Network Modeling

Overview

- Networks arise in numerous settings: transportation, electrical,
- and communication networks , for example.
- Network representations also are widely used for problems in such diverse areas as production, distribution, project planning, facilities location, resource management, and financial planning—to name just a few examples.
- Network representations provide a powerful visual and conceptual aid for portraying the relationships between the components of systems

Therminology of networks

- A network consists of a set of *points* and a set of *lines* connecting certain pairs of the points. The points are called **nodes** (or vertices).
- The lines are called **arcs** (or links or edges or branches)
  - If flow through an arc is allowed in only one direction the arc is said to be a **directed arc**
  - If flow through an arc is allowed in either direction the arc is said to be an **undirected arc**, frequently referred to **links.**

Therminology of networks

- A network that has only directed arcs is called a **directed network.**
- Similarly, if all its arcs are undirected, the network is said to be an **undirected network.**
- A network with a mixture of directed and undirected arcs (or even all undirected arcs) can be converted to a directed network, if desired, by replacing each undirected arc by a pair of directed arcs in opposite directions.
- A **path** between two nodes is a *sequence of distinct arcs* connecting these nodes.

Therminology of network

- A **directed path** from node *i* to node *j* is a sequence of connecting arcs whose direction (if any) is *toward* node *j*, so that flow from node *i* to node *j* along this path is feasible.
- An **undirected path** from node *i* to node *j* is a sequence of connecting arcs whose direction (if any) can be *either* toward or away from node *j*.
- A directed path also satisfies the definition of an undirected path, but not viceversa.
- Frequently, an undirected path will have some arcs directed toward node *j* but others directed away.

Therminology of network

- A path that begins and ends at the same node is called a **cycle.**
- In a *directed* network, a cycle is either a directed or an undirected cycle, depending on whether the path involved is a directed or an undirected path.
- Since a directed path also is an undirected path, a directed cycle is an undirected cycle.

Therminology of networks

- Two nodes are said to be **connected** if the network contains at least one *undirected* path between them.
- Note that the path does not need to be directed even if the network is directed.
- A **connected network** is a network where every pair of nodes is connected.

Therminology of network

- Consider a connected network with *n* nodes where all the arcs have been deleted.
- A “tree” can then be “grown” by adding one arc (or “branch”) at a time from the original network in a certain way.
- Each new arc creates a larger **tree,** which is a *connected network* (for some subset of the *n* nodes) that contains *no undirected cycles.*
- Once the (*n* - 1)st arc has been added, the process stops because the resulting tree *spans* (connects) all *n* nodes.
- This tree is called a **spanning tree,** i.e., a *connected network* for all *n* nodes that contains *no undirected cycles.*
- Every spanning tree has exactly *n* - 1 arcs, since this is the *minimum* number of arcs needed to have a connected network and the *maximum* number possible without having undirected cycles.

Therminology of network

- **Arc capacity** will be the maximum amount of flow (possibly infinity) that can be carried on a directed arc**.**
- A **supply node** (or source node or source) has the property that the flow *out* of the node exceeds the flow *into* the node.
- The reverse case is a **demand node** (or sink node or sink), where the flow *into* the node exceeds
- the flow *out* of the node.
- A **transshipment node** (or intermediate node) satisfies *conservation of flow,* so flow in equals flow out.

Typical problems

- Minimum Spanning tree
- Minimum cost
- Shortest path
- Maximum Flow
- Traveling salesman problem

Minimum spanning tree problem

- Given a connected graph $G = (V,E)$, with weight $c_{i,j}$ for all edge in E, find a spanning tree $G_T = (V_T, E_T)$ of minimum total weight.
- Given the *nodes* of a network, the *potential links* and the positive *length* for each
- Design a network by inserting enough links to satisfy the requirement that there be a path between *every* pair of nodes.
- The objective is to satisfy this requirement in a way that minimizes the total length of the links inserted into the network.

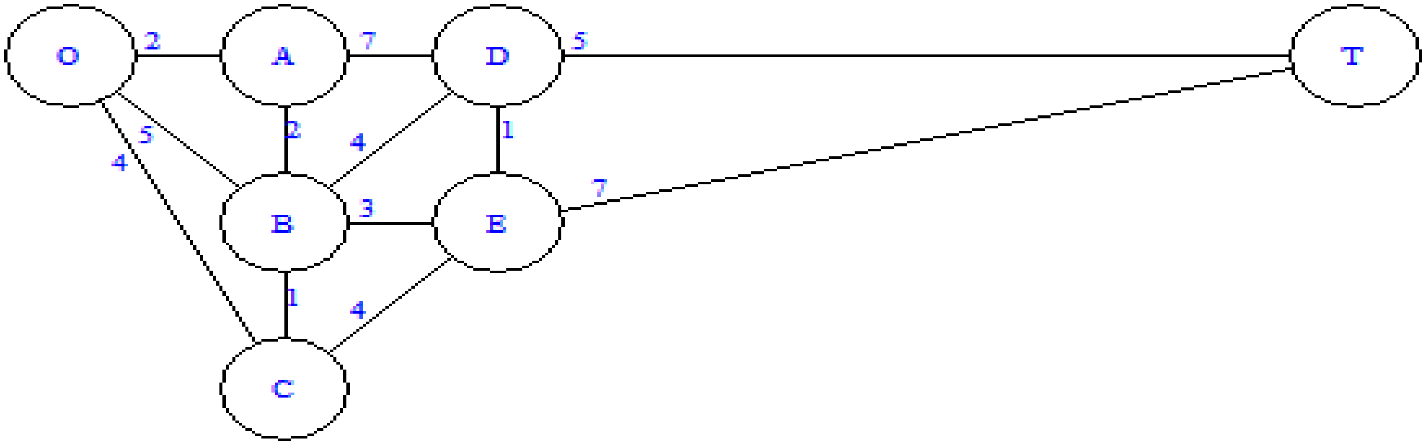
Algorithm

Algorithm for the Minimum Spanning Tree Problem.

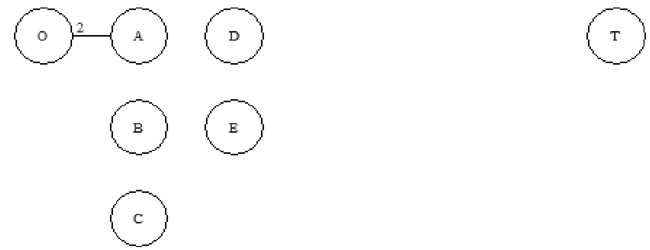
1. Select any node arbitrarily, and then connect it (i.e., add a link) to the nearest distinct node.
2. Identify the unconnected node that is closest to a connected node, and then connect these two nodes (i.e., add a link between them). Repeat this step until all nodes have been connected.
3. Tie breaking: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

