OR - HAND INS 3

Cemal Arican, i6081025
Yifeng Han, i6127017

March 2020

1 Exercise 8.16: reformulation of a Minimum Cost Flow problem

There is a container ship that has r containers on board, it goes to visit each port chronologically, mean it visits port 1 first, then port 2, up to port n. The income earned by shipping each container is p_{ij}, where $j > i$. In addition, there is a maximum amount of d_{ij} at each port to be transported from port i to port j. The goal here is maximize income for the cargo company, and formulate it as a minimum cost flow problem.

In this case we have to consider that the carrier has to pick up load between ports, if he chooses so. This means that we should consider extra node as storage node, for when decides to pick up shipment. We show an example below:

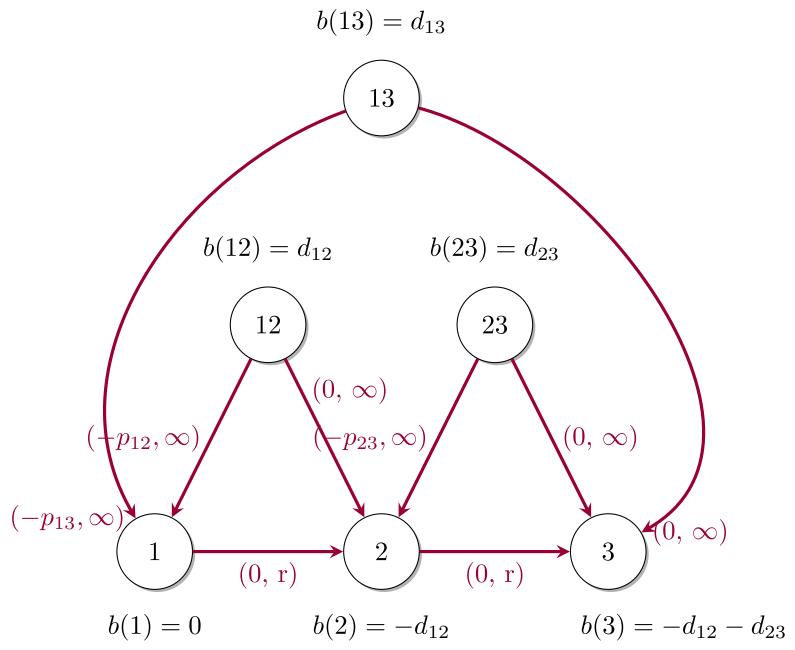

Figure 1: Example of a graph for this problem

Our graph will consists of the following vertices, all $v_i, i \in \{1, ..., n\}$ which are the ports, and we also have to add the these 'storage vertices' which create a total of $\frac{n(1+n)}{2}$ number of vertices. Coming to the arcs, we have the direct arcs from port to port, which there are $n-1$ of. Then the edges connecting all these 'storage' vertices to the main ports also creates an additional $n(n-1)$ number of arcs.

So, the set of vertices becomes: $V = \{v_1, ..., v_n\} \cup \{v_{ij} | 1 \leq i < j \leq n\}$. Then set of arcs: $A = \{v_1v_2, v_2v_3, ..., v_{n-1}v_n\} \cup \{v_{ij}v_i, v_{ij}v_j | 1 \leq i < j \leq n\}$, which make up the graph $G = (V, A)$.

The capacities, the arcs that go port to port directly i.e. $v_iv_{i-1}, i \in \{1, ..., n-1\}$ have capacity r, all others have capacity ∞

The arc costs: the arcs $v_{ij}v_i$ all have costs of $-p_{ij}$ and all others have costs of zero.

Then finally we have the supple and demand nodes, $b(v_i) = -\sum_{j<i} d_{ij}$ and $b(v_{ij}) = d_{ij}$, and $b(1) = 0$ is a transhipment node.

Let x_{ij} be the flow send through the graph to minimise cost, where $(i, j) \in A$. Hence, we have as main objective: min $\sum_{ij} -p_{ij}x_{ij}$

Exercise 7.12
Initilization:
For a graph G =(V,E), c_e is the capacity for arc e $\in E, s, t \in V. |E|$ is the number of edges in G. x_i denotes the capacity of the minimum cuts [A,B]$_i, i \in \{1, ..., n\}$. x'_i is the capacity of the cut in the graph after transformation, m_i is the number of crossing arcs in the graph after transformation.

Reasoning:
The Ford-Fulkerson Algorithm can help us to find the maximum flow, the value of the minimum cut is the same as the value of the maximum flow. Let x'_i=x_i($|E|$+1)+m_i. Our goal is to find the least m_i, now we are turning this question into finding the least x'_i.

For different minimum cuts of a graph, x_1 =x_2 =...= x_n. Let a,b $\in \{1, ..., n\}$.
If $m_a > m_b$, then x'_a= x_a($|E|$+1)+m_a

$$\begin{aligned} &= x_b(|E|+1)+m_a \\ &> x_b(|E|+1)+m_b = x'_b \end{aligned}$$

So $m_a > m_b \Longrightarrow$x$'_a > x'_b$. Therefore, once we know the least-capacity-cut, we know the cut with the least number of arcs among all minimum s-t arcs.

Algorithm:
Step 1: Use the Ford-Fulkerson Algorithm to find the maximum flow
Step 2: Find minimum cuts $[A, B]_i$ with x_i
Step 3: Transform the graph, let c'_e= c_e($|E|$+1)+1
Step 4: Return the minimum cut which has the least capacity