UNIT 3
MECHANICAL SYSTEM DESIGN

- *Chapter 5-Graph Modeling and analysis*
- *Chapter 6-Optimization Concept*

CHAPTER 5

Linear graph analysis

Linear Graph Modelling

The complete system model for a linear time-invariant system consists of (i) a set of n state equations, defined in terms of the matrices **A** and **B**, and (ii) a set of output equations that relate any output variables of interest to the state variables and inputs, and expressed in terms of the **C** and **D** matrices. The task of modeling the system is to derive the elements of the matrices, and to write the system model in the form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{1}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}. \tag{2}$$

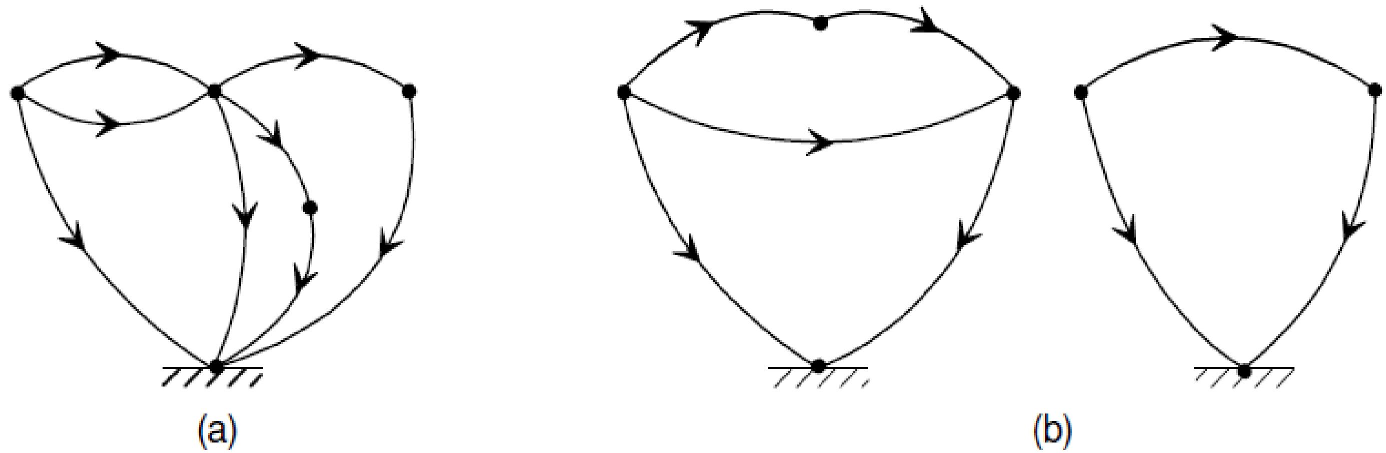
The matrices **A** and **B** are properties of the system and are determined by the system structure and elements. The output equation matrices **C** and **D** are determined by the particular choice of output variables.

The overall modeling procedure developed in this chapter is based on the following steps:

1. Determination of the system order n and selection of a set of state variables from the linear graph system representation.

2. Generation of a set of state equations and the system **A** and **B** matrices using a well defined methodology. This step is also based on the linear graph system description.

3. Determination of a suitable set of output equations and derivation of the appropriate **C** and **D** matrices.



Examples of (a) a connected system graph, and (b) an unconnected system graph.

The graph also represents the structure of the element interconnections, in terms of the continuity and compatibility constraint equations. In the following sections we use the properties of linear graphs to (i) derive the system structural constraints, (ii) define the set of state variables, and (iii) provide a systematic technique for deriving the system state equations