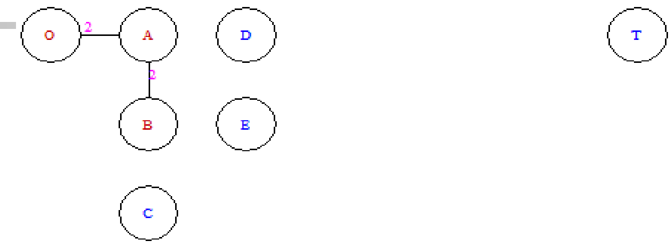
Example network in WinQSB

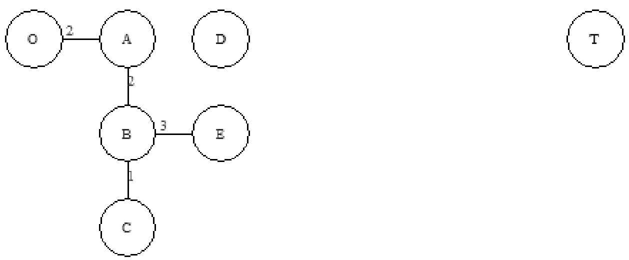
From \ To	O	A	B	C	D	E	T
O		2	5	4			
A			2		7		
B				1	4	3	
C						4	
D						1	5
E							7
T							

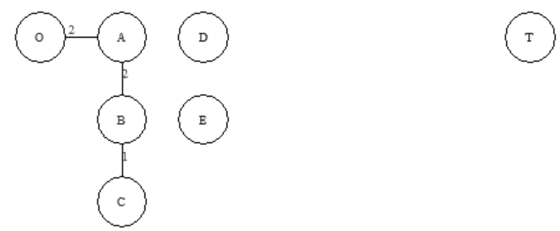


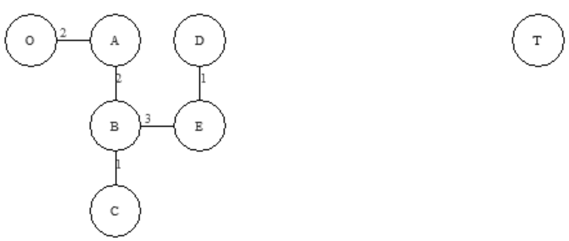
Steps for the algorithm

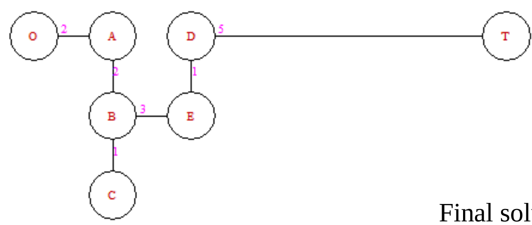












Final solution

From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
0	A	2	4	D	E	1
A	B	2	5	B	E	3
B	C	1	6	D	T	5
Total	Minimal	Connected	Distance	or Cost	=	14

LP formulation

$$\min Z = \sum_i \sum_j c_{i,j} x_{i,j}$$

Subject to:

$$\sum_v x_{i,j} = n - 1, \ \forall v \in V$$

$$\sum_{s \in S} x_{ij} \geq 1, \forall S = set\ of\ edges\ going\ from\ nodes\ in\ the\ subset\ \bar{V} \in V$$

$$x_{i,j} \begin{cases} 1, if\ edge\ from\ i\ to\ j\ exists \\ 0\ otherwise \end{cases}$$

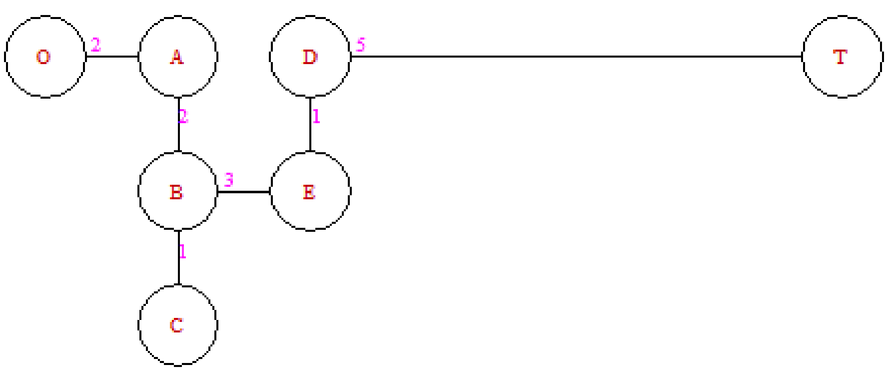
MPL formulation and solution

```
minimize

        minspanning=2OA+5OB+4OC+2AB+7AD+4BD+3BE+BC+4CE+5DT+7ET+DE;

SUBJECT TO

        OA+OB+OC+AD+AB+BD+BE+BC+CE+DT+ET+DE=6;
        OA+OB+OC>=1;
        OA+AB+AD>=1;
        OB+AB+BD+BE+BC>=1;
        OC+BC+CE>=1;
        AD+BD+DE+DT>=1;
        DE+BE+CE+ET>=1;
        DT+ET>=1;

BINARY
        OA OB OC AD AB BD BE BC CE DT ET DE;
END
```

```
MIN minspann =          14.0000

DECISION VARIABLES

PLAIN VARIABLES

Variable Name              Activity
----------------------------------------
OA                           1.0000
OB                           0.0000
OC                           0.0000
AB                           1.0000
AD                           0.0000
BD                           0.0000
BE                           1.0000
BC                           1.0000
CE                           0.0000
DT                           1.0000
ET                           0.0000
DE                           1.0000
```

Final solution

From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
O	A	2	4	D	E	1
A	B	2	5	B	E	3
B	C	1	6	D	T	5
Total	Minimal	Connected	Distance	or Cost	=	14



The minimum cost flow problem

- It holds a central position among network optimization models
  - It encompasses such a broad class of applications and because
  - It can be solved extremely efficiently
- The most important kind of application of minimum cost flow problems is to theoperation of a company's distribution network

Some applications

Kind of Application	Supply Nodes	Transshipment Nodes	Demand Nodes
Operation of a distribution network	Sources of goods	Intermediate storage facilities	Customers
Solid waste management	Sources of solid waste	Processing facilities	Landfill locations
Operation of a supply network	Vendors	Intermediate warehouses	Processing facilities
Coordinating product mixes at plants	Plants	Production of a specific product	Market for a specific product
Cash flow management	Sources of cash at a specific time	Short-term investment options	Needs for cash at a specific time

Hillier and Liebeman (2001)

Characteristics of the problem

- The network is a *directed and connected network.*
- *At least one of the nodes is a supply node.*
- *At least one of the other nodes is a demand node.*
- All the remaining nodes are *transshipment nodes.*
- Flow through an arc is allowed only in the direction indicated by the arrowhead, where the maximum amount of flow is given by the *capacity of that arc. (If flow can occur in* both directions, this would be represented by a pair of arcs pointing in opposite directions.)
- The network has enough arcs with sufficient capacity to enable all the flow generated at the *supply nodes to reach all the demand nodes.*
- The cost of the flow through each arc is *proportional to the amount of that flow, where* the cost per unit flow is known.
- The objective is to minimize the total cost of sending the available supply through the network to satisfy the given demand.

