OPERATIONAL RESEARCH & OPTIMISATION

Dr. Fang He
Email: F.He@westminster.ac.uk
Office: N7.118
University of Westminster

Network optimisation problems Outline

- We examine the characteristics of network models;
- Formulate some examples of these models;
- Study approach to their solution.

General Minimum Cost Flow Problem

Consider a directed and connected network where the n nodes include at least one supply node and at least one demand node. The decision variables are

x_{ij} = flow through arc $i \rightarrow j$,

and the given information includes

c_{ij} = cost per unit flow through arc $i \rightarrow j$,
u_{ij} = arc capacity for arc $i \rightarrow j$,
b_i = net flow generated at node i.

The value of b_i depends on the nature of node i, where

$b_i > 0$ if node i is a supply node,
$b_i < 0$ if node i is a demand node,
$b_i = 0$ if node i is a transshipment node.

$$\text{Minimise}: \quad Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{Subject to}: \quad \sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = b_i \quad \text{for} \quad i = 1,2\ldots n$$

$$0 \le x_{ij} \le u_{ij} \quad \text{for each arc} \quad i \rightarrow j$$

General Minimum Cost Flow Problem

The objective is to find the minimum-cost flow pattern to fulfill demands from the source nodes. Such problems usually are referred to as *minimum-cost flow* or *capacitated transshipment* problems. To transcribe the problem into a formal linear program, let

$$x_{ij} = \text{Number of units shipped from node } i \text{ to } j \text{ using arc } i\text{–}j.$$

Then the tabular form of the linear-programming formulation associated with the network of Fig. 8.1 is as shown in Table 8.2.

The first five equations are flow-balance equations at the nodes. They state the conservation-of-flow law,

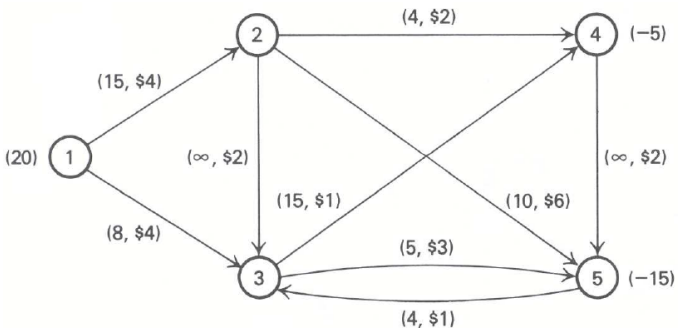
$$\begin{pmatrix}\text{Flow out}\\ \text{of a node}\end{pmatrix} - \begin{pmatrix}\text{Flow into}\\ \text{a node}\end{pmatrix} = \begin{pmatrix}\text{Net supply}\\ \text{at a node}\end{pmatrix}.$$

General Minimum Cost Flow Problem

As examples, at nodes 1 and 2 the balance equations are:

$$x_{12} + x_{13} = 20$$
$$x_{23} + x_{24} + x_{25} - x_{12} = 0.$$

It is important to recognize the special structure of these balance equations. Note that there is one balance equation for each node in the network. The flow variables x_{ij} have only 0, +1, and −1 coefficients in these



General Minimum Cost Flow Problem- Important special structure

Table 8.2 Tableau for Minimum-Cost Flow Problem

	x_{12}	x_{13}	x_{23}	x_{24}	x_{25}	x_{34}	x_{35}	x_{45}	x_{53}	*Righthand side*
Node 1	1	1								20
Node 2	−1		1	1	1					0
Node 3		−1	−1			1	1		−1	0
Node 4				−1		−1		1		−5
Node 5					−1		−1	−1	1	−15
Capacities	15	8	∞	4	10	15	5	∞	4	
Objective function	4	4	2	2	6	1	3	2	1	(Min)

General Minimum Cost Flow Problem

LP Formulation for Minimum Cost Flow Problem

$$\text{Minimise}: \quad Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{Subject to}: \quad \sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = b_i \quad \text{for} \quad i = 1,2\ldots n$$

$$0 \le x_{ij} \le u_{ij} \quad \text{for each arc} \quad i \to j$$

Flow balance

Flow capacity

- Minimum cost flow, a unified approach:
  - Transportation problem; (no transhipment nodes, no capacity limits)
  - Assignment problem; (a special type of transportation problem)
- Recap
- New:
  - Maximum flow problem;
  - Shortest-path problem;

LP Formulation for Transportation Problem

$$\text{Minimise}: \quad Z = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij}$$

$$\text{Subject to}: \quad \sum_{j=1}^{m} x_{ij} = s_i \quad \text{for } i = 1,2\ldots n$$

$$\sum_{i=1}^{n} x_{ij} = d_j \quad \text{for } j = 1,2\ldots m$$

$$x_{ij} \geq 0 \quad \text{for all } i \text{ and } j$$

This model of the transportation problem assumes that the problem is balanced.

Assignment Optimization Problems

The Assignment Problem

There is a set of N tasks and a set of N workers with a certain cost c_{ij} for assigning each task to each worker. Each task must be assigned to exactly one worker and each worker must undertake exactly one task. The problem is to assign all the tasks to the workers so that the total cost is minimised.

Minimise : $Z = \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij}$

Subject to : $\sum_{j=1}^{N} x_{ij} = 1 \quad \text{for } i = 1,2,\ldots N \quad (1)$

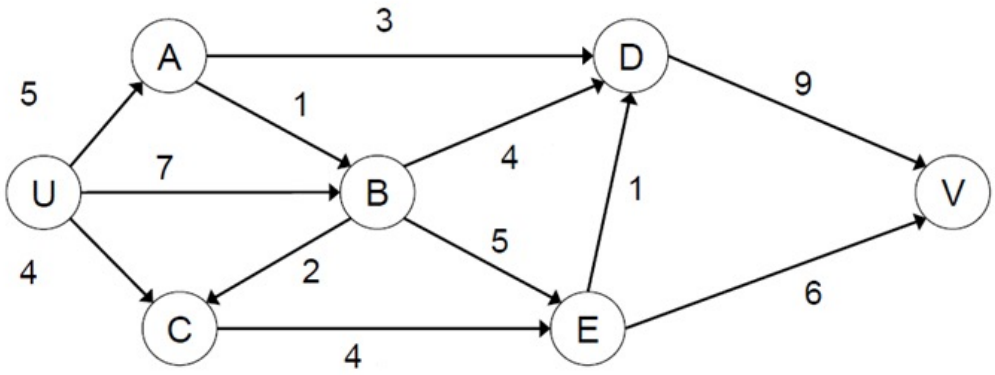
$\sum_{i=1}^{N} x_{ij} = 1 \quad \text{for } j = 1,2,\ldots N \quad (2)$

$x_{ij} = 1$ if task i is assigned to worker j, 0 otherwise

Maximum flow-a prototype example

Maximum Flow Problem

Consider a directed network with a set of nodes, a source node, a sink node and transhipment nodes. A non-negative maximum flow capacity is associated to each arc. The objective is to maximise the total flow from the source to the sink. Several algorithms exist for the maximum flow problem.

