1 Outline

In this lecture, we cover

- introduction to network models,
- a logistics example,
- the minimum cost flow model.

2 What is a network model?

Many real-world problems are intimately related to **networks**. For example,

- transportation and logistics networks,
- electricity networks,
- social networks,
- financial networks.

Some linear programming models can be represented graphically via a network. Converseley, problems on networks can be modeled as linear programs.

A **network** is a **directed graph** $D = (N, A)$ where N is a set of **nodes** and $A \subseteq N \times N$ is a set of **arcs**. Here, an arc is an ordered pair of two nodes. Figure 11.1 shows a network over 6 nodes.

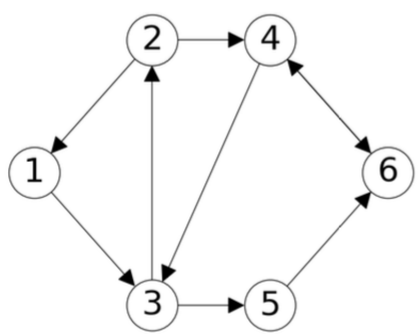

Figure 11.1: Network over 6 nodes

How many arcs are there? Note that nodes 4 and 6 are connected in both ways. Hence, there are 9 arcs in total. More precisely,

$$N = \{1, 2, 3, 4, 5, 6\} \quad \text{and} \quad A = \{(1,3), (2,1), (2,4), (3,2), (3,5), (4,3), (4,6), (5,6), (6,4)\}.$$

Nodes represent **entities** such as locations in a physical road network, people in social networks, and computers on the internet. Moreover, arcs represent **directed connections** between entities.

For example, roads between two locations, follower relationships on instagram, and connections between two computers.

For some applications, the direction of a connection is not as important. For such applications, we may use **undirected graph** $G = (V, E)$ where V is a set of **vertices** and E is a set of **edges**. Here, vertices correspond to nodes, and edges correspond to arcs.

3 Logistics example

We want to transport goods from **warehouses** to **customer locations**. Some customer locations are **hubs**, meaning that they can tranship goods that they receive.

- **Warehouses**: Phoenix, Austin, Gainesville.
- **Hubs**: Dallas, Atlanta.
- **Customer locations**: Chicago, Los Angeles, New York.

The logistics network over the cities is illustrated in Figure 11.2. Note that warehouses and hubs

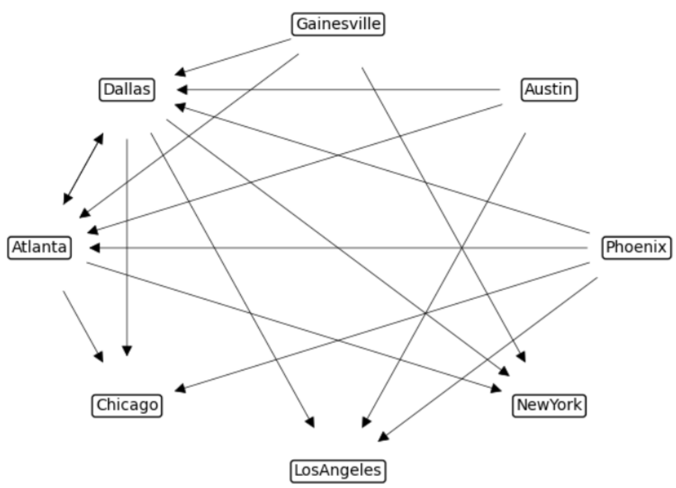

Figure 11.2: Logistics network over US cities

do not necessarily ship to all cities. This may be due to restrictions on available transport routes.

Transhipment incurs cost. The following table shows the cost of transhipping one unit of the good from a ocation to another.

c_{ij} : cost from i to j	Dallas	Atlanta	Chicago	Los Angeles	New York
Phoenix	3	7	6	3	
Austin	2	5		7	
Gainesville	6	4			2
Dallas		2	4	5	6
Atlanta	2		4		5

c_{ij} represents the transhipment cost from the city at row i and the city at column j.

The hubs and the customer locations have certain demands for the product. The demands of the cities are summarized in the following table.

Dallas	Atlanta	Chicago	Los Angeles	New York
300	150	200	200	250

Note that the demans sum up to 1100.

Warehouses have some stock of goods that are supposed to be sent out to the hubs and the customer locations. Hence, the stock of each warehouse corresponds to supplies.

Phoenix	Austin	Gainesville
700	200	200

The supplies add up to 1100 as well.

We may solve this problem by a linear programming model.

Decisions: We use variable x_{ij} to denote the number of units to send from city i and city j.

Constraints : First, we have $x_{ij} \geq 0$. Next, we need to impose that the demands of hubs and customer locations are satisfied. To represent the demand satisfaction constraints, note that

$$\sum_{k:(k,i)\in A} x_{ki}$$

is the total amount of goods shipped into city i. Moreover,

$$\sum_{j:(i,j)\in A} x_{ij}$$

is the total amount of goods shipped out of city i. Then we impose

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{k:(k,i)\in A} x_{ki} = b_i$$

for each city $i \in N$. Here, b_i is the **net supply** of city i where

- b_i is positive, indicating a supply, if city i is a warehouse,
- b_i is negative, indicating a demand, if city i is a hub or a customer location.

Objective: The objective is to minimize the total transhipment cost, given by

$$\sum_{(i,j)\in A} c_{ij}x_{ij}$$

where c_{ij} is the cost from city i to city j.

