1 Outline

In this lecture, we cover

- the minimum cost flow problem with outflow and inflow capacities,
- the shortest path problem.

2 Minimum cost flow with outflow and inflow capacities

In some applications, we have additional constraints such as imposing capacities of the total outflow from a node and the inflow from a node. An outflow capacity constraint can be imposed by

$$\text{outflow}(i;x) = \sum_{j\in N:(i,j)\in A} x_{ij} \le \alpha_i.$$

Similarly, an inflow capacity constraint can be imposed by

$$\text{inflow}(i;x) = \sum_{k\in N:(k,i)\in A} x_{ki} \le \beta_i.$$

Let $N_1 \subseteq N$ be the set of nodes that have an outflow capacity, and let $N_2 \subseteq N$ be the set of nodes that have an inflow capacity.

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} \\
\text{s.t.} \quad & \sum_{j\in N:(i,j)\in A} x_{ij} - \sum_{k\in N:(k,i)\in A} x_{ki} = b_i, \quad i \in N \\
& \sum_{j\in N:(i,j)\in A} x_{ij} \le \alpha_i, \quad i \in N_1 \\
& \sum_{k\in N:(k,i)\in A} x_{ki} \le \beta_i, \quad i \in N_2 \\
& \ell \le x \le u.
\end{aligned}$$

How do we solve this model? Do we directly tackle this linear program? In fact, there is a simple technique to reduce the problem to the minimum cost flow problem without outflow/inflow capacities.

The idea is **splitting nodes**. Suppose that we have a node i with outflow capacity α_i. We have node i that has arcs into it and arcs out of it as in Figure 13.1. Then we split node i into two nodes i_1 and i_2 as in Figure 13.2. Here, splitting is done by the following procedure.

- We replace node i by two nodes i_1 and i_2.

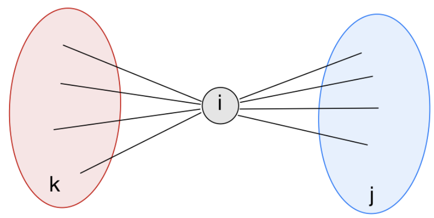

Figure 13.1: Before splitting node i

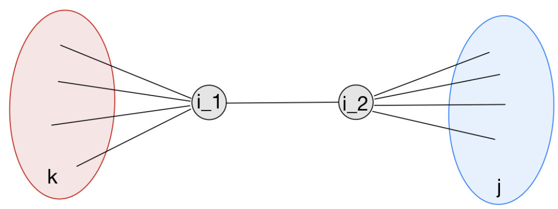

Figure 13.2: After splitting node i

- We replace the set of incoming arcs $\{(k,i) \in A : \ k \in N\}$ by
$$\{(k, i_1) : \ k \in N, \ (k,i) \in A\}.$$
- We replace the set of outgoing arcs $\{(i,j) \in A : \ j \in N\}$ by
$$\{(i_2, j) : \ j \in N, \ (i,j) \in A\}.$$
- We set $b_{i_1} = b_i$ and $b_{i_2} = 0$ for the case when i has an outflow capacity.
- We set the flow upper bound on the new arc (i_1, i_2) to $u_{i_1 i_2} = \alpha_i$.

After this splitting procedure, we end up with

$$\begin{aligned} x_{i_1 i_2} - \sum_{k \in N : (k, i_1) \in A} x_{k i_1} &= b_i, \\ x_{i_1 i_2} &\leq \alpha_i, \\ \sum_{j \in N : (i_2, j) \in A} x_{i_2 j} - x_{i_1 i_2} &= 0. \end{aligned}$$

Adding the first and the third equalities, we obtain

$$\sum_{j \in N : (i_2, j) \in A} x_{i_2 j} - \sum_{k \in N : (k, i_1) \in A} x_{k i_1} = b_i,$$

which is equivalent to the flow balance constraint imposed on node i. Adding the second and the third constraints, we obtain

$$\sum_{j \in N : (i_2, j) \in A} x_{i_2 j} \leq \alpha_i,$$

which is equivalent to the outflow capacity capacity on node i.

