Optimization Theory and Methods

2025 Autumn

魏可佶 kejiwei@tongji.edu.cn

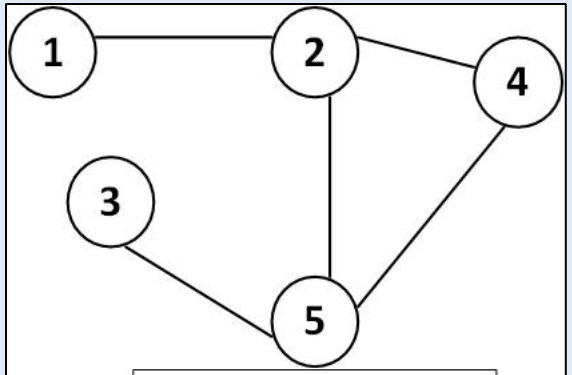
https://kejiwei.github.io/

- Introduction to network models
  - Network flow problem
- Variants of the network flow problem
  - Min-Cost Flow Problem, Maximum Flow Problem, Transportation Problem, Assignment Problem, Shortest Path Problem
- Other network-based problems
  - Multi-Commodity Flow Problem, Minimum Spanning Tree Problem, Travelling Salesperson Problem, Arc Covering Problem
- Efficient algorithms

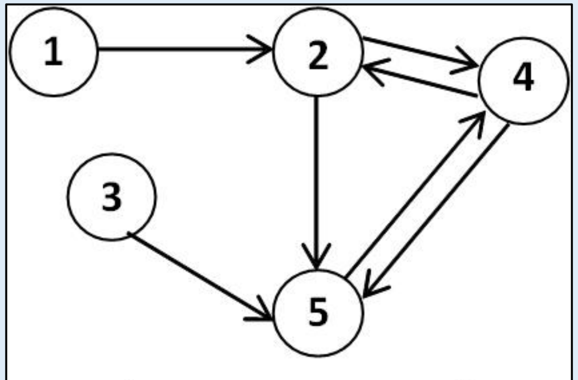
- **Networks are *everywhere*!**
- **Some are obvious**
  - E.g., physical infrastructure networks (e.g., roads), communication networks (wires and wireless links), and biological networks (e.g., arteries and veins)
- **Some are not so obvious**
  - E.g., organizational charts, family trees, time-space networks, project management networks, social networks

- Some apparently non-network problems have an underlying network structure
  - E.g., inventory management, allocating interns to jobs and room-mates to rooms, etc.
- Network models are 'nice' because they are often easy to solve
  - Usually much easier than general IPs
  - Sometimes even easier than LPs!!
  - Some categories of network-based problems though, can be very 'hard'

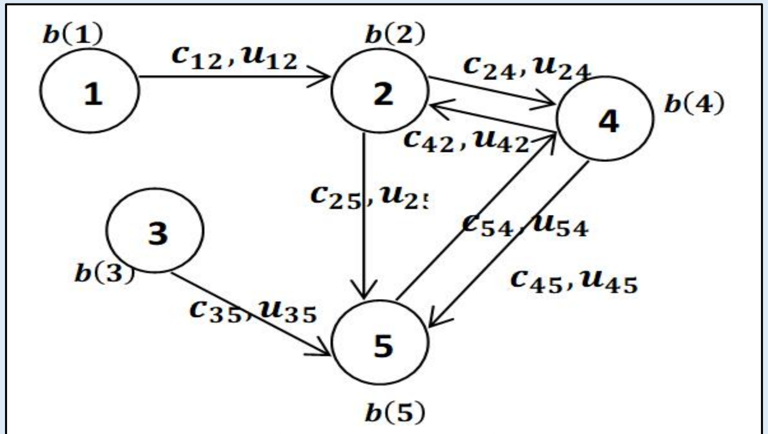
- Network problems are defined on graphs
  - Undirected and directed graphs $G = (N, A)$
    - N = Set of nodes
    - A = Set of feasible links/arcs
- Additional numerical information such as:
  - $b(i)$ representing supply and demand at each node i.
  - u_{ij} representing the capacity of each arc ij.
  - l_{ij} representing the lower bound on flow for each arc ij.
  - c_{ij} representing the cost of each arc ij.



Undirected Graph



Directed Graph



Network

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

s.t.