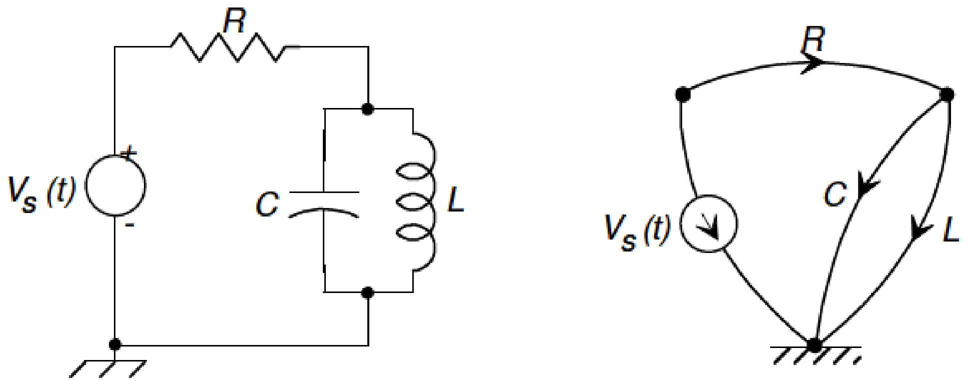


Figure 2: An electrical system and its linear graph

$$\frac{di_L}{dt} = \frac{1}{L}v_L$$
$$i_R = \frac{1}{R}v_R$$

giving three equations in six unknowns. The system structure defined by the linear graph imposes additional constraints; the three variables on the right-hand side may be eliminated by using (i) a continuity equation

$$i_C = i_R - i_L$$

to define i_C, and (ii) two compatibility equations

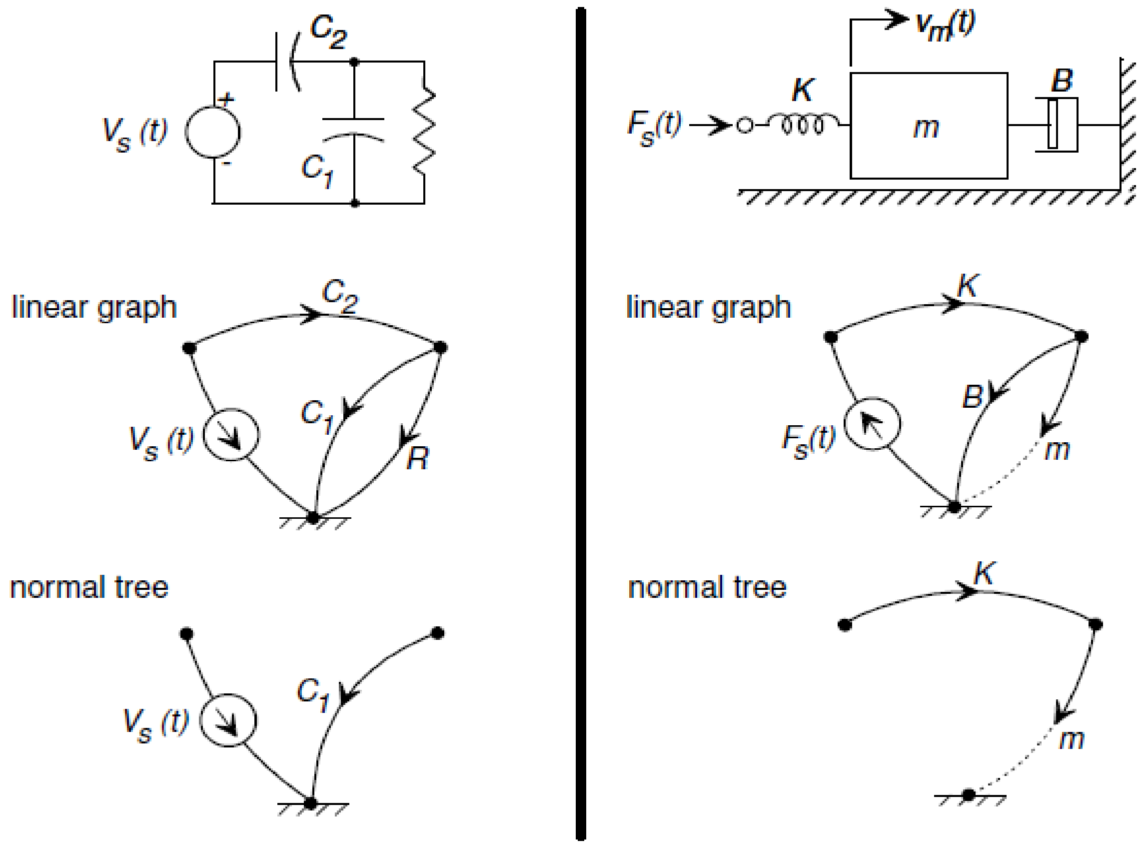


Figure 6: Two systems with dependent energy storage elements and their normal trees; (a) an electrical system with dependent capacitors, and (b) a mechanical system with a dependent spring.