What about inflow capacity constraints? We repeat the same splitting procedure, but we set the net supply values and the flow upper bound differently as follows.

- We set $b_{i_1} = 0$ and $b_{i_2} = b_i$ for the case when i has an inflow capacity.
- We set the flow upper bound on the new arc (i_1, i_2) to $u_{i_1 i_2} = \beta_i$.

Then we deduce

$$\begin{aligned} x_{i_1 i_2} - \sum_{k \in N:(k,i_1) \in A} x_{k i_1} &= 0, \\ x_{i_1 i_2} &\leq \beta_i, \\ \sum_{j \in N:(i_2,j) \in A} x_{i_2 j} - x_{i_1 i_2} &= b_i. \end{aligned}$$

Again, adding the first and the third equalities, we obtain

$$\sum_{j \in N:(i_2,j) \in A} x_{i_2 j} - \sum_{k \in N:(k,i_1) \in A} x_{k i_1} = b_i.$$

However, subtracting the first equality from the second inequality, we deduce that

$$\sum_{k \in N:(k,i_1) \in A} x_{k i_1} \leq \beta_i.$$

3 Logistics example with outflow capacities

Imagine a manufacturing firm that has two production plants based in Dallas and Houston.

- The plants in Dallas and Houston have production limits.
- Customers are based in NYC and SF, and products are shipped by air.
- Sometimes it is cheaper to fly via Chicago or LA, but these routes have limited capacities.

The following table shows the production capacities of the plants in Dallas and Houston, the routing capacities of the airports in Chicago and LA, and the customer demands of NYC and SF.

Production capacities		Routing capacities		Customer demands	
Dallas	Houston	Chicago	Los Angeles	San Francisco	New York City
200	160	90	80	140	140

The next table shows transportation costs between cities.

c_{ij}	Chicago	LA	SF	NYC
Dallas	11	10	26	29
Houston	9	12	27	26
Chicago		7	12	16
LA	7		13	15

The logistics network is given as in Figure 13.3.

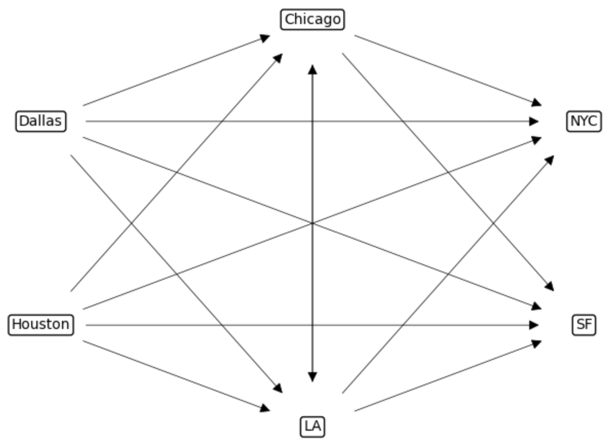

Figure 13.3: The second logistics network

Demands for SF and NYC are 140 each. Note that SF and NYC do not have any outflow, we may add

$$\text{net-supply}(\text{SF}; x) = -\text{inflow}(\text{SF}; x) \leq -140$$
$$\text{net-supply}(\text{NYC}; x) = -\text{inflow}(\text{NYC}; x) \leq -140$$

Dallas and Houston have production capacities. The production capacity of Dallas (Houston) upper bounds the outflow of Dallas (Houston). Nevertheless, Dallas and Houston do not have any inflow. Hence, the production capacities can be encoded by constrains on the net supply values.

$$\text{net-supply}(\text{Dallas}; x) = \text{outflow}(\text{Dallas}; x) \leq 200$$
$$\text{net-supply}(\text{Houston}; x) = \text{outflow}(\text{Houston}; x) \leq 160$$

