

Assumed that supply of factory i is s_i, demand of warehouse j is d_j,

Introduce the variable $x_{ij} \in N^{n\times m}$, which indicate the quantity of products shiped from the i^{th} factory to the j^{th} warehouse.

Thus the mathematical model of the transportation problem is:

$$
\begin{aligned}
min & \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \\
s.t. & \sum_{j=1}^{m} x_{ij} \leq s_i, \quad i = 1, ..., n \\
& \sum_{i=1}^{n} x_{ij} \leq d_j, \quad j = 1, ..., m \\
& x_{ij} \geq 0
\end{aligned}
$$

2.2 Bonus

If there are multiple parts indexed by k and the unit shipping costs are c_{ijk}, and you have a unit size of d_{ijk}, you want to limit the shipments from each warehouse to a (say trucksize) D. Revise your above model.

2.2.1 Solution

Based on the above fomula, we can add a maximum shipment size constraint states as the total size of the shipment(supply quantity $\times$ item size) from each warehouse should not exceed the maximum shipment size D .

This can be expressed mathematically as:

$$
\begin{aligned}
min & \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{p} c_{ijk}x_{ijk} \\
s.t. & \sum_{j=1}^{m}\sum_{k=1}^{p} x_{ijk} \leq s_i, \quad i = 1, ..., n \\
& \sum_{i=1}^{n}\sum_{k=1}^{p} x_{ijk} \leq d_j, \quad j = 1, ..., m \\
& \sum_{j=1}^{m}\sum_{k=1}^{p} d_{ijk}x_{ijk} \leq D, \quad i = 1, ..., n \\
& x_{ijk} \geq 0
\end{aligned}
$$

3 Question III

Paper and wood products companies need to define cutting schedules that will maximize the total wood yield of their forests over some planning period. Suppose that a company with control of p forest units wants to identify the best cutting schedule over a planning horizon of k years. Forest unit i has a total acreage of a_j units, and studies that the company has undertaken predict that this unit will have w_{ij} tons of woods available for harvesting in the year j. Based on its prediction of economic conditions, the company believes that it should harvest at least I_j tons of wood in year j. Due to the availability of equipment and personnel, the company can harvest at most U_j tons of wood in year j. Formulate the problem of determining a schedule with maximum wood yield as a network flow problem.

3.0.1 Solution

Introduce the variable $x_{ij} \in N^{n \times m}$, which indicate the quantity in tons of woods cut from the i^{th} unit forest in the j^{th} year.

So, the problem could be seemed as a IP problem where the decision variables are x_{ij} and the constraints are linear. The objective is to maximize the total wood yield by finding the optimal schedule for harvesting wood from the forest units over the planning horizon of k years, subject to the constraints of available wood, minimum and maximum harvest, forest unit acreage and non-negativity as:

$$
\begin{aligned}
max \sum_{i=1}^{p} \sum_{j=1}^{k} x_{ij} \\
s.t. \sum_{j=1}^{k} x_{ij} \leq a_j, \quad i = 1, 2, \ldots, p \\
\sum_{i=1}^{p} x_{ij} \geq I_j, \quad j = 1, 2, \ldots, k \\
\sum_{i=1}^{p} x_{ij} \leq U_j, \quad j = 1, 2, \ldots, k \\
x_{ij} \leq w_{ij}, \quad i = 1, 2, \ldots, p, \quad j = 1, 2, \ldots, k \\
x_{ij} \geq 0
\end{aligned}
$$

4 Question IV

For a network $G = (V, E)$ with source node s and terminal node t, let $c_{ij} \geq 0$ be the arc capacity for arc $(i, j) \in E$.

4.1 Write down the LP formulation for the max-flow problem.

4.1.1 Solution

Set a source node $s \in V$, a sink node $t \in V$

Introduce the variable x_{ij} represents the flow on arc (i, j) in the network. Then we can build a standrad LP max-flow problem formula as:

$$
\begin{aligned}
max \sum_{j \in V} x_{ij} \\
s.t. \sum_{j \in V} x_{ij} = \sum_{j \in V} x_{ji} \quad \forall i \in V \setminus s, t \\
x_{ij} \leq c_{ij} \quad \forall (i, j) \in E \\
x_{ij} \geq 0 \quad \forall (i, j) \in E
\end{aligned}
$$

4.2 Write the LP dual of the formulation.

4.2.1 Solution

Introduce variables: $y_{i,j}$ represents the potential of arc (i,j) in the network.

Objective function: Minimize the sum of the potentials of all nodes in the cut set.