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = b(i), \forall i \in N \ldots \text{(1)}$$

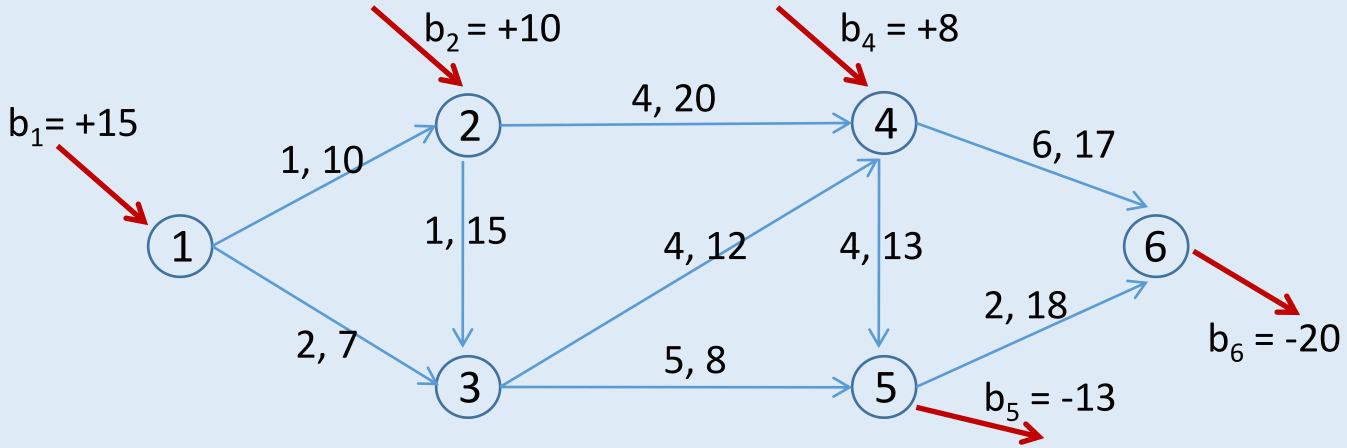
$$x_{ij} \geq l_{ij}, \forall (i,j) \in A \ldots\ldots\ldots\ldots \text{(2)}$$

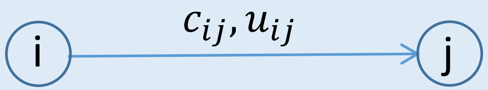
$$x_{ij} \leq u_{ij}, \forall (i,j) \in A \ldots\ldots\ldots\ldots \text{(3)}$$

$$x_{ij} \in \mathbb{Z}^{+}\ \forall (i,j) \in A \ldots\ldots\ldots\ldots \text{(4)}$$

- Constraint coefficient matrix is also called the node arc incidence matrix in this case
  - Columns = arcs
  - Rows = nodes
- Outgoing arc ⇒ +1
- Incoming arc ⇒ -1
- No arc incident ⇒ 0
- Sum of each column of A = 0

(1) Balance constraints: Flow out minus flow in must equal the supply/demand at the node.
(2) Flow lower bound constraints (usually lower bound is not stated & is 0).
(3) Capacity constraints.
(4) Integrality constraints.





c_{ij}	1	2	1	4	4	5	4	6	2
u_{ij}	10	7	15	20	12	8	13	17	18

Node	x_{12}	x_{13}	x_{23}	x_{24}	x_{34}	x_{35}	x_{45}	x_{46}	x_{56}
1	1	1	0	0	0	0	0	0	0
2	-1	0	1	1	0	0	0	0	0
3	0	-1	-1	0	1	1	0	0	0
4	0	0	0	-1	-1	0	1	1	0
5	0	0	0	0	0	-1	-1	0	1
6	0	0	0	0	0	0	0	-1	-1

b
15
10
0
8
-13
-20

Full formulation:

- **Minimize** $1x_{12} + 2x_{13} + 1x_{23} + 4x_{24} + 4x_{34} + 5x_{35} + 4x_{45} + 6x_{46} + 2x_{56}$
- **Subject to:**