Formulation

- 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 j,*
- c_{ij} *= cost per unit flow through arc i j,*
- u_{ij} *= arc capacity for arc i j,*
- b_i *= net flow generated at node i.*
- The value of b_i *depends on the nature of node i, where*
- b_i *> if node i is a supply node,*
- b_i *< if node i is a demand node,*
- b_i *= 0 if node i is a transshipment node.*

$$\text{Minimize} \qquad Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij},$$

subject to

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = b_i, \qquad \text{for each node } i,$$

and

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

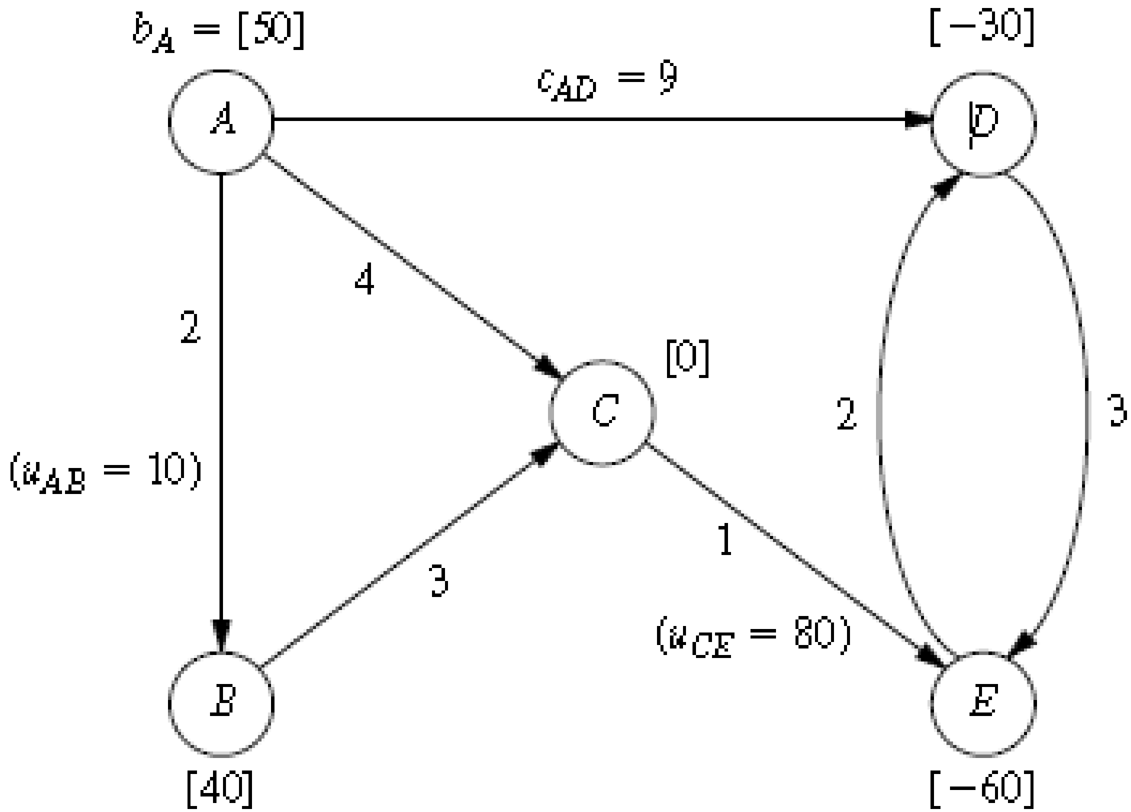
The first summation in the *node constraints represents the total flow out of node i, whereas* the second summation represents the total flow *into node i, so the difference is the net* flow generated at this node.

Feasible solutions property: A necessary condition for a minimum cost flow problem to have any feasible solutions is that

$$\sum_{i=1}^{n} b_i = 0.$$

That is, the total flow being generated at the supply nodes equals the total flow being absorbed at the demand nodes.

Example: formulate and solve the following problem



$$\text{Minimize} \quad Z = 2x_{AB} + 4x_{AC} + 9x_{AD} + 3x_{BC} + x_{CE} + 3x_{DE} + 2x_{ED},$$

subject to

$$\begin{array}{rrrrrrrcr}
x_{AB} & + x_{AC} & + x_{AD} & & & & & = & 50 \\
-x_{AB} & & & + x_{BC} & & & & = & 40 \\
& - x_{AC} & & - x_{BC} & + x_{CE} & & & = & 0 \\
& & - x_{AD} & & & + x_{DE} & - x_{ED} & = & -30 \\
& & & & - x_{CE} & - x_{DE} & + x_{ED} & = & -60
\end{array}$$

and

$$x_{AB} \leq 10, \qquad x_{CE} \leq 80, \qquad \text{all } x_{ij} \geq 0.$$

MPL formulation and solution

```
minimize

        Flowcost=2AB+4AC+9AD+3BC+CE+DE+2ED

subject to
        AB+AC+AD = 50;
        -AB + BC = 40;
        -AC-BC+CE = 0;
        -AD+DE-ED = -30;
        -CE-DE+ED=-60;
        AB<=10;
        CE<=80;
integer
        AB AC AD BC CE DE ED;
end
```

Optimal integer solution found

MIN Flowcost = 490.0000

DECISION VARIABLES

PLAIN VARIABLES

Variable Name	Activity	Reduced Cost
AB	0.0000	2.0000
AC	40.0000	4.0000
AD	10.0000	9.0000
BC	40.0000	3.0000
CE	80.0000	1.0000
DE	0.0000	1.0000
ED	20.0000	2.0000

Constraint Name	Slack	Shadow Price
c1	0.0000	0.0000
c2	0.0000	0.0000
c3	0.0000	0.0000
c4	0.0000	0.0000
c5	0.0000	0.0000
c6	10.0000	0.0000
c7	0.0000	0.0000

The shortest path problem

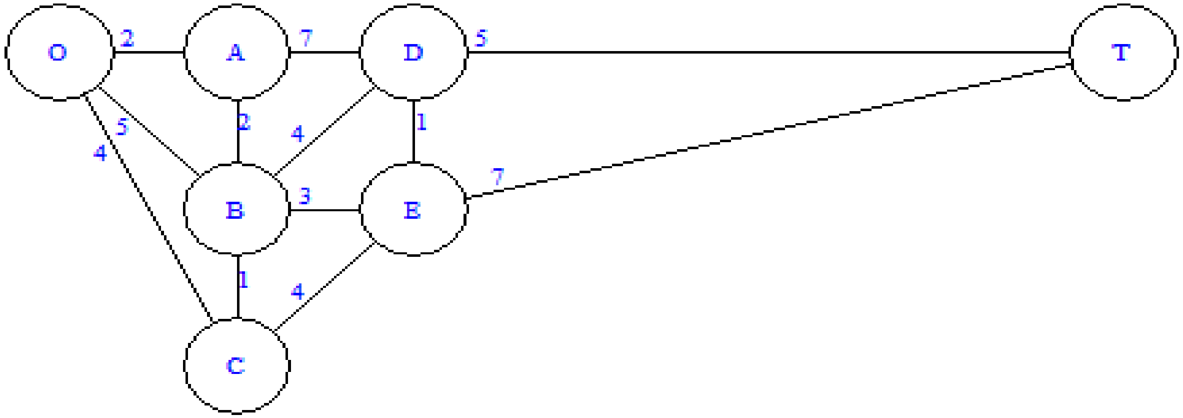
- Consider an *undirected* and *connected* network with two special nodes called the *origin* and the *destination.*
- Associated with each of the *links* (undirected arcs) is a nonnegative *distance.*
- The objective is to find the shortest path (the path with the minimum total distance) from the origin to the destination.