DESCRIPTION

Many important optimization problems can be analyzed by means of graphical or network representation. In this chapter the following network models will be discussed:

1. Shortest path problems
2. Maximum flow problems
3. CPM-PERT project scheduling models
4. Minimum Cost Network Flow Problems
5. Minimum spanning tree problems

WHAT IS CPM/PERT FOR?

CPM/PERT are fundamental tools of project management and are used for one of a kind, often large and expensive, decisions such as building docks, airports and starting a new factory. Such decisions can be described via mathematical models, but this is not essential. Some would argue that CPM/PERT is not a pure OR topic. CPM/PERT really falls into gray area that can be claimed by fields other than OR also.

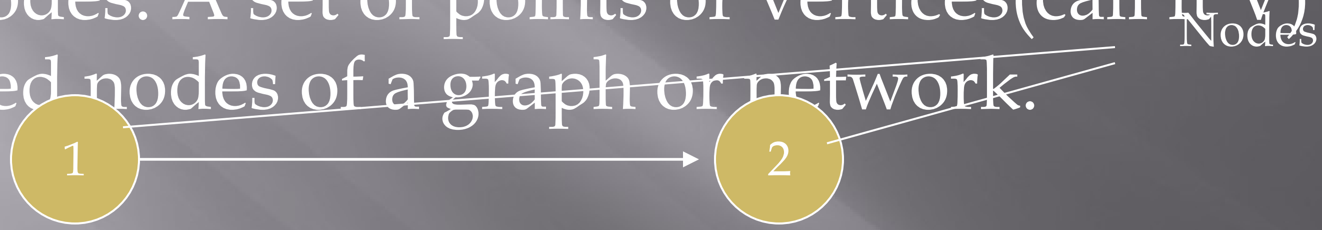
GENERAL COMMENTS ON CPM/PERT VS. ALB

- Assembly Line Balancing (ALB) are naturally not discussed in this text, but it is important to be aware of the huge difference between the ALB and CPM/PERT concepts because the precedence diagrams look so similar.
- Activity on node (AON) method of network precedence diagram drawing (not used in this chapter) and the ALB diagram are identical looking at first. The ALB deals with small repetitive items such as TV's while CPM/PERT deals with large one of a kind projects.

5.1 BASIC DEFINITIONS

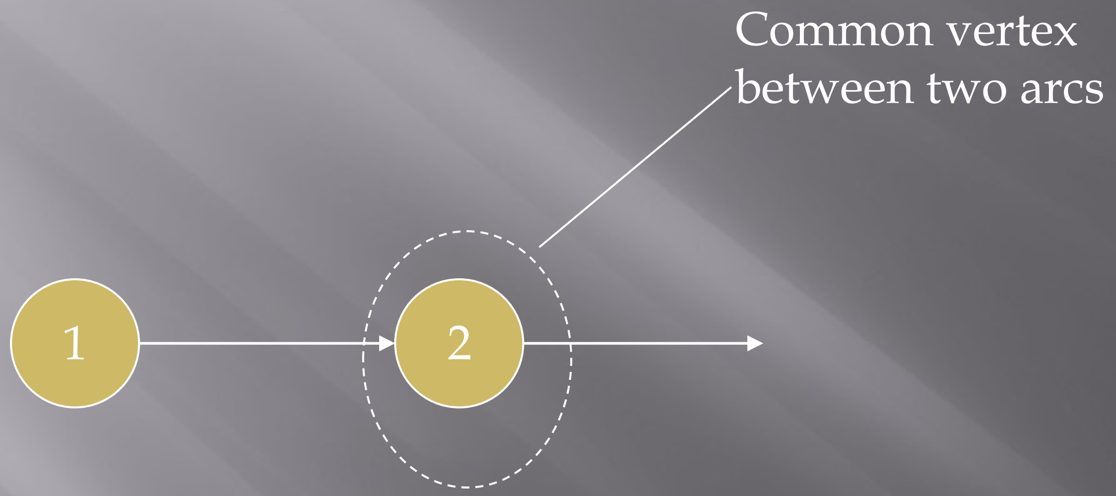
A graph or network is defined by two sets of symbols:

- Nodes: A set of points or vertices(call it V) are called nodes of a graph or network.