$$1x_{12} + 1x_{13} + 0x_{23} + 0x_{24} + 0x_{34} + 0x_{35} + 0x_{45} + 0x_{46} + 0x_{56} = 15$$
$$-1x_{12} + 0x_{13} + 1x_{23} + 1x_{24} + 0x_{34} + 0x_{35} + 0x_{45} + 0x_{46} + 0x_{56} = 10$$
$$0x_{12} - 1x_{13} - 1x_{23} + 0x_{24} + 1x_{34} + 1x_{35} + 0x_{45} + 0x_{46} + 0x_{56} = 0$$
$$0x_{12} + 0x_{13} + 0x_{23} - 1x_{24} - 1x_{34} + 0x_{35} + 1x_{45} + 1x_{46} + 0x_{56} = 8$$
$$0x_{12} + 0x_{13} + 0x_{23} + 0x_{24} + 0x_{34} - 1x_{35} - 1x_{45} + 0x_{46} + 1x_{56} = -13$$
$$0x_{12} + 0x_{13} + 0x_{23} + 0x_{24} + 0x_{34} + 0x_{35} + 0x_{45} - 1x_{46} - 1x_{56} = -20$$
$$x_{12} \leq 10, x_{13} \leq 7, x_{23} \leq 15, x_{24} \leq 20,$$
$$x_{34} \leq 12, x_{35} \leq 8, x_{45} \leq 13, x_{46} \leq 17, x_{56} \leq 18$$
$$x_{12}, x_{13}, x_{23}, x_{24}, x_{34}, x_{35}, x_{45}, x_{46}, x_{56} \in Z^+$$

In this problem, there are no lower bounds (so the default lower bounds, i.e. zeros, apply).

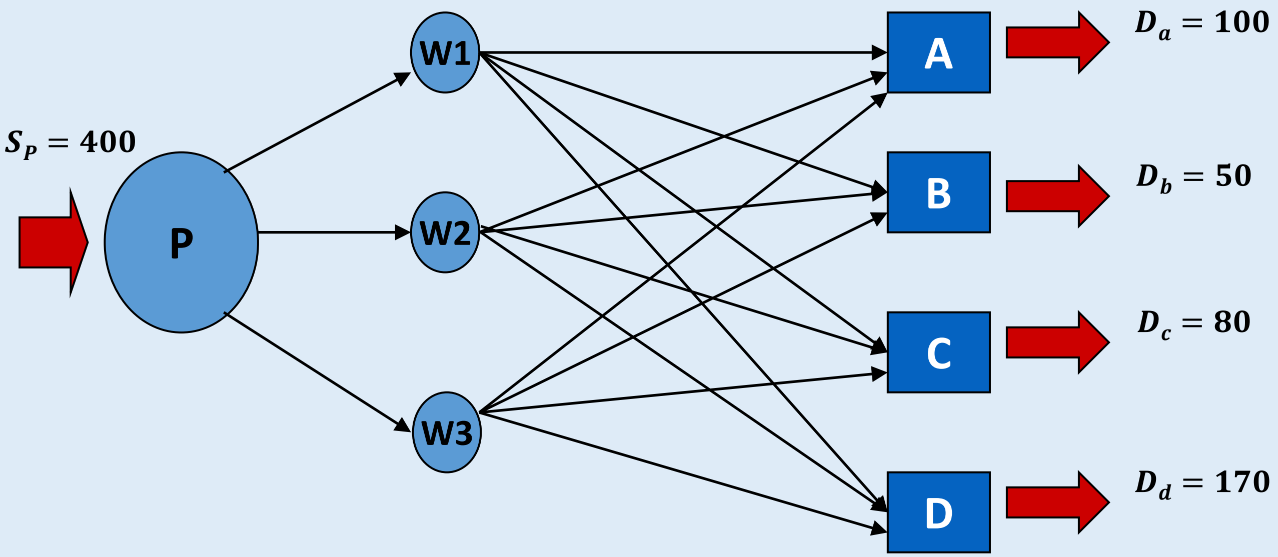
- **Solving the LP relaxation of a network flow problem with integer data (supply/demand values, upper bounds and lower bounds), yields an integer solution.**
- Network flow problems are special cases of LPs and any algorithm for solving an LP can be directly applied.
- Network flow problems have a special structure which results in substantial simplification of general methods (e.g., Network Simplex).
- Many other algorithms can also be used to solve network flow problems.
- Network-based problems, which do not have network flow structure, are sometimes more complicated (e.g., multi-commodity flow problem, travelling salesperson problem etc.).
- But there is further good news:
  - Many other types of network-based problems can be converted into the network flow problem.
  - Can just be considered as variants of the network flow problem.

- **Objective:** Determine the least cost movement of a commodity or people through a network in order to satisfy demands at certain nodes from available supplies at other nodes.
- **Applications:**
  - Distribution of a product from manufacturing plants to warehouses, and/or from warehouses to retailers.
  - Flow of raw material and intermediate goods through the various machining stations in a production line.
  - Reconstructing the left ventricle from x-ray projections.
  - Optimal loading of a hopping airplane.
  - Job shop scheduling with deferral costs.
  - Racial/ethnic diversification in schools.
  - Optimal equipment replacement problem, etc.