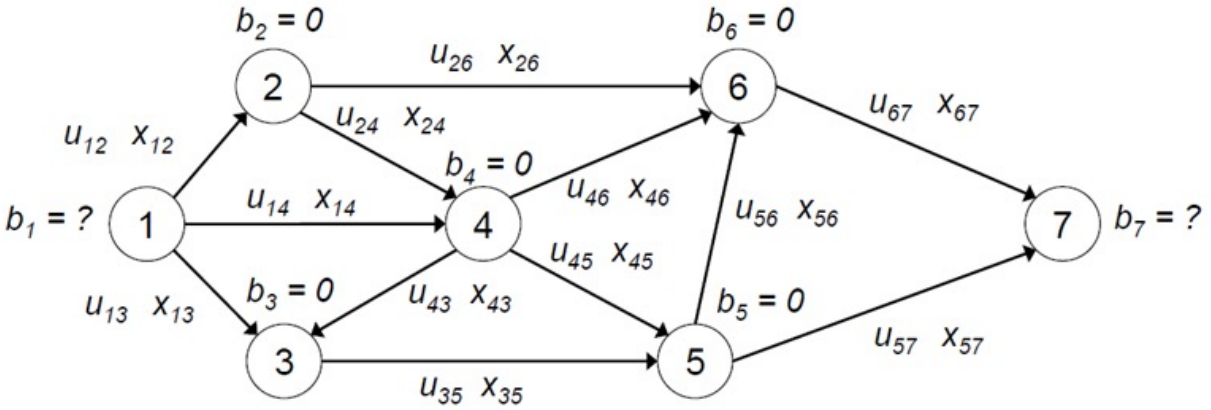


Maximum Flow: 14

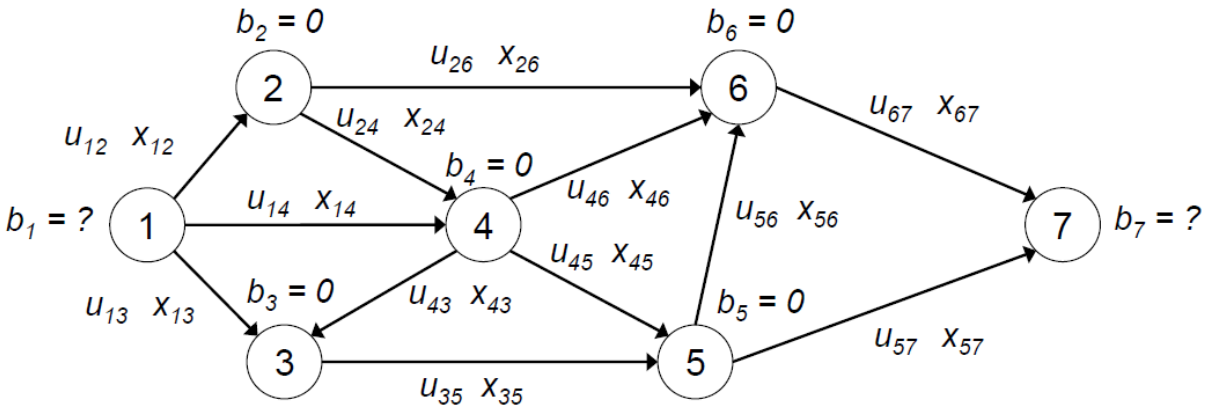
Scenario

Let us again consider a simple example. A city has constructed a piping system to route water from a lake to the city reservoir. The system is now underutilized and city planners are interested in its overall capacity. The situation is modeled as finding the maximum flow from node u, the lake, to node v, the reservoir, in the network shown in Fig. 8.3.

Maximum flow problem



- Exercise: formulate the problem;



$$\text{Maximise}: \quad Z = x_{12} + x_{13} + x_{14}$$

$$\text{Subject to}: \quad -x_{12} + x_{24} + x_{26} = 0 \qquad (1)$$

$$-x_{13} - x_{43} + x_{35} = 0 \qquad (2)$$

$$-x_{14} - x_{24} + x_{43} + x_{45} + x_{46} = 0 \qquad (3)$$

$$-x_{35} - x_{45} + x_{56} + x_{57} = 0 \qquad (4)$$

$$-x_{26} - x_{46} - x_{56} + x_{67} = 0 \qquad (5)$$

$$0 \le x_{ij} \le u_{ij} \quad \text{for each arc } i \to j \qquad (6)$$

Note: The objective function; the values of $b_{i;}$ Check if it presents the special structure of the network model;

LP Formulation for Maximum Flow

$$\text{Maximise}: \quad Z = \sum_{j=2}^{n} x_{1j} \qquad \text{for each arc } 1 \rightarrow j$$

$$\text{alternatively}: \quad Z = \sum_{i=1}^{n-1} x_{in} \qquad \text{for each arc } i \rightarrow n$$