Algorithm

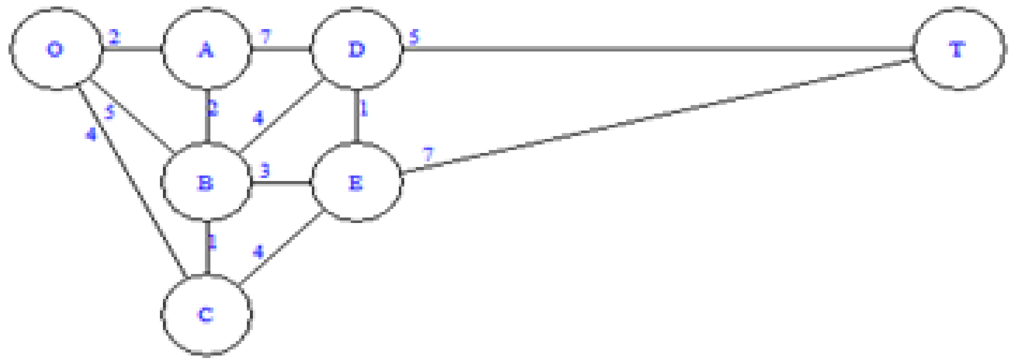
- *Objective of* n*th iteration:* Find the *n*th nearest node to the origin (to be repeated for *n* 1, 2, . . . until the *n*th nearest node is the destination.
- *Input for* n*th iteration: n* 1 nearest nodes to the origin (solved for at the previous iterations),including their shortest path and distance from the origin. (These nodes, plus the origin, will be called *solved nodes;* the othersare *unsolved nodes.*)
- *Candidates for* n*th nearest node:* Each solved node that is directly connected by a link to one or more unsolved nodes provides *one* candidate — the unsolved node with the *shortest* connecting link. (Ties provide additional candidates.)
- *Calculation of* n*th nearest node:* For each such solved node and its candidate, add the distance between them and the distance of the shortest path from the origin to this solved node.
- The candidate with the smallest such total distance is the *n*th nearest node (ties provide additional solved nodes), and its shortest path is the one generating this distance.

Example network

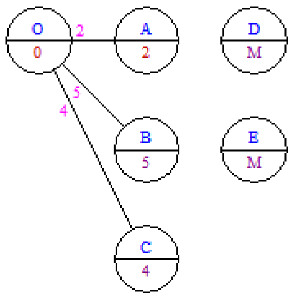
From \ To	O	A	B	C	D	E	T
O		2	5	4			
A			2		7		
B				1	4	3	
C						4	
D						1	5
E							7
T							



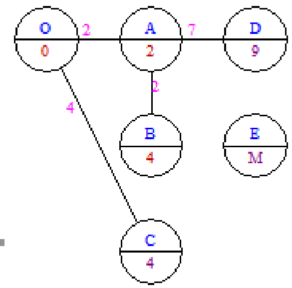
Solution process



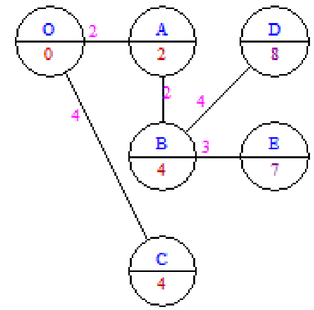
N	Solved node connected to an unsolved node	Closest unsolved node	Total distance	Nth. closest unsolved node	Minimum distance	Last connection
1	O	A	2	A	2	**OA**
2	O A	C B	4 2+2=4	C B	4 4	OC **AB**
3	A B C	D E E	2+7=9 4+3=7 4+4=8	E	7	- **BE** -
4	A B E	D D D	2+7=9 4+4=8 7+1=8	D D	8 8	- **BD** **ED**
5	D E	T T	8+5=13 7+7=14	T	13	**DT** -



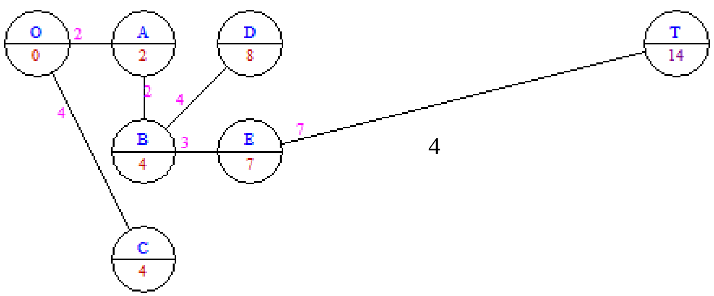
1



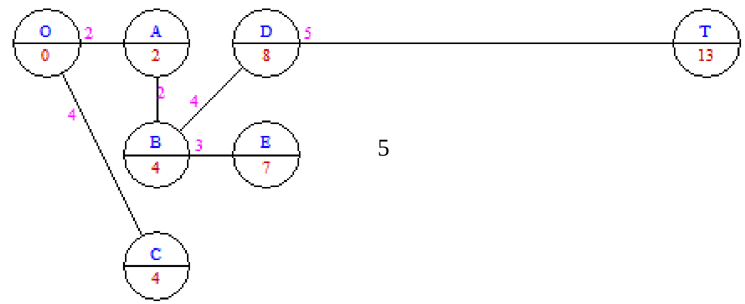
2



3



4



5

Final solution

From	To	Distance/Cost	Cumulative Distance/Cost
O	A	2	2
A	B	2	4
B	D	4	8
D	T	5	13
From O	To T	=	13

LP formulation

$$\min \sum_i \sum_j c_{ij} x_{ij}$$

Subject to

$$\sum_{arcs\ out} x_{ij} - \sum_{arcs\ in} x_{ij} = 1 \quad \text{Origin Node i}$$

$$\sum_{arcs\ out} x_{ij} - \sum_{arcs\ in} x_{ij} = 0 \quad \text{Intermediate nodes } \forall \text{i,j}$$

$$\sum_{arcs\ in} x_{ij} - \sum_{arcs\ out} x_{ij} = 1 \quad \text{Destination node j}$$

For unacceptable route add a new constraint

$$\sum x_{ij} = 0$$

$$x_{i,j} \begin{cases} 1, \text{if edge from } i \text{ to } j \text{ exists} \\ 0 \text{ otherwise} \end{cases}$$

MPL formulation and solution

```
title short_path;
minimize
        path=2OA+5OB+4OC+2AB+7AD+BC+4BD+3BE+4CE+DE+5DT+7ET;
subject to
        OA+OB+OC = 1;
        AD+AB-OA = 0;
        BC+BD+BE-OB-AB = 0;
        CE-BC-OC = 0;
        DT+DE+-AD-BD = 0;
        ET-BE-CE-DE=0;
        DT+DE = 1;
binary
        OA OB OC AB AD BC BD BE CE DE DT ET;
end
```

```
Optimal integer solution found

   MIN path              =          13.0000
```

DECISION VARIABLES

PLAIN VARIABLES

Variable Name	Activity	Reduced Cost
OA	1.0000	2.0000
OB	0.0000	5.0000
OC	0.0000	4.0000
AB	1.0000	2.0000
AD	0.0000	7.0000
BC	0.0000	1.0000
BD	1.0000	4.0000
BE	0.0000	3.0000
CE	0.0000	4.0000
DE	0.0000	1.0000
DT	1.0000	5.0000
ET	0.0000	7.0000