- **Formulation:** $G = (N, A)$: Directed network defined by a set N of n nodes and a set A of m directed arcs.
  - c_{ij}: Cost per unit flow on arc $(i, j) \in A$.
  - u_{ij}: Capacity. Maximum amount that can flow on arc (i, j).
  - l_{ij}: Lower bound. Minimum amount that must flow on the arc.
  - $b(i)$: Supply or demand at node $i \in N$.
    - If $b(i) > 0 \Rightarrow$ Node i is a supply node.
    - If $b(i) < 0 \Rightarrow$ Node i is a demand node.
    - If $b(i) = 0 \Rightarrow$ Node i is a transshipment node.
  - x_{ij}: Decision variables. Represent the quantity of flow on arc $(i, j) \in A$.

- Company A serves its 4 customers from 3 warehouses. It costs $\$c_{ij}$ to transport a unit from warehouse i to customer j. Transportation from the plant P to the warehouses is free. Transportation of the products from the warehouse to the customers is done by truck. Company A cannot send more than 100 units of product from any warehouse to any customer. Finally, there is a demand for D_j units of the product by customer j.
- Company A would like to determine how many units of product they should ship to each warehouse and how many units of product to send from each warehouse to each customer, in order to minimize costs.

↳ Warehouse Distribution Example (cont.)



Transportation costs:

c_{ij}	A	B	C	D
W1	5	2	6	7
W2	4	4	3	1
W3	3	8	5	3

Company A would like to determine how many units of product they should ship to each warehouse and how many units of product to send from each warehouse to each customer, in order to minimize costs.

↳ Warehouse Distribution : Formulation

	Node-Arc Matrix															
c_{ij}	0	0	0	5	2	6	7	4	4	3	1	3	8	5	3	D_i
	P1	P2	P3	1A	1B	1C	1D	2A	2B	2C	2D	3A	3B	3C	3D	
P	1	1	1													400
1	-1			1	1	1	1									0
2		-1						1	1	1	1					0
3			-1									1	1	1	1	0
A				-1				-1				-1				-100
B					-1				-1				-1			-50
C						-1				-1				-1		-80
D							-1				-1				-1	-170

$$\boldsymbol{Min}(5x_{1A} + 2x_{1B} + 6x_{1C} + 7x_{1D} + 4x_{2A} + 4x_{2B} + 3x_{2C} + 1x_{2D} + 3x_{3A} + 8x_{3B} + 5x_{3C} + 3x_{3D})$$

$\boldsymbol{s.t.}$

$x_{P1} + x_{P2} + x_{P3} = 400,\qquad x_{1A} + x_{1B} + x_{1C} + x_{1D} - x_{P1} = 0$

$x_{2A} + x_{2B} + x_{2C} + x_{2D} - x_{P2} = 0,\qquad x_{3A} + x_{3B} + x_{3C} + x_{3D} - x_{P3} = 0$

$-x_{1A} - x_{2A} - x_{3A} = -100,\qquad -x_{1B} - x_{2B} - x_{3B} = -50$

$-x_{1C} - x_{2C} - x_{3C} = -80,\qquad -x_{1D} - x_{2D} - x_{3D} = -170$

$x_{1A}, x_{1B}, x_{1C}, x_{1D}, x_{2A}, x_{2B}, x_{2C}, x_{2D}, x_{3A}, x_{3B}, x_{3C}, x_{3D} \le 100$