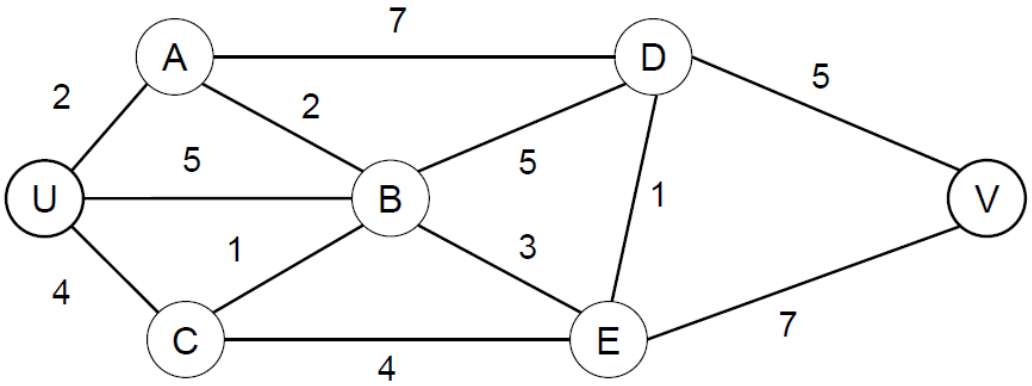
$$\text{Subject to}: \quad \sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = b_i \quad \text{for } i = 2 \ldots (n-1)$$

$$0 \leq x_{ij} \leq u_{ij} \quad \text{for each arc } i \rightarrow j$$

This model of the maximum flow problem assumes that there is one source and one sink node.

Shortest Path as a Flow Problem

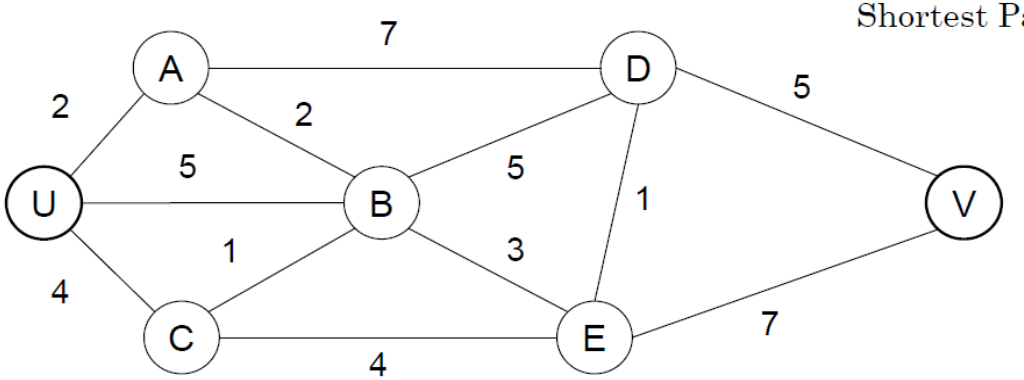
Consider an undirected network with non-negative distance associated to each link. The objective is to find the shortest path between the origin node and the destination node. Several algorithms exist for the shortest path problem.



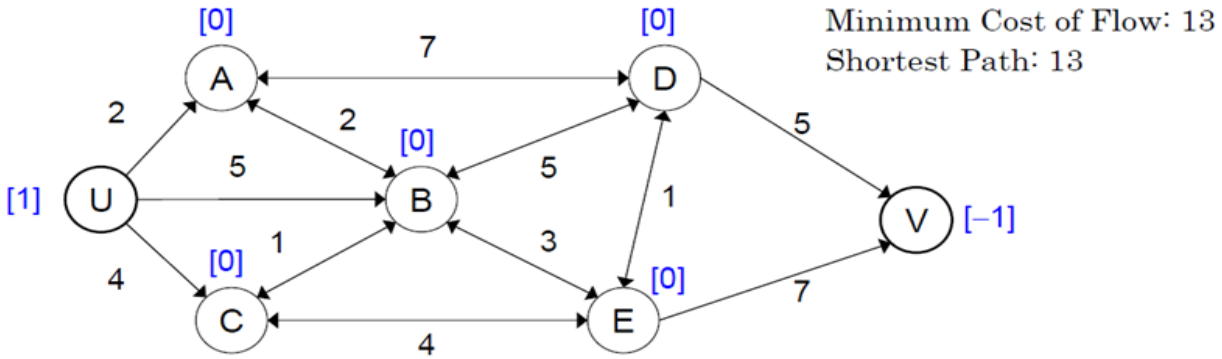
Shortest Path: 13

Formulating Shortest Path as Minimum Cost Flow

Shortest Path: 13



- distance d_{ij} becomes cost of flow c_{ij}
- maximum flow capacity is set as ∞ for all arcs
- flow in each arc in the shortest path will be 1, other flows will be 0
- origin node becomes supply node with supply $b_u = 1$
- destination node becomes demand node with demand $b_v = -1$
- all the other nodes are transhipment nodes with $b_i = 0$
- links are replaced by directed arcs



$$
\begin{aligned}
\text{Minimise}: \quad & Z = 2x_{UA} + 5x_{UB} + 4x_{UC} + 2(x_{AB} + x_{BA}) + (x_{BC} + x_{CB}) + 7(x_{AD} + x_{DA}) + \\
& 4(x_{CE} + x_{EC}) + 5(x_{BD} + x_{DB}) + 3(x_{BE} + x_{EB}) + (x_{DE} + x_{ED}) + 5x_{DV} + 7x_{EV} && (1)\\
\text{Subject to}: \quad & x_{UA} + x_{UB} + x_{UC} = 1 && (2)\\
& x_{AB} + x_{AD} - x_{UA} - x_{BA} - x_{DA} = 0 && (3)\\
& x_{BA} + x_{BC} + x_{BD} + x_{BE} - x_{UB} - x_{AB} - x_{CB} - x_{DB} - x_{EB} = 0 && (4)\\
& x_{CB} + x_{CE} - x_{UC} - x_{BC} - x_{EC} = 0 && (5)\\
& x_{DA} + x_{DB} + x_{DE} + x_{DV} - x_{AD} - x_{BD} - x_{ED} = 0 && (6)\\
& x_{EB} + x_{EC} + x_{ED} + x_{EV} - x_{BE} - x_{CE} - x_{DE} = 0 && (7)\\
& -x_{DV} - x_{EV} = -1 && (8)\\
& 0 \le x_{ij} \le 1 \text{ or more precisely } x_{ij} \in \{0,1\} \text{ for } i, j \in \{U, V, A, B, C, D, E\} && (9)
\end{aligned}
$$