CONSTRAINTS

PLAIN CONSTRAINTS

Constraint Name	Slack	Shadow Price
c1	0.0000	0.0000
c2	0.0000	0.0000
c3	0.0000	0.0000
c4	0.0000	0.0000
c5	0.0000	0.0000
c6	0.0000	0.0000
c7	0.0000	0.0000

From	To	Distance/Cost	Cumulative Distance/Cost
O	A	2	2
A	B	2	4
B	D	4	8
D	T	5	13
From O	To T	=	13

The maximum flow problem

- For all flow through a directed and connected network originates at one node, called the **source,** and terminates at one other node, called the **sink.**
- All the remaining nodes are *transshipment nodes.*
- Flow through an arc is allowed only in the direction indicated by the arrowhead, where the maximum amount of flow is given by the *capacity* of that arc.
- At the *source,* all arcs point away from the node. At the *sink,* all arcs point into the node.
- The objective is to maximize the total amount of flow from the source to the sink.
- This amount is measured in either of two equivalent ways, namely, either the amount *leaving the source* or the amount *entering the sink.*

The algorithm: the augmented path algorithm

- It is based on two intuitive concepts, a *residual network* and an *augmenting path.*
- The **residual network** is the *remaining* arc capacity (called **residual capacity**) for assigning *additional* flows. Whenever some amount of flow is assigned to an arc, that amount is *subtracted* from the residual capacity in the same direction and *added* to the residual capacity in the opposite direction.

The algorithm: the augmented path algorithm

- An **augmenting path** is a directed path from the source to the sink in the residual network such that *every* arc on this path has *strictly positive* residual capacity.
- The *minimum* of these residual capacities is called the *residual capacity of the augmenting path* because it represents the amount of flow that can feasibly be added to the entire path.
- Therefore, each augmenting path provides an opportunity to further augment the flow through the original network.

The algorithm: the augmented path algorithm

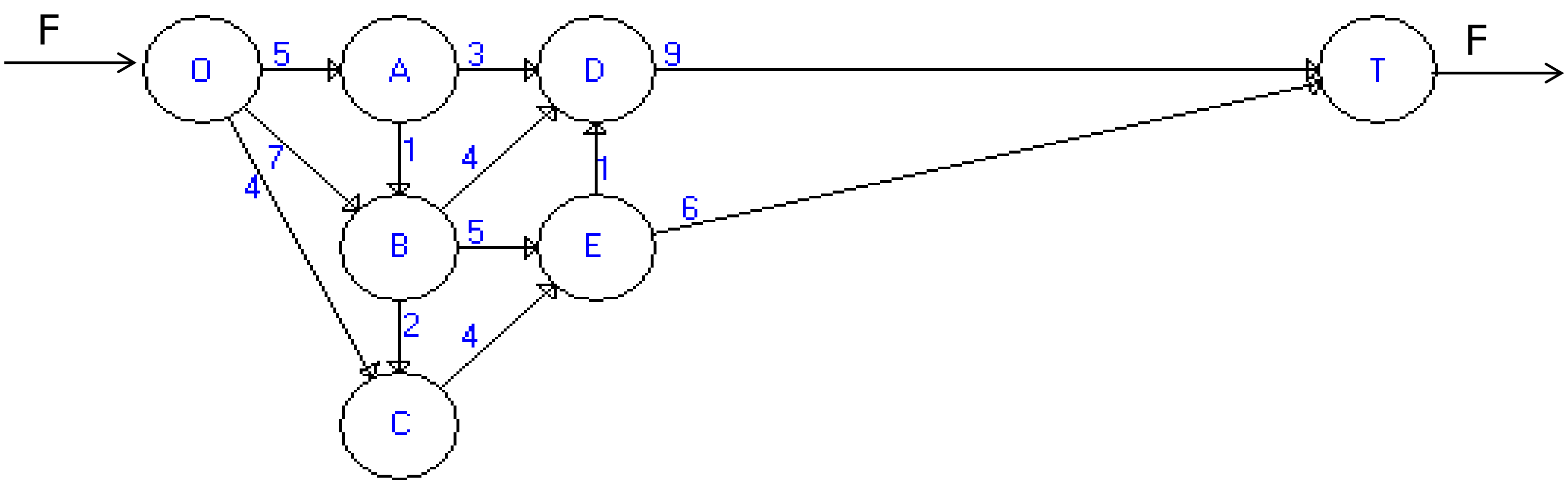
- The augmenting path algorithm repeatedly selects some augmenting path and adds a flow equal to its residual capacity to that path in the original network.
- This process continues until there are no more augmenting paths, so the flow from the source to the sink cannot be increased further.
- The key to ensuring that the final solution necessarily is optimal is the fact that augmenting paths can cancel some previously assigned flows in the original network, so an indiscriminate selection of paths for assigning flows cannot prevent the use of a better combination of flow assignments.

The algorithm: the augmented path algorithm

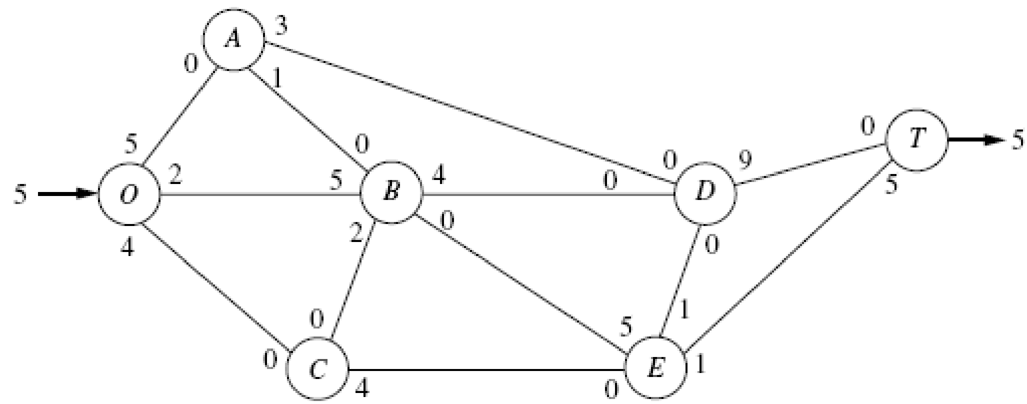
1. Identify an augmenting path by finding some directed path from the source to the sink in the residual network such that every arc on this path has strictly positive residual capacity. (If no augmenting path exists, the net flows already assigned constitute an optimal flow pattern.)
2. Identify the residual capacity c^* of this augmenting path by finding the *minimum* of the residual capacities of the arcs on this path. *Increase* the flow in this path by c^*.
3. *Decrease* by c^* the residual capacity of each arc on this augmenting path. *Increase* by c^* the residual capacity of each arc in the opposite direction on this augmenting path.
4. Return to step 1.