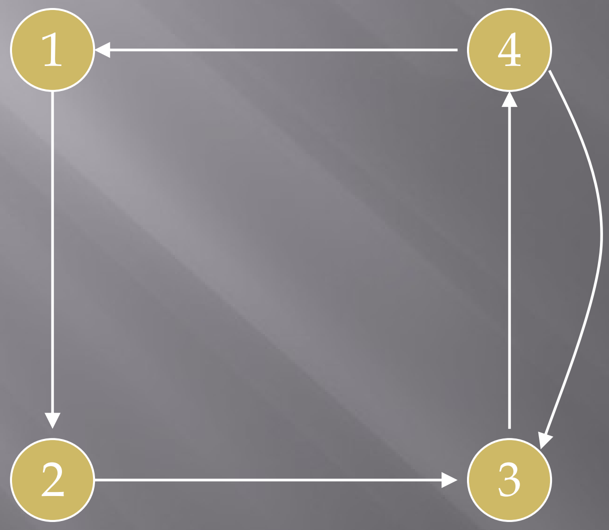
- Arcs: An arc consists of an ordered pair of vertices and represents a possible direction of motion that may occur between vertices.

• Chain: A sequence of arcs such that every arc has exactly one vertex in common with the previous arc is called a chain.



- Path: A path is a chain in which the terminal node of each arc is identical to the initial node of next arc.

For example in the figure below (1,2)-(2,3)-(4,3) is a chain but not a path; (1,2)-(2,3)-(3,4) is a chain and a path, which represents a way to travel from node 1 to node 4.



5.2 SHORTEST PATH PROBLEMS

Assume that each arc in the network has a length associated with it. Suppose we start with a particular node. The problem of finding the shortest path from node 1 to any other node in the network is called a shortest path problem. The general structure and solution methods of a shortest path problem will be shown in the following example.

CAR (OR MACHINE) REPLACEMENT EXAMPLE:

Let's assume that we have just purchased a new car (or machine) for $12,000 at time 0. The cost of maintaining the car during a year depends on the age of the car at the beginning of the year, as given in the table below.

Age of Car (Years)	Annual Maintenance cost	Age of Car (Years)	Trade-in Price
0	$2,000	1	$7,000
1	$4,000	2	$6,000
2	$5,000	3	$2,000
3	$9,000	4	$1,000
4	$12,000	5	$0

In order to avoid the high maintenance cost associated with an older car, we may trade in the car and purchase a new car. The trade-in prices are also given in the table. To simplify the computations we assume that at any time it costs $12,000 to purchase a new car. Our goal is to minimize the net cost incurred during the next five years. Let's formulate this problem as a shortest path problem. Our network will have six nodes. Node i is the beginning of year i and for $i<j$, an arc (i,j) corresponds to purchasing a new car at the beginning of year i and keeping it until the beginning of year j. The length of arc (i,j) (call it c_{ij}) is the total net cost incurred from year i to j.

Thus:

c_{ij} = maintenance cost incurred during years $i, i+1, \ldots, j-1$

+cost of purchasing a car at the beginning of year i

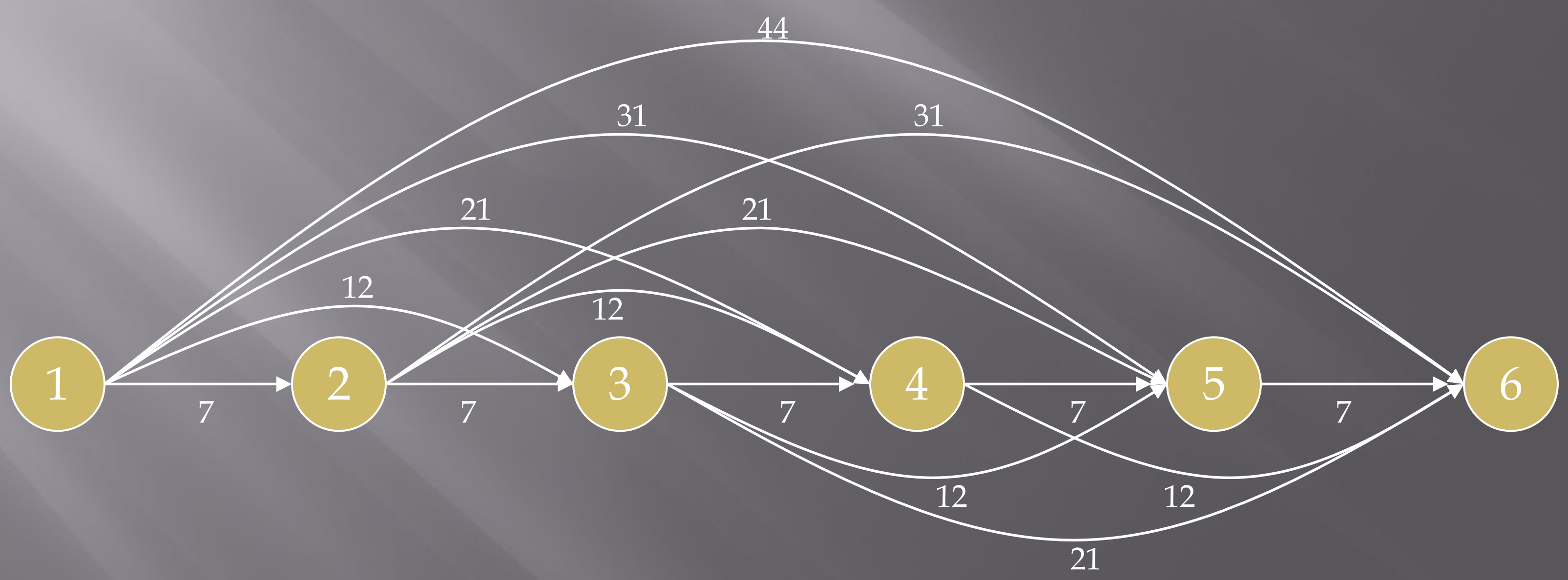
-trade-in value received at the beginning of year j.