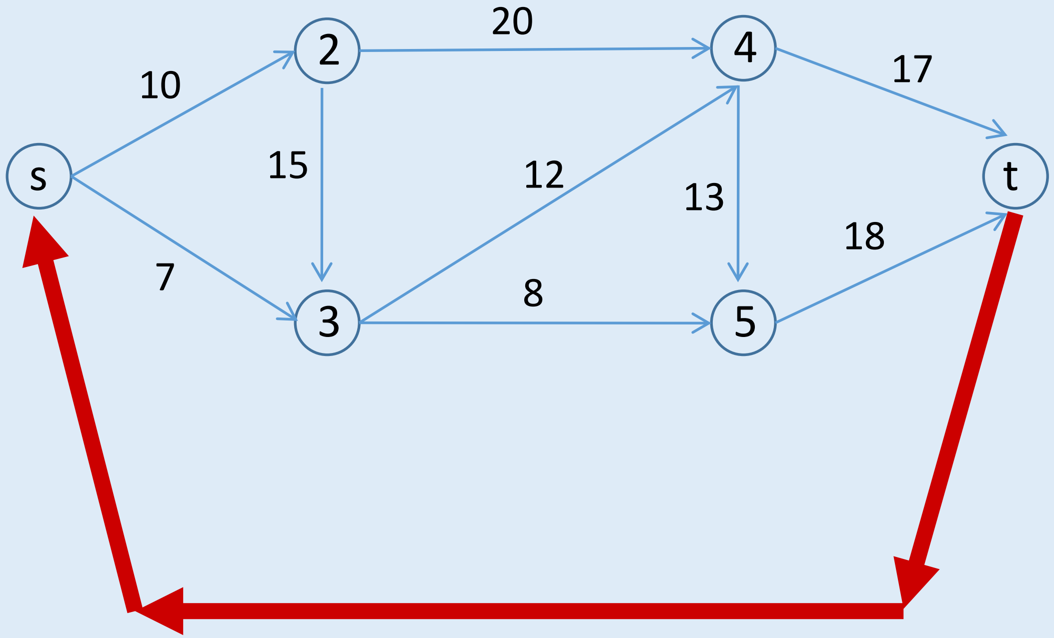
$x_{P1}, x_{P2}, x_{P3}, x_{1A}, x_{1B}, x_{1C}, x_{1D}, x_{2A}, x_{2B}, x_{2C}, x_{2D}, x_{3A}, x_{3B}, x_{3C}, x_{3D} \ge 0$

- How would you model the following:
  - Maximum capacity of warehouse W1 is 150 units.
  - Maximum capacity of warehouse W1 is 150 units and there are 2 production plants.
  - The total demand exceeds total supply, i.e., not all demand needs to be satisfied. We just need to use all the supply. Let total supply be 300.
  - Let total supply be only 300. We need to use all the supply. Customers B and C, between them get at least 100 units.

- **Objective:** Find a set of arc flows such that the maximum amount of flow is sent from a specified source node s to another specified sink node t.
- Maximum flow problem incurs no cost, but it is restricted by arc capacities.
- **Applications:**
  - Finding the maximum steady state flow of petroleum products in a pipeline network, cars in a road network, messages in a telecom network, electricity in an electrical network.
  - Finding a balanced governing council for a town with limits on maximum representatives of each political party.
  - Matrix rounding problem.
  - Scheduling on uniform parallel machines.
  - Distributed computing on a two-processor computer, etc.

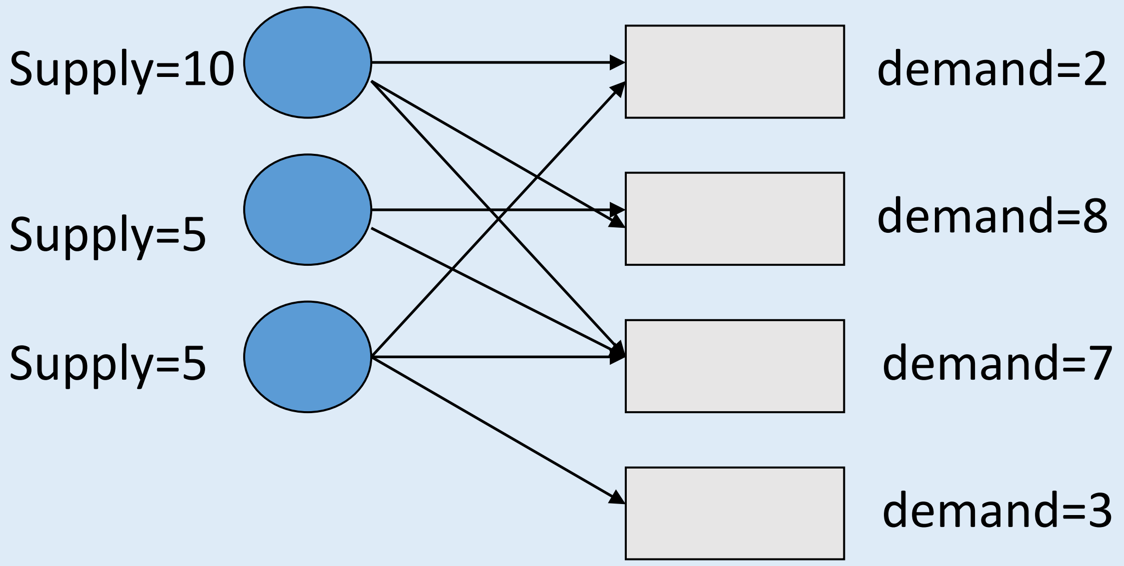
- **Formulation:** $G = (N, A)$: Directed network with set N of nodes and set A of m directed arcs.