Network example

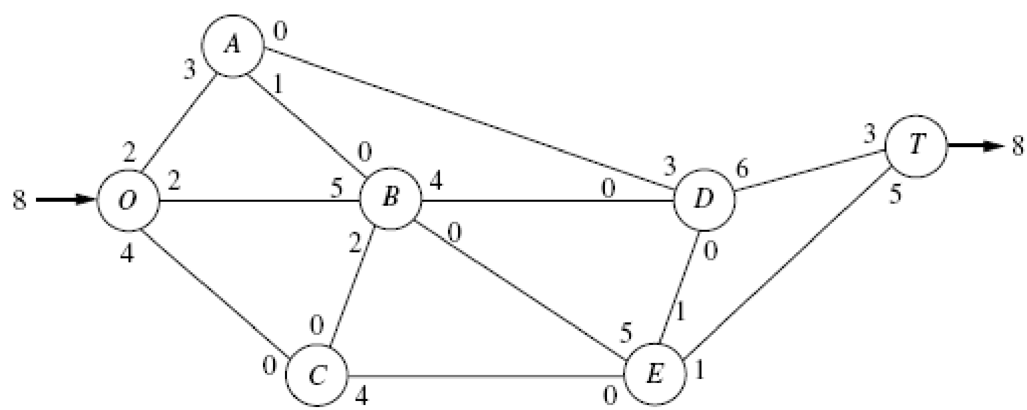
From \ To	O	A	B	C	D	E	T
O		5	7	4			
A			1		3		
B				2	4	5	
C						4	
D							9
E					1		6
T							



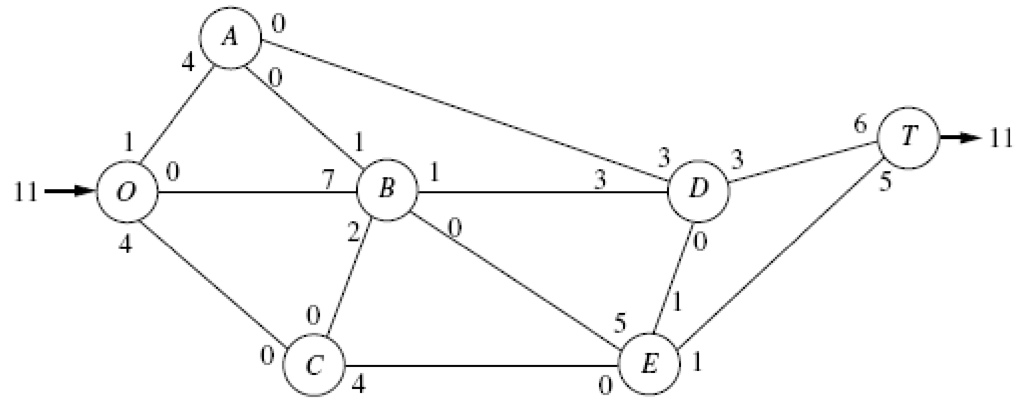
Iteration 1: In Fig. 9.7, one of several augmenting paths is $O \rightarrow B \rightarrow E \rightarrow T$, which has a residual capacity of $\min\{7, 5, 6\} = 5$. By assigning a flow of 5 to this path, the resulting residual network is



Iteration 2: Assign a flow of 3 to the augmenting path $O \rightarrow A \rightarrow D \rightarrow T$. The resulting residual network is

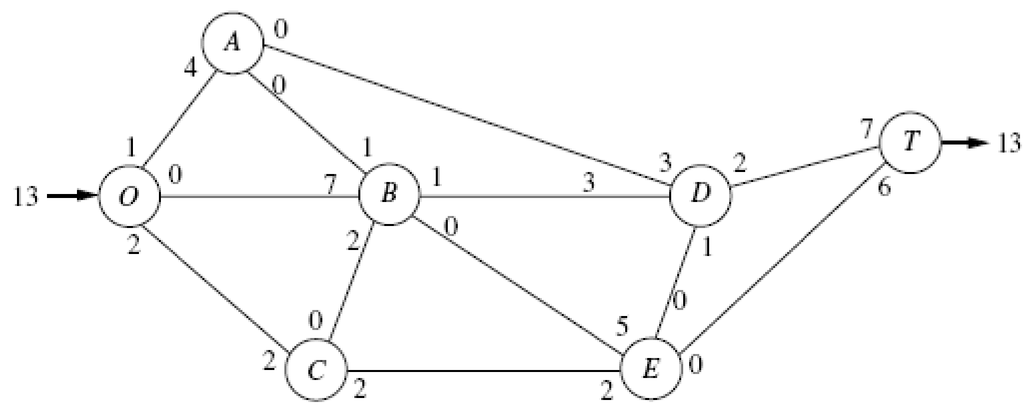


Iteration 4: Assign a flow of 2 to the augmenting path $O \to B \to D \to T$. The resulting residual network is



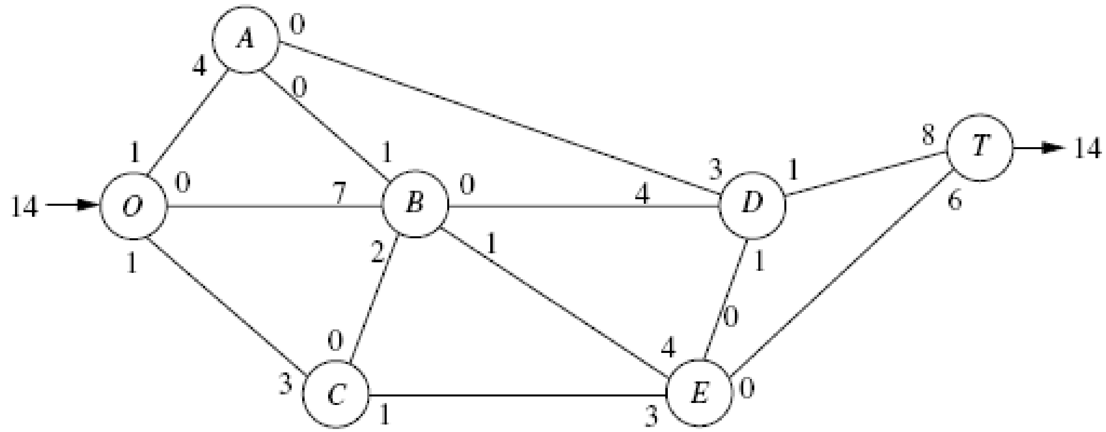
Iteration 5: Assign a flow of 1 to the augmenting path $O \to C \to E \to D \to T$.