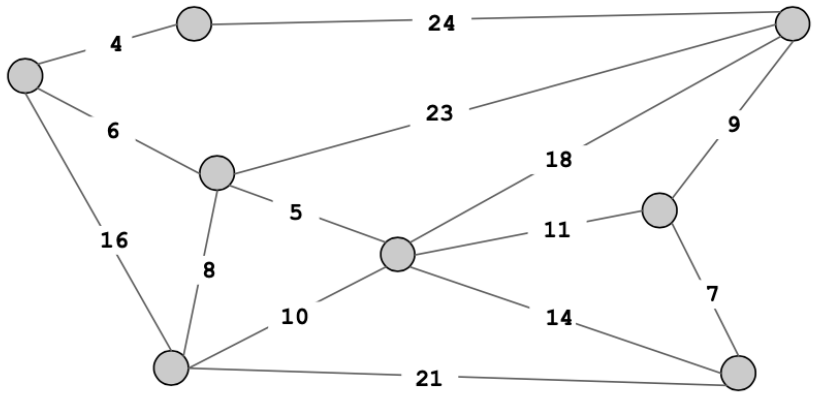
Note: the direction of the arcs; the values of b_i;

Network models

- Minimum cost flow, a unified approach:
  - Transportation problem; (no transhipment nodes, no capacity limits)
  - Assignment problem; (a special type of transportation problem)
  - Maximum flow problem;
  - Shortest-path problem;
- Minimum spanning tree:

Minimum Spanning Tree

MST. Given connected graph G with positive edge weights, find a min weight set of edges that connects all of the vertices.



G

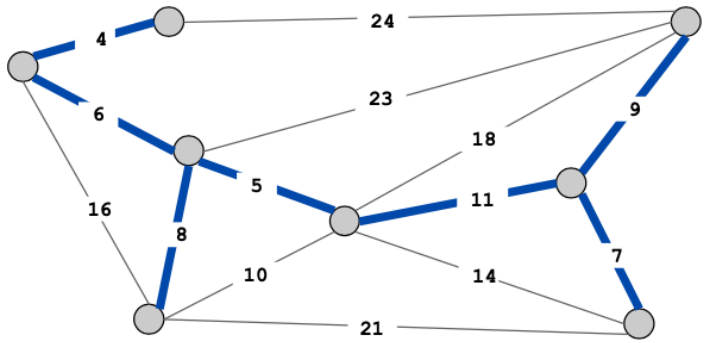
- Slides 21,23,24,37 are from: Robert Sedgewick and Kevin Wayne Copyright © 2006 http://www.Princeton.EDU/~cos226

Minimum spanning tree from Wikipedia

- A telecommunications company trying to lay cable in a new neighbourhood. If it is constrained to bury the cable only along certain paths (e.g. roads), then there would be a graph containing the points (e.g. houses) connected by those paths. Some of the paths might be more expensive, because they are longer, or require the cable to be buried deeper; these paths would be represented by edges with larger weights.
- A *spanning tree* for that graph would be a subset of those paths that has no cycles but still connects every house; there might be several spanning trees possible. A *minimum spanning tree* would be one with the lowest total cost, representing the least expensive path for laying the cable.

Minimum Spanning Tree

MST. Given connected graph G with positive edge weights, find a min weight set of edges that connects all of the vertices.



`cost(T) = 50`

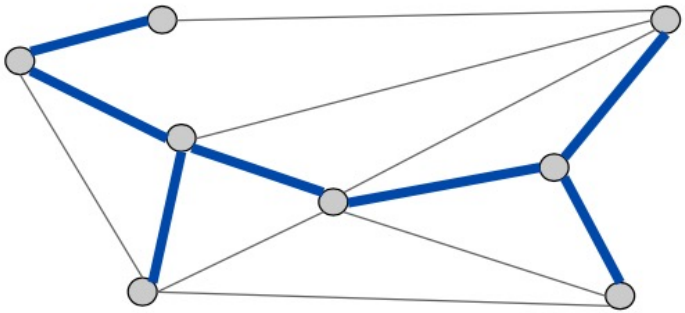
Theorem. [Cayley, 1889] There are V^{V-2} spanning trees on the complete graph on V vertices.

can't solve by brute force

MST Origin

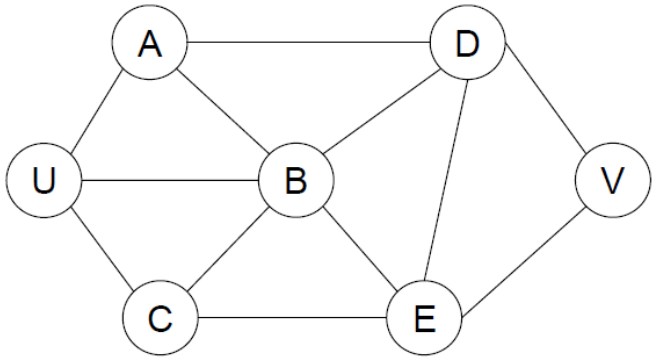
Otakar Boruvka (1926).

- Electrical Power Company of Western Moravia in Brno.
- Most economical construction of electrical power network.
- Concrete engineering problem is now a cornerstone problem in combinatorial optimization.

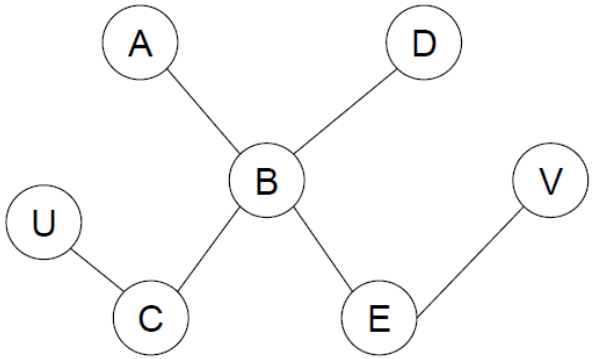


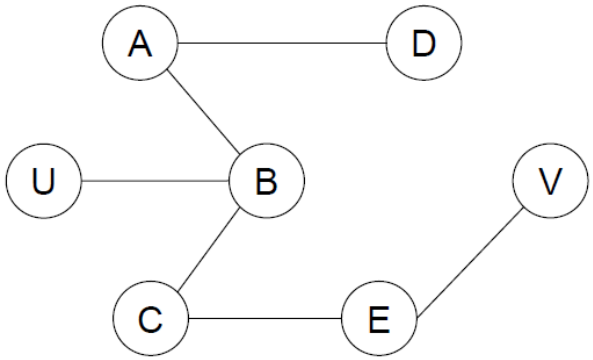
Otakar Boruvka

Minimum Spanning Tree Problem



For n nodes, the spanning tree has n–1 links





Minimum Spanning Tree Formulation

- Let x_{ij} be 1 if edge ij is in the tree T.
- Need constraints to ensure that:
  - $n-1$ edges in T
  - no cycles in T.