The routing capacity of Chicago (Los Angeles) upper bounds the outflow of Chicago (Los Angeles). Unlike Dallas and Houston, Chicago and Los Angeles have both their outflows and inflows. Moreover, Chicago and Los Angeles do not have customer demands. Hence, we have to add two types of constraints. First, upper bounds on outflow capacities.

$$\text{outflow}(\text{Chicago}; x) \leq 90$$
$$\text{outflow}(\text{Los Angeles}; x) \leq 80$$

Next, the net supplies of the two cities are 0.

$$\text{net-supply}(\text{Chicago}; x) = \text{outflow}(\text{Chicago}; x) - \text{inflow}(\text{Chicago}; x) = 0$$
$$\text{net-supply}(\text{Los Angeles}; x) = \text{outflow}(\text{Los Angeles}; x) - \text{inflow}(\text{Los Angeles}; x) = 0$$

As Chicago and Los Angeles have outflow capacities, we apply the splitting technique. After splitting, we have Chicago1, Chicago2 and LA1, LA2.

Figure 13.5 shows the optimal flow values for the network. We can see that Dallas and Houston send out 120 and 160 units in total, obeying the production capacities. Note that Houston's production reaches its capacity. SF and NYC both have a demand of 140 units, and each of them receives 140 units. Note that the outflow of Chicago and that of LA are at their out flow capacities.

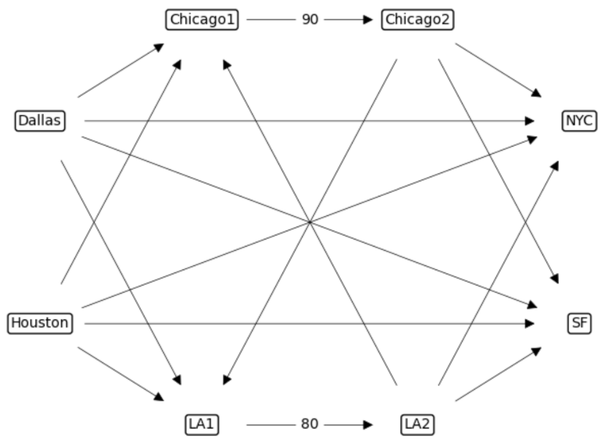

Figure 13.4: Logistics network after splitting

Variables						
	Chicago1	Chicago2	Los Angeles1	Los Angeles2	San Francisco	New York City
Dallas	0	0	80	0	40	0
Houston	90	0	0	0	0	70
Chicago1	0	90	0	0	0	0
Chicago2	0	0	0	0	90	0
Los Angeles1	0	0	0	80	0	0
Los Angeles2	0	0	0	0	10	70
Flow values						
	Chicago	Los Angeles	San Francisco	New York City		
Dallas	0	80	40	0		
Houston	90	0	0	70		
Chicago		0	90	0		
Los Angeles	0		10	70		
Objectives						
Minimize	6730					

Figure 13.5: Solution obtained by Excel's Solver

4 Shortest path problem

Given a directed graph $D = (N, A)$ and two distinct nodes $s, t \in N$, a **(directed)** st**-path** is a sequence of nodes $v_0,\ v_1,\ \ldots,\ v_\ell$ such that

- $v_0 = s$ and $v_\ell = t$,
- $v_0, \ldots, v_\ell$ are distinct nodes,
- $(v_{i-1}, v_i) \in A$ for $i = 1, \ldots, \ell$.

Here we often call s the **origin node** and t the **destination node**. We can define an st-path with arcs. A directed st-path can be defined as a sequence of arcs $a_1,\ \ldots,\ a_\ell$ such that

- $a_i = (v_{i-1}, v_i)$ for $i = 1, \ldots, \ell$ for some nodes $v_0, \ldots, v_\ell$,
- $v_0 = s$ and $v_\ell = t$,

- $v_0, \ldots, v_\ell$ are distinct nodes.