Iteration 6: Assign a flow of 1 to the augmenting path $O \to C \to E \to T$. The resulting residual network is

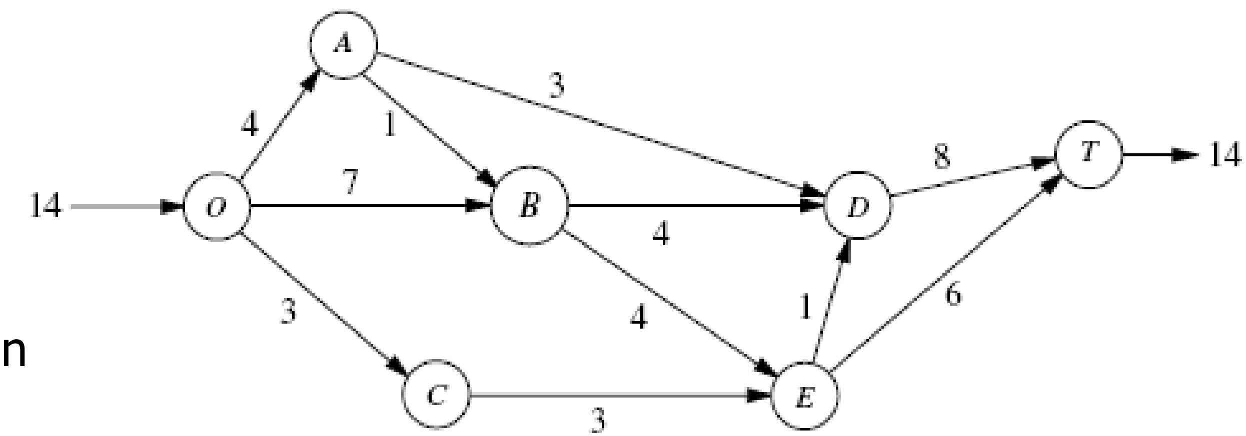


Iteration 7: Assign a flow of 1 to the augmenting path $O \to C \to E \to B \to D \to T$. The resulting residual network is

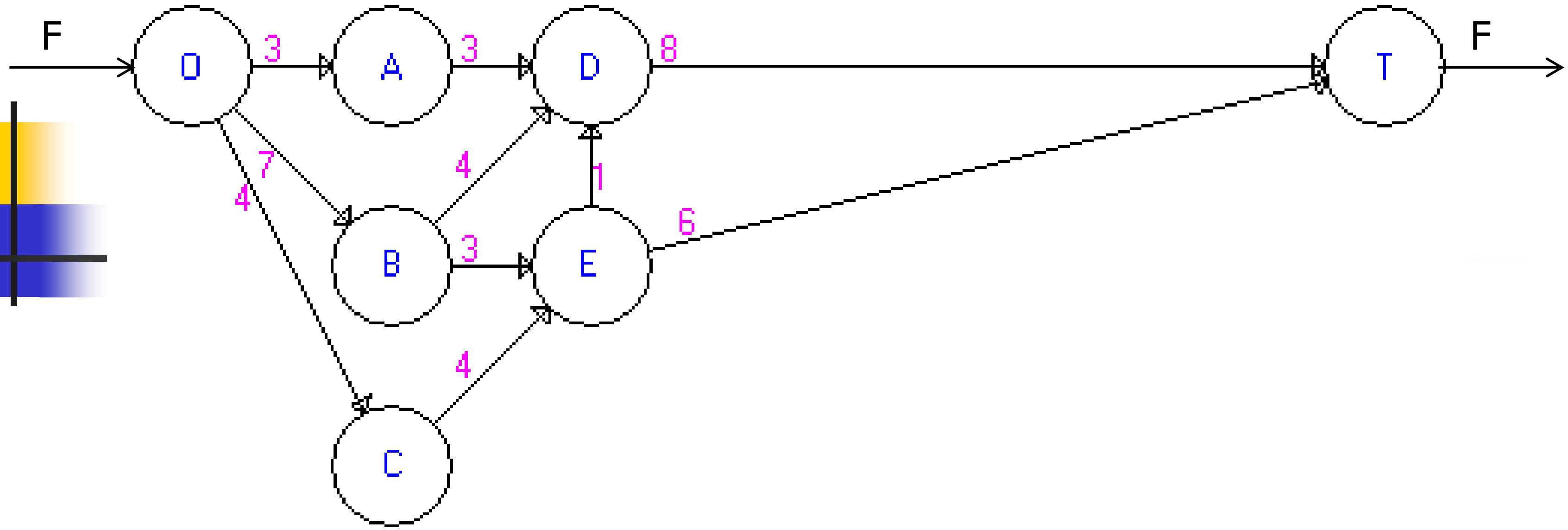
Iteration 7: Assign a flow of 1 to the augmenting path $O \to C \to E \to B \to D \to T$. The resulting residual network is



There are no more augmenting paths, so the current flow pattern is optimal.



Optimal solution



Alternate solution in QSB

From	To	Net Flow		From	To	Net Flow
O	A	3	6	B	E	3
O	B	7	7	C	E	4
O	C	4	8	D	T	8
A	D	3	9	E	D	1
B	D	4	10	E	T	6
Net Flow	From	O	To	T	=	14

LP formulation

$$\max F$$

$$\sum_{arcs\ out} x_{ij} - F = 0 \text{ at Origin}$$

$$\sum_{arcs\ out} x_{ij} - \sum_{arcs\ in} x_{ij} = 0 \text{ Intermediate nodes } \forall\, i$$

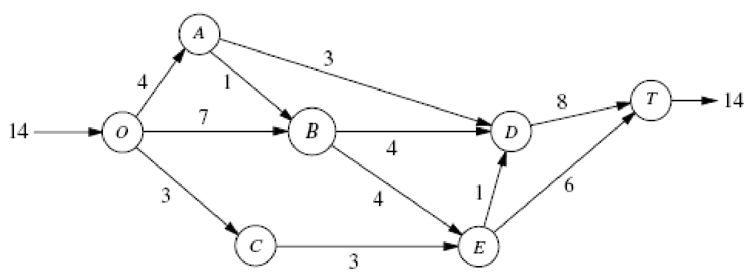
$$\sum_{arcs\ in} x_{ij} - F = 0 \text{ at Destinations}$$

$$x_{ij} \leq f_{ij}\ \forall \text{ nodes}$$

$$x_{ij} \geq 0$$

MPL formulation and solution

```
title max_flow;
maximize
        Flow = F;
subject to
        OA+OB+OC-F=0;
        AD+AB-OA=0;
        BC+BD+BE-OB-AB=0;
        CE-BC-OC=0;
        DT-AD-BD-ED=0;
        ET+ED-CE-BE=0;
        DT+ET-F=0;
        OA<=5;
        OB<=7;
        OC<=4;
        AB<=1;
        BC<=2;
        AD<=3;
        BD<=4;
        BE<=5;
        CE<=4;
        DT<=9;
        ED<=1;
        ET<=6;
integer
```



SOLUTION RESULT

Optimal integer solution found

MAX Flow = 14.0000

DECISION VARIABLES

PLAIN VARIABLES

Variable Name	Activity	Reduced Cost
F	14.0000	0.0000
OA	4.0000	1.0000
OB	7.0000	1.0000
OC	3.0000	1.0000
AD	3.0000	0.0000
AB	1.0000	0.0000
BC	0.0000	0.0000
BD	4.0000	0.0000
BE	4.0000	0.0000
CE	3.0000	0.0000
DT	8.0000	0.0000
ED	1.0000	0.0000
ET	6.0000	0.0000

Traveling salesman problem (TSP)