- To transform to a minimum cost network flow problem:
  - Introduce a new arc: arc $t - s$.
  - Capacity of arc $t - s$: infinite.
  - Cost of arc $t - s$: -1.
  - Cost of all other arcs: 0.



- Transformation into minimum cost flow problem:
  - $c_{ij} = 0 \; \forall (i,j) \in A$. Introduce arc from t to s s.t. $c_{ts} = -1$.
  - u_{ij} as given; $u_{ts} = \infty$.
  - l_{ij}: Lower bound if given.
  - $b(i) = 0$ for all $i \in N$.
  - x_{ij}: Decision variables, representing the quantity of flow on arc $(i,j) \in A$.
- **Solution**: Minimize the cost of flow on arc $t - s$. This is equivalent to maximizing the flow from s to t because any flow on arc (t,s) must travel from node s to node t through the arcs in A [because each $b(i) = 0$]. Thus, the solution to the minimum cost flow problem will maximize the flow from s to t.

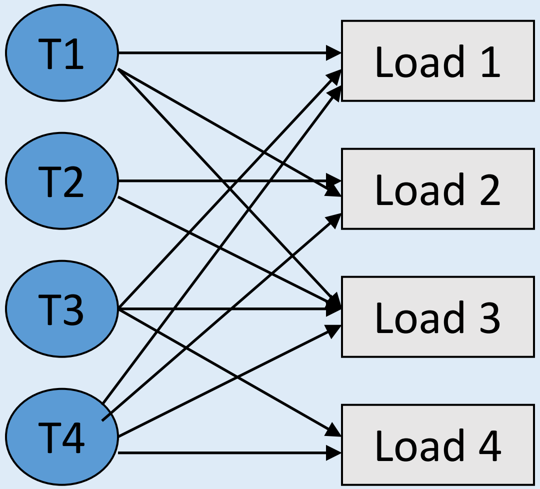
c_{ij}	1	2	3	4
1	6	4	5	n/a
2	n/a	3	6	n/a
3	5	n/a	4	3



- **Objective:** Transport the goods from the suppliers to the consumers at minimum cost given that:
  - There are m suppliers and n consumers (m can be different from n).
  - The i^{th} supplier can provide s_i units and the j^{th} consumer has a demand for d_j units.
  - We assume that total supply equals total demand.
- **Applications:**
  - Distribution of goods from warehouses to customers.
  - Automatic karyotyping of chromosomes.
  - Warehouse layout problem, etc.

- **Formulation:** $G = (N_1 \cup N_2, A)$: Directed network with set $N_1 + N_2$ of nodes and set A of directed arcs.
  - N_1: Supply nodes. N_2: Demand nodes. $|N_1| = m$. $|N_2| = n$.
  - $(i, j) \in A$ such that $i \in N_1$ and $j \in N_2$.
  - $b(i)$:
    - $b(i) = s_i$ for all $i \in N_1$
    - $b(i) = -d_i$ for all $i \in N_2$
  - c_{ij}, u_{ij}, and l_{ij}: Unit transportation cost, minimum amount and maximum amount of goods from supplier i to consumer j.
  - x_{ij}: Decision variables representing quantity of goods flowing from i to j.

↳ Assignment Problem



c_{ij}	Load 1	Load 2	Load 3	Load 4
T1	6	4	5	n/a
T2	n/a	3	6	n/a
T3	5	n/a	4	3
T4	7	5	5	5

- Trucking company TC needs to pick up 4 loads of products at different locations. Currently, 4 truck drivers are available to pick up those shipments. The cost of having driver i pick-up load j is illustrated in the above table.
- Formulate the problem of assigning each driver to a load, in order to minimize costs.

- **Objective:** Pair, at minimum possible cost, each object in set N_1 with exactly one object in set N_2.
- Special case of the transportation problem where number of suppliers equals number of customers and each supplier has unit supply, each consumer has unit demand.
- **Applications:**
  - Assigning people to projects, truckloads to truckers, jobs to machines, tenants to apartments, swimmers to events in a swimming competition, school graduates to internships.
  - Discrete facility location problem.
  - Matching objects in space using sensor measurements.
  - Determining chemical bonds.
  - Dual completion of oil wells, etc.