First constraint:

$$\sum_{ij \in E} x_{ij} = n - 1$$

Second constraint. Subtour elimination constraint. Any subset of k vertices must have at most $k-1$ edges contained in that subset.

$$\sum_{ij \in E: i \in S, j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subseteq V$$

IP formulation

$$
\begin{aligned}
\textbf{minimize } & \sum_{ij \in E} c_{ij} x_{ij} \\
\textbf{subject to } & \sum_{ij \in E} x_{ij} = n - 1 \\
& \sum_{ij \in E : i \in S, j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subseteq V \\
& x_{ij} \in \{0, 1\} \quad \forall ij \in E
\end{aligned}
$$

- This is an exponential-sized IP. One can imagine that it is not a great idea to write down and solve this directly.
- One can formulate the LP relaxation:

$$
\begin{aligned}
\textbf{minimize } & \sum_{ij \in E} c_{ij} x_{ij} \\
\textbf{subject to } & \sum_{ij \in E} x_{ij} = n - 1 \\
& \sum_{ij \in E : i \in S, j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subseteq V \\
& x_{ij} \geq 0 \quad \forall ij \in E
\end{aligned}
$$

Minimise : $Z = \sum_{e \in E} w_e x_e$

Subject to : $\sum_{e \in E} x_e = N - 1$ (1)

$\sum_{e \in (S,S)} x_e \leq |S| - 1 \quad \forall S \subseteq V$ (2)

$x_e \in \{1,0\}$ is to select or not edge e

w_e is the weight of edge e

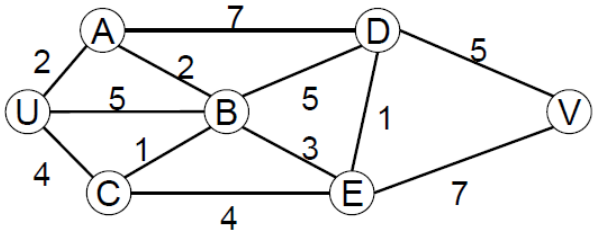
E is the set of edges

V is the set of nodes or vertices

$N = |V|$ is the number of nodes or vertices

S is a subset of V

(S,S) is the set of all edges that go from one node in S to another node in S



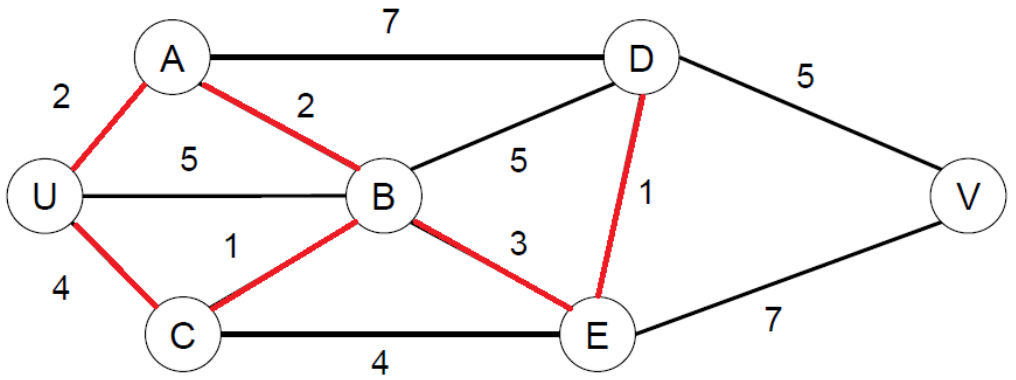
Applying this BIP formulation to the above MST problem could results in an algebraic formulation with 119 constraints in total.

- Exercise: how many cycle can you find with number of arcs of 3, 4, 5,6?

- Another slightly better way to do it:
- Add the sub-tour (cycle) elimination constraints iteratively.

Solving the MST Problem

$$\begin{aligned}\text{Minimise}: \quad & Z = 2x_{UA} + 5x_{UB} + 4x_{UC} + 2x_{AB} + 7x_{AD} + x_{BC} + \\ & 5x_{BD} + 3x_{BE} + 4x_{CE} + x_{DE} + 5x_{DV} + 7x_{EV} \\ \text{Subject to}: \quad & x_{UA} + x_{UB} + x_{UC} + x_{AB} + x_{AD} + x_{BC} + x_{BD} + x_{BE} + x_{CE} + x_{DE} + x_{DV} + x_{EV} = 6 \qquad (1) \\ & 0 \le x_{ij} \le 1\end{aligned}$$



Z = 13

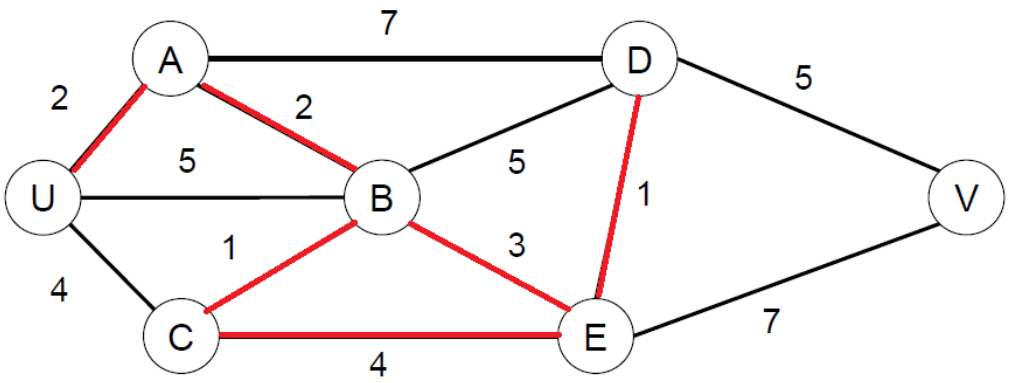
Solving the MST Problem

$$\text{Minimise}: \quad Z = 2x_{UA} + 5x_{UB} + 4x_{UC} + 2x_{AB} + 7x_{AD} + x_{BC} + 5x_{BD} + 3x_{BE} + 4x_{CE} + x_{DE} + 5x_{DV} + 7x_{EV}$$