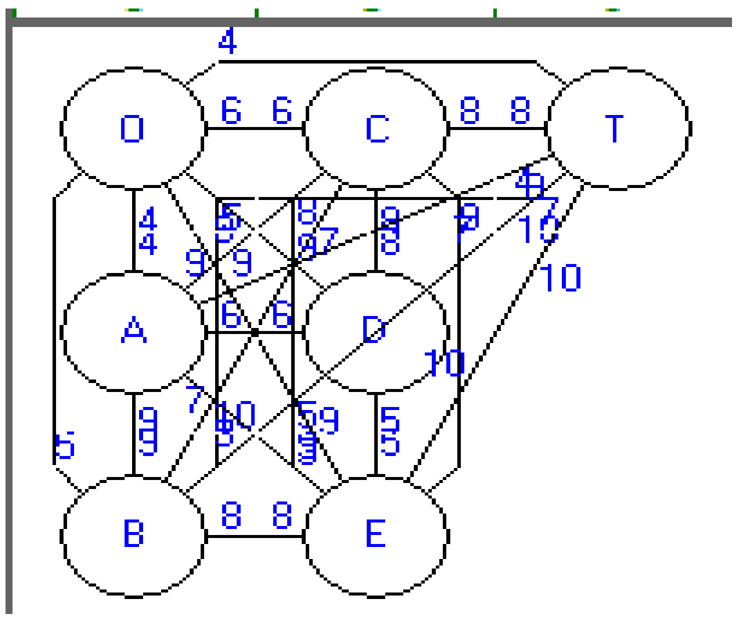
- It is an NP-hard problem in combinatorial optimization studied in operations research and theoretical computer science.
- Given a list of cities and their pairwise distances, the task is to find a shortest possible tour that visits each city exactly once.
- The problem was first formulated as a mathematical problem in 1930 and is one of the most intensively studied problems in optimization.
- It is used as a benchmark for many optimization methods.
- Even though the problem is computationally difficult, a large number of heuristics and exact methods are known, so that some instances with tens of thousands of cities can be solved.

As a network problem

- TSP can be modeled as an undirected weighted graph, such that cities are the graph's vertices, paths are the graph's edges, and a path's distance is the edge's length.
- In the *symmetric TSP*, the distance between two cities is the same in each opposite direction, forming an undirected graph. This symmetry halves the number of possible solutions.
- In the *asymmetric TSP*, paths may not exist in both directions or the distances might be different, forming a directed graph.

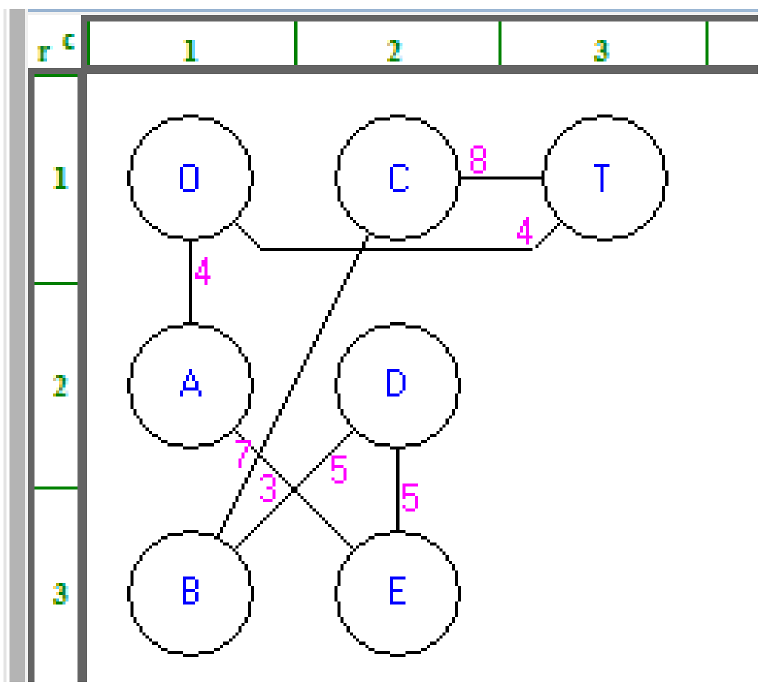
Network example: A symmetric network

From \ To	O	A	B	C	D	E	T
O		4	5	6	7	5	9
A	4		4	7	10	5	6
B	5	4		5	7	9	11
C	6	7	5		5	5	8
D	7	10	7	5		4	7
E	5	5	9	5	4		11
T	9	6	11	8	7	11	



One possible solution

11-21-2010	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	0	A	4	5	B	C	7
2	A	E	3	6	C	T	8
3	E	D	5	7	T	0	4
4	D	B	5				
	Total	Minimal	Traveling	Distance	or Cost	=	36
	(Result	from	Nearest	Neighbor	Heuristic)		

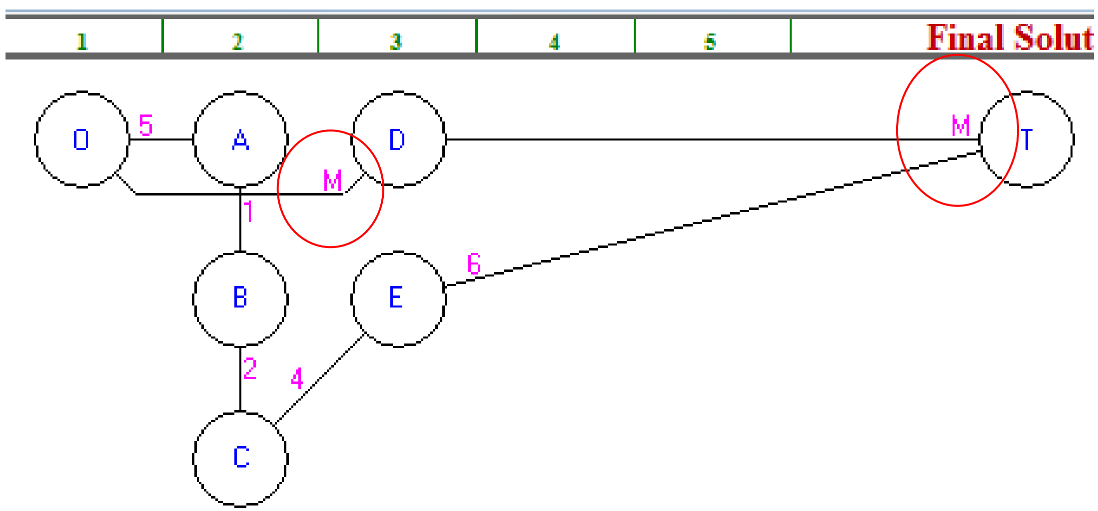


Network example: asymmetric

From \ To	O	A	B	C	D	E	T
O		5	7		7	7	9
A			1	4	3	6	8
B				2	4	5	6
C					9	4	4
D						7	9
E							6
T							

A possible solution

11-21-2010	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	0	A	5	5	E	T	6
2	A	B	1	6	T	D	M
3	B	C	2	7	D	0	M
4	C	E	4				C
	Total	Minimal	Traveling	Distance	or Cost	=	M
	(Result	from	Nearest	Neighbor	Heuristic)		



A generalized formulation

Minimize

$$Z_{TSP}(u) = \sum_{s \in R} \sum_{i \in M} \sum_{j \in (M \setminus \{i\})} c_{isj} u_{is} u_{j,s+1}$$

Subject to:

$$\sum_{i \in M} u_{is} = 1 \quad s \in S$$

$$\sum_{s \in S} u_{is} = 1 \quad i \in M$$

$$u_{is} \in \{0, 1\} \quad i \in M; \; s \in S$$