- **Formulation:** $G = (N_1 \cup N_2, A)$: Directed network with set $N_1 + N_2$ of nodes and set A of m directed arcs.
  - $|N_1| = |N_2|$.
  - $(i, j) \in A$ such that $i \in N_1$ and $j \in N_2$.
  - $b(i)$: Supply or demand at node $i \in N$.
    - ➢ $b(i) = 1$ for all $i \in N_1$.
    - ➢ $b(i) = -1$ for all $i \in N_2$.
  - c_{ij}: cost per unit flow on arc $(i, j) \in A$.
  - $u_{ij} = 1$ for all $(i, j) \in A$.
  - l_{ij}: Lower bound if given. Minimum amount that must flow on the arc. It effectively removes the node pair from network.
  - x_{ij}: Decision variables representing the quantity flowing on arc $(i, j) \in A$.

- **Objective:** Find the path of minimum cost (or length) from a specified source node s to another specified sink t, assuming that each arc $(i, j) \in A$ has an associated cost (or length) c_{ij}.
- **Applications:**
  - Project scheduling, cash flow management, message routing in communication systems, traffic flow through a congested city.
  - DNA sequence alignment.
  - Dynamic lot sizing in production and inventory management.
  - Approximating piecewise linear functions.
  - Optimizing the spacing between words in a text editor.
  - Telephone operator scheduling.
  - Optimizing the path of a postman, etc.

TONGJI SEM 同济经管

- **Formulation:** $G = (N, A)$: Directed network with set N of nodes and set A of m directed arcs.
  - c_{ij}: Cost per unit flow on arc $(i, j) \in A$.
  - $b(i)$: Supply or demand at node $i \in N$.
    - $b(s) = 1$, $b(t) = -1$.
    - $b(i) = 0$ for all other nodes.
  - x_{ij}: Decision variables representing the quantity of flow on arc $(i, j) \in A$.
  - No need for any upper or lower bounds on arc flows.
- If we want to determine shortest path from source to every other node in the system, then: $b(s) = (n-1)$; $b(i) = -1$ for all other nodes. (**Why?**)

- Several commodities use the same underlying network.
  - **Objective:** Allocate the capacity of each arc to the individual commodities in a way that minimizes overall costs.
  - Commodities may either be differentiated by their physical characteristics or simply by their origin-destination pairs.
  - Different commodities have different origins and destinations, and commodities have separate balance constraints at each node.
  - Commodities share the arc capacities.
  - Cannot be converted into a standard network flow problem.
  - Integrality property doesn't hold.

Applications:

- Transportation of passengers from different origins to different destinations within a city.
- Routing of non-homogeneous tankers.
- Worldwide shipment of different varieties of grains from countries that produce grains to countries that consume it.
- The transmission of messages in a communication network between different origin-destination pairs, etc.

- Given the costs of constructing each arc, design a network (i.e., construct arcs) so that a given group of nodes are connected, while minimizing the total construction cost.
- A common problem encountered in a variety of transportation and telecommunication problems.
  - Designing a road/railway network to connect cities/depots.
  - Designing communication network to connect a set of computers and/or phones.
  - Grouping objects into clusters.
  - Sequencing of amino acids in proteins, etc.
- This is called a Minimum Spanning Tree (**MST**) problem. (Why tree? Why spanning?)

- **Kruskal's Algorithm** — An algorithm for ***finding a minimum spanning tree***. Let ***G*** be an undirected connected weighted graph of order ***n***.

  (1) Sort all edges in non-decreasing order of weight (excluding cycles), i.e. $W(e_1) \le W(e_2) \le \ldots \le W(e_m)$.

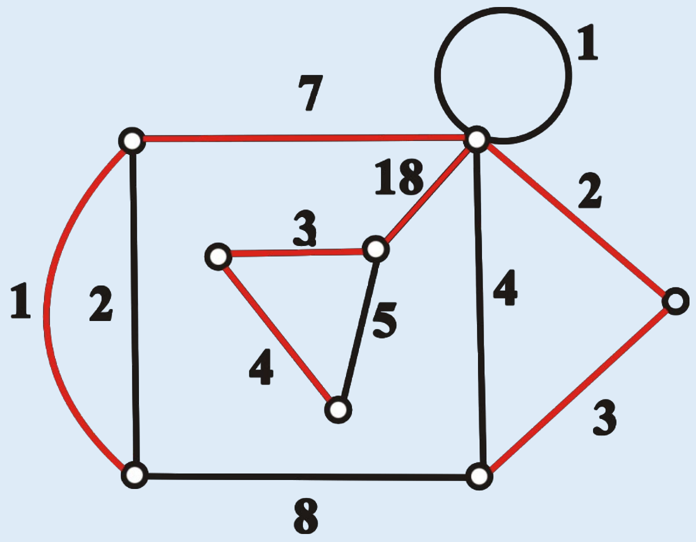
  (2) Initialize $T \leftarrow \varnothing$, $i \leftarrow 1$, $k \leftarrow 0$.