In general, a (directed) path is an st-path where s and t are the first and the last nodes in the path. Let c_{uv} be the length of arc $(u,v) \in A$. Then the length of a path P is

$$\sum_{(u,v)\in P} c_{uv}$$

where $(u,v) \in P$ means that arc (u,v) is on the path P. Now the problem is to find a shortest st-path, that is, a directed st-path of the minimum length.

We will show that the problem of finding a shortest st-path can be posed as an instance of the minimum cost flow problem. Let $x_{uv} \in \{0,1\}$ denote the variable for arc (u,v) to indicate whether

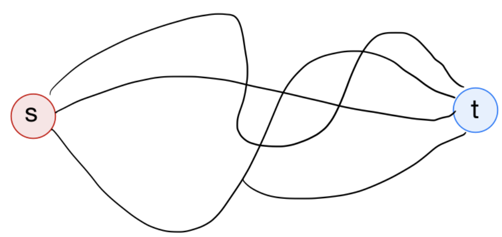

Figure 13.6: st-path as a unit flow

arc $(u,v) \in A$ is chosen to be part of my path. Then we may look at $x \in \{0,1\}^A$ whose components correspond to the arc set A. If x corresponds to the arc set of an st-path, then

$$\sum_{(u,v)\in A} c_{uv} x_{uv}$$

would be the length of the path.

When does a 0,1 vector $x \in \{0,1\}^A$ correspond to an st-path? Observe the following.

- The origin node s has an outgoing arc on the path. No other arc of the path is incident to s. We may model this as
$$\sum_{j\in N:(s,j)\in A} x_{sj} - \sum_{k\in N:(k,s)\in A} x_{ks} = 1.$$

- The destination node t has an incoming arc on the path. No other arc of the path is incident to t. We may model this as
$$\sum_{j\in N:(t,j)\in A} x_{tj} - \sum_{k\in N:(k,t)\in A} x_{kt} = -1.$$

- Let $i \in N \setminus \{s,t\}$. If i is on the path, then i has an incoming arc and an outgoing arc on the path. No other arc is incident to i. If i is not on the path, then no arc of the path is incident to i. This implies that the number of arcs going into i and the number of arcs going out of i are the same. This can modeled as
$$\sum_{j\in N:(i,j)\in A} x_{ij} - \sum_{k\in N:(k,i)\in A} x_{ki} = 0.$$

Therefore, an st-path can be viewed as the source node s sending one unit of flow to the sink node t. More precisely, the origin node s has supply 1, and the destination node t has demand 1. The other nodes have 0 net supply, meaning that they are transhipment nodes. Then the problem can be formulated as

$$
\begin{aligned}
\min \quad & \sum_{(u,v)\in A} c_{uv}x_{uv} \\
\text{s.t.} \quad & \sum_{j\in N:(s,j)\in A} x_{sj} - \sum_{k\in N:(k,s)\in A} x_{ks} = 1 \\
& \sum_{j\in N:(t,j)\in A} x_{tj} - \sum_{k\in N:(k,t)\in A} x_{kt} = -1 \\
& \sum_{j\in N:(i,j)\in A} x_{ij} - \sum_{k\in N:(k,i)\in A} x_{ki} = 0, \quad \forall i \in N \setminus \{s,t\} \\
& x_{ij} \geq 0, \quad \forall (i,j) \in A
\end{aligned}
$$

The formulation is an instance of the minimum cost flow formulation. Therefore, solving this linear program will return a solution x^* that has integer entries only, which corresponds to a shortest st-path.

A **directed cycle** is a sequence of nodes $v_0,\ v_1,\ \ldots,\ v_\ell$ such that

- $v_0, \ldots, v_{\ell-1}$ are distinct nodes,
- $v_\ell = v_0$.

Remark 13.1. If D contains a directed cycle of negative length, then the linear program is unbounded. If D contains no directed cycle of negative length, then the linear program would have an optimal solution.