$$\text{Subject to}: \quad x_{UA} + x_{UB} + x_{UC} + x_{AB} + x_{AD} + x_{BC} + x_{BD} + x_{BE} + x_{CE} + x_{DE} + x_{DV} + x_{EV} = 6 \qquad (1)$$

$$x_{UA} + x_{UC} + x_{AB} + x_{BC} \le 3 \qquad (2)$$

$$0 \le x_{ij} \le 1$$



Z = 13

Solving the MST Problem

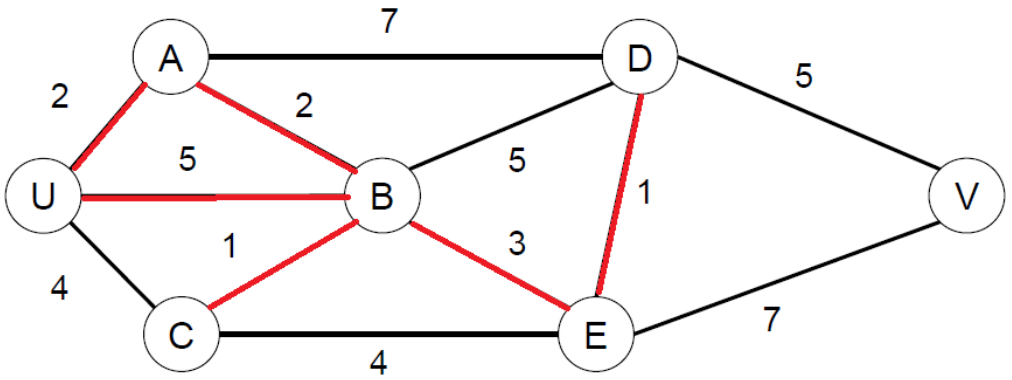
$$\text{Minimise}: \quad Z = 2x_{UA} + 5x_{UB} + 4x_{UC} + 2x_{AB} + 7x_{AD} + x_{BC} + 5x_{BD} + 3x_{BE} + 4x_{CE} + x_{DE} + 5x_{DV} + 7x_{EV}$$

$$\text{Subject to}: \quad x_{UA} + x_{UB} + x_{UC} + x_{AB} + x_{AD} + x_{BC} + x_{BD} + x_{BE} + x_{CE} + x_{DE} + x_{DV} + x_{EV} = 6 \qquad (1)$$

$$x_{UA} + x_{UC} + x_{AB} + x_{BC} \le 3 \qquad (2)$$

$$x_{BC} + x_{BE} + x_{CE} \le 2 \qquad (3)$$

$$0 \le x_{ij} \le 1$$



Z = 14

Solving the MST Problem

$$\text{Minimise}: \quad Z = 2x_{UA} + 5x_{UB} + 4x_{UC} + 2x_{AB} + 7x_{AD} + x_{BC} + 5x_{BD} + 3x_{BE} + 4x_{CE} + x_{DE} + 5x_{DV} + 7x_{EV}$$

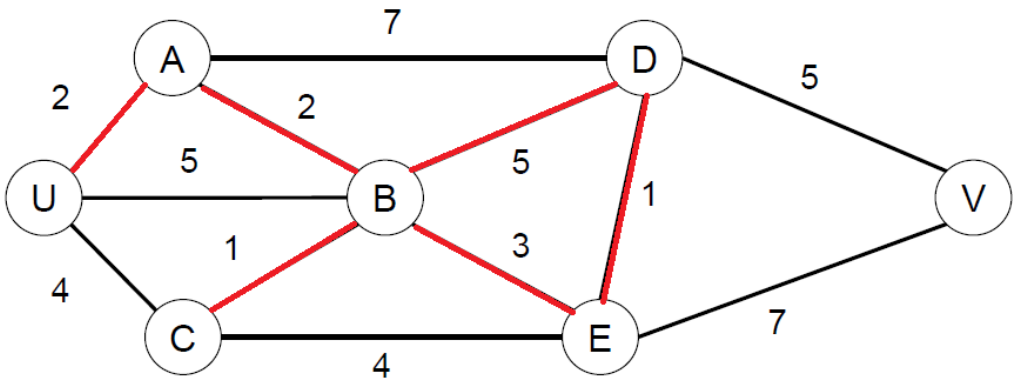
$$\text{Subject to}: \quad x_{UA} + x_{UB} + x_{UC} + x_{AB} + x_{AD} + x_{BC} + x_{BD} + x_{BE} + x_{CE} + x_{DE} + x_{DV} + x_{EV} = 6 \qquad (1)$$

$$x_{UA} + x_{UC} + x_{AB} + x_{BC} \le 3 \qquad (2)$$

$$x_{BC} + x_{BE} + x_{CE} \le 2 \qquad (3)$$

$$x_{UA} + x_{UB} + x_{AB} \le 2 \qquad (4)$$

$$0 \le x_{ij} \le 1$$



Z = 14

Solving the MST Problem

Minimise : $Z = 2x_{UA} + 5x_{UB} + 4x_{UC} + 2x_{AB} + 7x_{AD} + x_{BC} +$
$5x_{BD} + 3x_{BE} + 4x_{CE} + x_{DE} + 5x_{DV} + 7x_{EV}$

Subject to : $x_{UA} + x_{UB} + x_{UC} + x_{AB} + x_{AD} + x_{BC} + x_{BD} + x_{BE} + x_{CE} + x_{DE} + x_{DV} + x_{EV} = 6$ (1)