This can be expressed mathematically as:

$$
\begin{aligned}
min \sum_{(i,j)\in E} & c_{i,j} y_{i,j} \\
s.t. \sum_{(i,j)\in E} & y_{i,j} - \sum_{(j,i)\in E} y_{j,i} = 0, \quad \forall i \in V \setminus s, t \\
& \sum y_{ij} \leq 1 \quad \forall i \in V \setminus s \\
& y_{i,j} \geq 0 \quad \forall (i,j) \in E
\end{aligned}
$$

4.3 In your own words, provide an intuition behind the dual problem.

4.3.1 Solution

The intuition behind the LP dual of the max-flow problem is that it provides a way to understand the capacity constraints of the network from a different perspective. The primal LP formulation of the max-flow problem is focused on finding the maximum flow that can be sent from the source to the terminal node, subject to the capacity constraints of the arcs. In contrast, the LP dual of the max-flow problem, also known as the "flow-interdiction LP", is focused on finding the minimum amount of flow that can be blocked in order to disconnect the source and the terminal nodes.

The dual problem's decision variables, y_i, represent the flow on each node in the network, while the slack variables, z_{ij}, represent the amount by which the flow on each arc exceeds its capacity constraint. The objective function of the dual problem is to minimize the total flow from the source to the terminal node, which is equivalent to finding the minimum cut of the network. In this way, the LP dual problem provides an alternative way of understanding the capacity constraints of the network.

5 Question V

The figure below shows an instance of the multi-commodity flow problem. The network has 2 commodities, and a source and sink node for each commodity. There are 6 transshipment nodes. The arc costs are given alongside. In all but one arc (from node 2 to node 5), assume that the capacity is infinity.

5.1 Formulate the multi-commodity flow problem as an LP.

5.1.1 Solution

Introduce $f(i,j)$ denotes the flow of arc (i,j).

$$
\begin{aligned}
min f_{12} + 5f_{14} + f_{25} + f_{32} + 6f_{36} + f_{54} + f_{56} \\
s.t. f_{12} + f_{14} = 5 \\
f_{32} + f_{36} = 2 \\
f_{14} + f_{54} = 5 \\
f_{36} + f_{56} = 2 \\
f_{12} + f_{32} = f_{25} \\
f_{25} = f_{54} + f_{56} \\
f_{25} \leq 5 \\
f_{ij} \geq 0, \forall (i, j) \in E
\end{aligned}
$$