We may solve this linear program using Solver, an Excel add-in program.

1. Download Excel and its add-in application Solver.
2. Write down your linear programming model (variables, objective, constraints).
3. Solve the linear program with the Simplex method implemented withint Solver.

	A	B	C	D	E	F
1	Variables					
2		Dallas	Atlanta	Chicago	Los Angeles	New York
3	Phoenix	300	0	200	200	0
4	Austin	200	0	0	0	0
5	Gainesville	0	0	0	0	200
6	Dallas	0	150	0	0	50
7	Atlanta	0	0	0	0	0
8						
9						
10	Objectives					
11	Minimize	4100				
12						
13	Constraints (supplies)					
14	Phoenix	700	=	700		
15	Austin	200	=	200		
16	Gainesville	200	=	200		
17						
18	Constraints (demands)					
19	Dallas	-300	=	-300		
20	Atlanta	-150	=	-150		
21	Chicago	-200	=	-200		
22	Los Angeles	-200	=	-200		
23	New York	-250	=	-250		

Figure 11.3: Solving the linear programming model of the logistics example using Excel

Figure 11.3 shows the result from solving the linear programming model for the logistics example. To summarize the optimal transhipment decision minimizing the total transportation cost, we have

	Dallas	Atlanta	Chicago	Los Angeles	New York
Phoenix	300	0	200	200	
Austin	200	0		0	
Gainesville	0	0			200
Dallas		150	0	0	50
Atlanta	0		0		0

Visualizing this result, we have

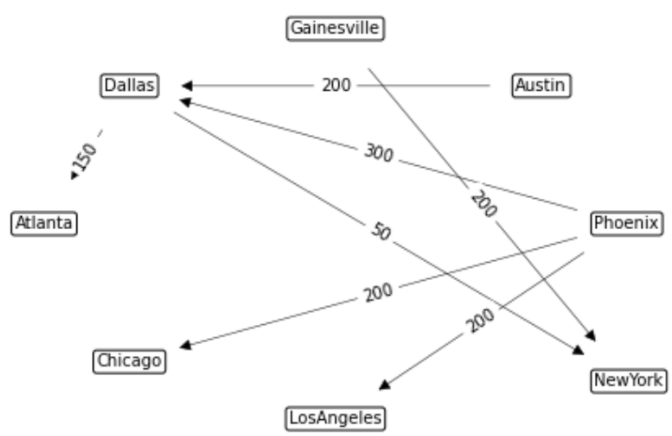

Figure 11.4: Optimal transhipment amounts

Note that the optimal transhipment amounts, encoded by the optimal decision vector x, are all integers. In fact, this is **not a coincidence**, it is one of the key properties of network flow models. If **supplies and demands are all integers**, then **the optimal solution will consists of integer**

values. Hence, if we are able to formulate our problem with a network flow model, then this is good because we do not need to worry about integer constraints as long as our data are integers.

In fact, there are further advantages with network models. Special network flow models admit **faster solution methods, such as algorithms**, than the plain simplex method.

4 Minimum cost flow model

One of the most general network flow models is the **minimum cost flow model**. Here, think of **flow** as some quantity, such as water, electricity, money, and products, that needs to be routed around the network. The minimum cost transhipment problem over the small logistics network is an instance of the minimum cost flow problem. Given a network $D = (N, A)$, we define the following components of the problem.

Decisions: We use variable x_{ij} for each arc $(i, j) \in A$ to decide how much flow travels across arc (i, j).

Flow bound constraints: There are upper and lower bounds on how much flow an arc can accommodate. For each arc $(i, j) \in A$,

$$\ell_{ij} \leq x_{ij} \leq u_{ij}$$

for some $\ell_{ij}, u_{ij} \geq 0$. Here, u_{ij} can take $+\infty$, in which case we simply write $x_{ij} \geq \ell_{ij}$ without the upper bound. ℓ_{ij} is often set to 0. By defining vectors ℓ and u that collect the lower and upper bounds of arc flows, we can summarize the constraints as

$$\ell \leq x \leq u.$$

Flow balance constraints: For each node $i \in N$ and the vector x of flow values on the arcs, the **outflow** is defined as the amount of flow out of the node i:

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

The **inflow** is defined as the amount of flow into the node i:

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

The **net supply** of node i is the difference of the outflow and the inflow:

$$\text{net-supply}(i; x) = \text{outflow}(i; x) - \text{inflow}(i; x) = \sum_{j \in N:(i,j) \in A} x_{ij} - \sum_{k \in N:(k,i) \in A} x_{ki}.$$

Each node i satisfies a flow balance constraint, given by

$$\text{net-supply}(i; x) = \sum_{j \in N:(i,j) \in A} x_{ij} - \sum_{k \in N:(k,i) \in A} x_{ki} = b_i.$$

Here, each node i is either a supply or a demand node, given by a parameter b_i.

- If $b_i > 0$, i.e., the net supply is positive, then i is a **supply node**.
- If $b_i < 0$, i.e., the net supply is negative, then i is a **demand node**.
- If $b_i = 0$, i.e., the net supply is zero, then i is a **transhipment node**.

Objective: Directing one unit of flow from node i to node j incurs a cost of c_{ij}. Then the objective is to minimize the total cost.

$$\text{minimize} \quad \sum_{(i,j)\in A} c_{ij}x_{ij}.$$

To summarize, the minimum cost flow problem over network $D = (N, A)$ is modeled as the following linear program.

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