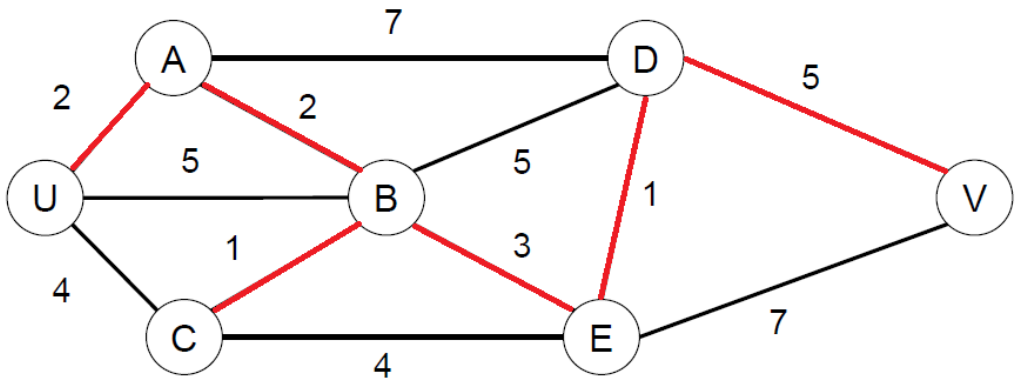
$x_{UA} + x_{UC} + x_{AB} + x_{BC} \le 3$ (2)

$x_{BC} + x_{BE} + x_{CE} \le 2$ (3)

$x_{UA} + x_{UB} + x_{AB} \le 2$ (4)

$x_{BD} + x_{BE} + x_{DE} \le 2$ (5)

$0 \le x_{ij} \le 1$



Z = 14

Minimum Spanning Tree Problem

- More efficient method maybe:
- Heuristics method!

Heuristic

- The word “heuristic” means “to find” or “to discover” by trial and error.
- A heuristic is a rule of thumb to guide decision making or to find approximate solutions to optimisation problems.
- Heuristic Search:
  - Heuristic search refers to the techniques with the aim of finding “good” solutions for a very hard optimisation and decision problems within a reasonable amount of computational time.
  - But, it inability to produce guaranteed optimal solutions. And it is difficult to know, in many cases, how far away from the optimum a heuristic solution might be…

Heuristics

- **Construction heuristics: builds a solution from scratch (starting with nothing).**
  - **Often called "greedy heuristics". Each step looks good, but it doesn't look ahead.**
- **Improvement heuristics (neighborhood search): starts with a solution, and then tries to improve the solution, usually by making small changes in the current solution.**

Two Greedy Algorithms

Kruskal's algorithm. Consider edges in ascending order of cost. Add the next edge to T unless doing so would create a cycle.

Prim's algorithm. Start with any vertex s and greedily grow a tree T from s. At each step, add the cheapest edge to T that has exactly one endpoint in T.

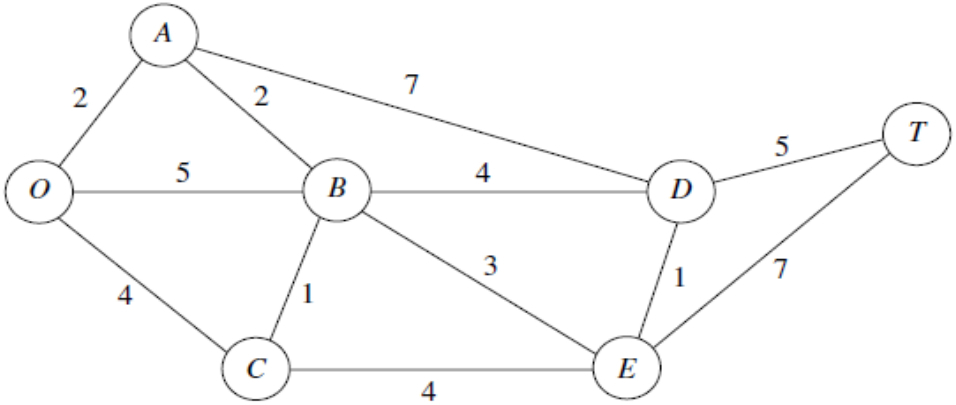
Theorem. Both greedy algorithms compute an MST.

> Greed is good. Greed is right. Greed works. Greed clarifies, cuts through, and captures the essence of the evolutionary spirit." - Gordon Gecko

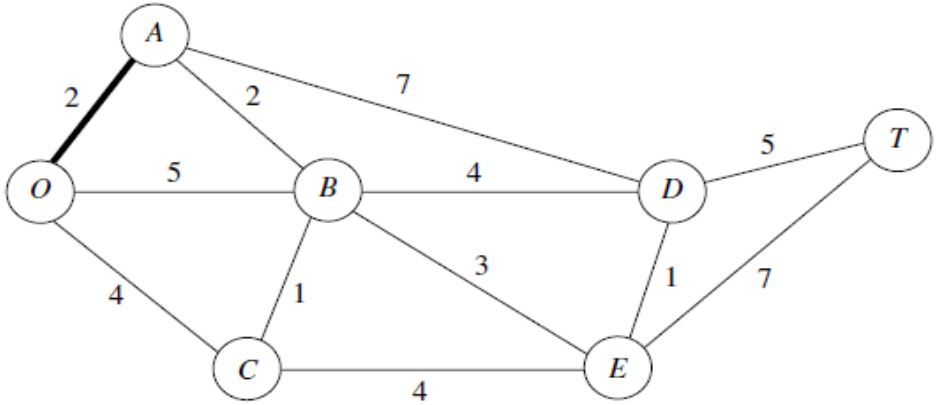
(greedy) Prim's algorithm

1. Select any node arbitrarily, and then connect it (i.e., add a link) to the nearest distinct node.
2. Identify the unconnected node that is closest to a connected node, and then connect these two nodes (i.e., add a link between them). Repeat this step until all nodes have been connected.
3. Tie breaking: Ties for the nearest distinct node (step 1) or the closest unconnected node (step 2) may be broken arbitrarily, and the algorithm must still yield an optimal solution. However, such ties are a signal that there may be (but need not be) multiple optimal solutions. All such optimal solutions can be identified by pursuing all ways of breaking ties to their conclusion.