5.2 Find the optimal solution (using any method – by hand/a computer program).

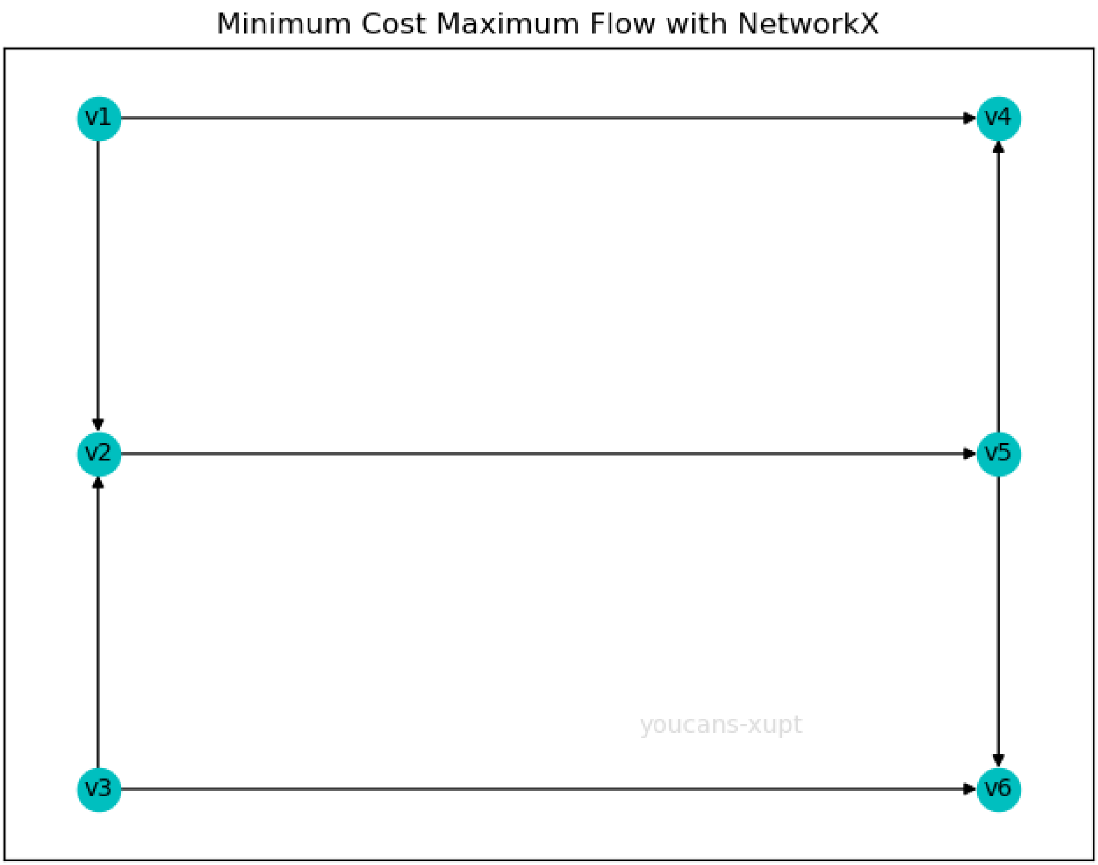
5.2.1 Solution

Solve by networkx

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import networkx as nx
G1 = nx.DiGraph()
G1.add_edges_from([('v1','v2',{'weight': 1}),
                   ('v1','v4',{'weight': 5}),
                   ('v2','v5',{'capacity': 5, 'weight': 1}),
                   ('v3','v2',{'weight': 1}),
                   ('v3','v6',{'weight': 6}),
                   ('v5','v4',{'weight': 1}),
                   ('v5','v6',{'weight': 1})])

# nx.min_cost_flow()
G1.add_node("v1", demand=-5)  # sources
G1.add_node("v3", demand=-2)
G1.add_node("v4", demand=5)  # sinks
G1.add_node("v6", demand=2)
```

[2]:
```
pos={'v2':(0,5),'v3':(0,2),'v1':(0,8),'v6':(10,2),'v4':(10,8),'v5':(10,5)}  #␣
↪position
fig, ax = plt.subplots(figsize=(8,6))
ax.text(6,2.5,"youcans-xupt",color='gainsboro')
ax.set_title("Minimum Cost Maximum Flow with NetworkX")
nx.draw(G1,pos,with_labels=True,node_color='c',node_size=300,font_size=10)
plt.axis('on')
plt.show()
```




```
[3]: # Min Cost
minFlowCost = nx.min_cost_flow_cost(G1)
minFlowDict = nx.min_cost_flow(G1)

# Trasfer Data Stucture
edgeCapacity = nx.get_edge_attributes(G1, 'weight')
edgeLabel = {}
# Sort Labels of Edge
for i in edgeCapacity.keys():
    edgeLabel[i] = f'w={edgeCapacity[i]:}'
edgeLists = []

for i in minFlowDict.keys():
    for j in minFlowDict[i].keys():
        edgeLabel[(i,j)] += ',f=' + str(minFlowDict[i][j])
        if minFlowDict[i][j] > 0:
            edgeLists.append((i, j))

print("Minimum Cost:{}".format(minFlowCost))
print("Path and flow of the minimum cost flow:", minFlowDict)
```

```python
print("Path", edgeLists)
```

```
Minimum Cost:25
Path and flow of the minimum cost flow: {'v1': {'v2': 3, 'v4': 2}, 'v2': {'v5':
5}, 'v4': {}, 'v5': {'v4': 3, 'v6': 2}, 'v3': {'v2': 2, 'v6': 0}, 'v6': {}}
Path [('v1', 'v2'), ('v1', 'v4'), ('v2', 'v5'), ('v5', 'v4'), ('v5', 'v6'),
('v3', 'v2')]
```

[4]:

```python
fig, ax = plt.subplots(figsize=(8,6))
ax.set_title("Capacity network with multi source and multi sink")
nx.draw(G1,pos,with_labels=True,node_color='skyblue',node_size=200,font_size=10)
edgeLabel1 = nx.get_edge_attributes(G1, 'weight')
nx.draw_networkx_nodes(G1, pos, nodelist=['v1','v3'], node_color='orange') ␣
 ↪#sources
nx.draw_networkx_nodes(G1, pos, nodelist=['v4','v6'], node_color='c')      ␣
 ↪#sinks
nx.draw_networkx_edge_labels(G1,pos,edgeLabel,font_size=20,label_pos=0.5)
nx.draw_networkx_edges(G1,pos,edgelist=edgeLists,edge_color='m',width=1)
plt.axis('on')
plt.show()
```