  (3) If e_i does not form a cycle with the edges already in ***T***, then set $T \leftarrow T \cup \{e_i\}$, $k \leftarrow k+1$.

  (4) If $k<n-1$, then set $i \leftarrow i+1$, and repeat step (3).

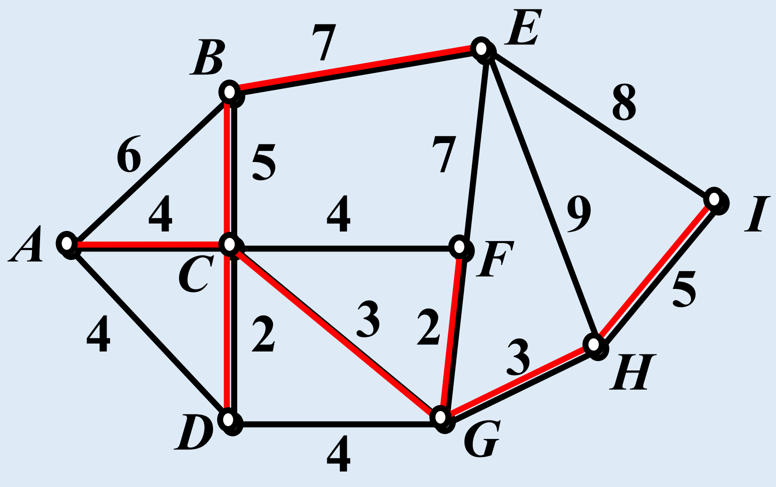
同济经管 TONGJI SEM

- **Example:** Find a minimum spanning tree of the graph.



$W(T)=38$

- **Example:** A telecommunications company has a ***fiber-optic network*** coverage requirement as shown in the figure. The possible cable routes between buildings and their lengths (in meters) are given in the diagram.
- **Question:** Under the condition that the entire network must be connected, how should the cable routes be selected to ensure that the ***total cable length is minimized***?
- **Solution:**



Total Length :

2+2+3+3+4+5+5+7=31(meters)

↳ Solving Minimum Spanning Tree Problem

- A very 'easy' problem to solve
  - Greedy algorithms 'work'.
- Kruskal's algorithm:
  - Arrange arcs in increasing (non-decreasing) order of costs.
  - Start from first arc in the list. Select if it does not create a cycle.
  - Move to next arc.
  - Repeat until you have looked at all arcs.
- Prim's algorithm:
  - Start from any node and grow the tree greedily.
  - Consider all outgoing arcs from the tree nodes to the non-tree nodes.
  - Select the one with the least cost among them. Add to the tree.
  - Repeat until no non-tree nodes are left.

↳ Node Covering Problem (Traveling Salesperson Problem (TSP))

- A salesperson travels from city-to-city.
- Find a tour that starts from a city (node), visits every other city (node) exactly once and returns to the starting city (node).
- A commonly arising problem in many fields
  - Optimizing vehicle routing.
  - Determining the landing sequence of aircraft on a runway.
  - Machine scheduling in a machine shop.
  - Genome sequencing.
  - Imaging of celestial objects.
  - Testing in semi-conductor manufacturing.
  - Optimizing the design of a fiber optics network, etc.

↳ Solving Traveling Salesperson Problem (TSP)

- A very difficult problem to solve (*NP-hard*).
- No fast algorithm exists.
- A very hot research topic. One of the most exciting open problems in mathematics today!
- If you can solve this, then you can automatically solve dozens of other ‘hard’ problems.
- Most existing approaches are heuristics and approximation methods.
- Some of these methods involve solving several other, easier network problems, e.g. MST, assignment, etc.

↳ Arc Covering Problem (Chinese Postman Problem (CPP))

- Postman travels from door to door covering each street at least once to deliver mail.
- Find a tour that starts from a node, traverses each arc at least once, and returns to the starting node.
- Many variations and applications.
  - Snow plowing.
  - Street sweeping.
  - Mail delivery.
  - CPP with time windows: Some arcs can be covered only during some times of the day.
  - Rural CPP: Some arcs don’t have to be covered.
  - Etc.

Objective :

Key Concepts :