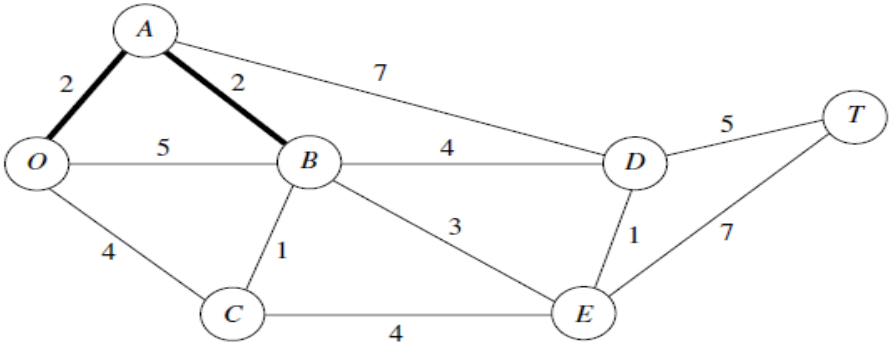
The fastest way of executing this algorithm manually is the graphical approach illustrated next.



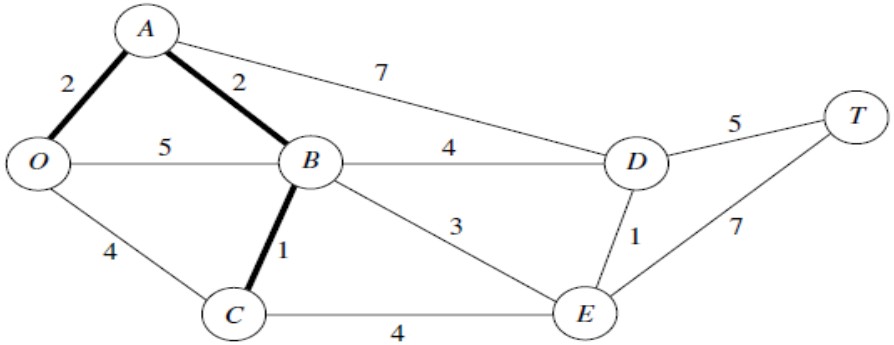
Arbitrarily select node O to start. The unconnected node closest to node O is node A. Connect node A to node O.



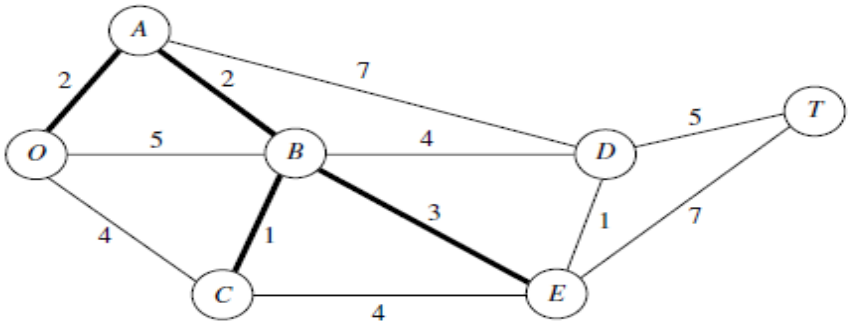
The unconnected node closest to either node O or node A is node B (closest to A). Connect node B to node A.



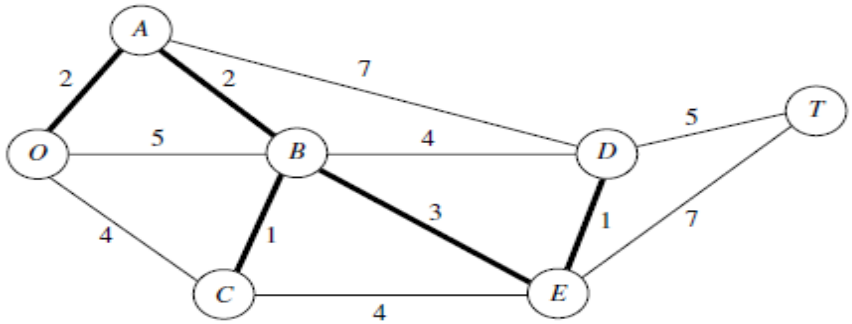
The unconnected node closest to node O, A, or B is node C (closest to B). Connect node C to node B.



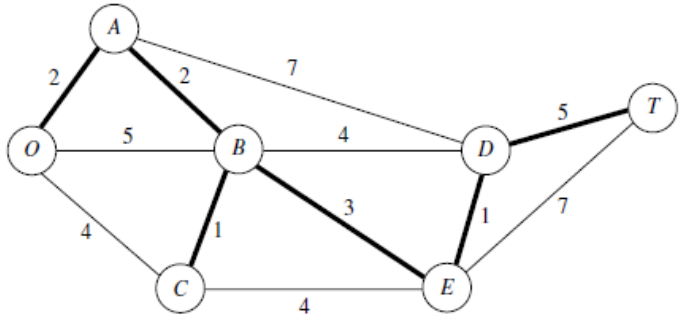
The unconnected node closest to node O, A, B, or C is node E (closest to B). Connect node E to node B.



The unconnected node closest to node O, A, B, C, or E is node D (closest to E). Connect node D to node E.



The only remaining unconnected node is node T. It is closest to node D. Connect node T to node D.



Real-world applications

- Identify which network models should be used to model these real-world problems:
  - Blood Bank Distribution: Blood donation centres have supplies of different blood types. Hospitals require specific amounts for surgeries and emergencies.
  - Energy Distribution: Power plants generate electricity. Cities / industrial zones consume electricity.
  - Cloud Computing / Data Server Allocation: Data centres (supply of computing resources) serve. Users / client servers (demand). The "cost" corresponds to latency, bandwidth usage, or energy consumption.

Real-world applications-continue

- Internet & Network Data Routing: Flow = data packets. Capacity = bandwidth of cables or routers. Determine how much data can be sent between two computers/data centres without overloading the network.
- Evacuation & Emergency Planning: Flow = number of people Capacity = corridor or exit width. Determine how many people can safely exit a stadium or building within a time limit.
- Emergency Response Planning: Fire trucks, ambulances, and police need the quickest possible route. The system considers: Traffic density, road closures and distance and speed limits etc.
- Electrical Power Grid Design: To build a power line network connecting multiple substations, we want minimum construction cost, minimum wire length, no unnecessary loops

- Q&A