Applying this formula to the information we will obtain the following:

c_{12}=2+12-7=7	c_{13}=2+4+12-6=12	c_{14}=2+4+5+12-2=21
c_{15}=2+4+5+9+12-1=31	c_{13}=2+4+5+9+12+12-0=44	c_{23}=2+12-7=7
c_{24}=2+4+12-6=12	c_{25}=2+4+5+12-2=21	c_{26}=2+4+5+9+12-1=31
c_{34}=2+12-7=7	c_{35}=2+4+12-6=12	c_{36}=2+4+5+12-2=21
c_{45}=2+12-7=7	c_{46}=2+4+12-6=12	c_{56}=2+12-7=7

NETWORK FOR MINIMIZING CAR COSTS

From the figure below we can see that both path 1-3-5-6 and 1-2-4-6 will give us the shortest path with a value of 31.

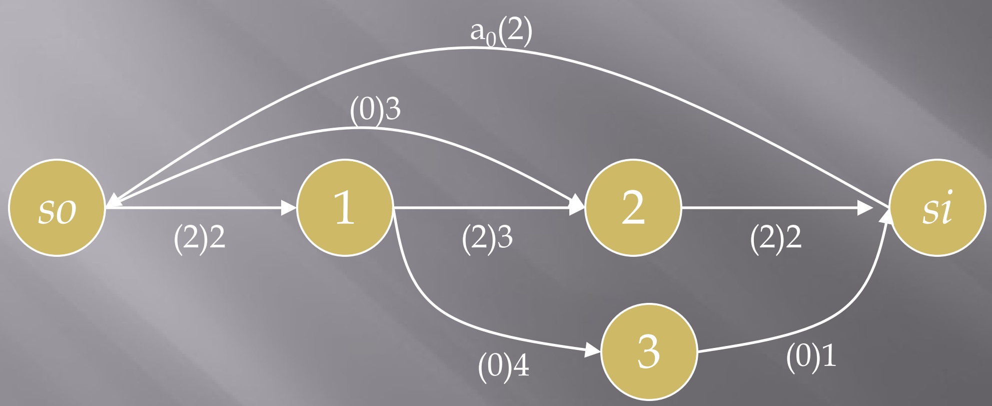


5.3 MAXIMUM FLOW PROBLEMS

Many situations can be modeled by a network in which the arcs may be thought of as having a capacity that limits the quantity of a product that may be shipped through the arc. In these situations, it is often desired to transport the maximum amount of flow from a starting point (called the source) to a terminal point (called the sink). Such problems are called maximum flow problems.

AN EXAMPLE FOR MAXIMUM FLOW PROBLEM

Sunco Oil wants to ship the maximum possible amount of oil (per hour) via pipeline from node *so* to node *si* as shown in the figure below.



Arc	*Capacity*
(*so*,1)	2
(*so*,2)	3
(1,2)	3
(1,3)	4
(3,*si*)	1
(2,*si*)	2

The various arcs represent pipelines of different diameters. The maximum number of barrels of oil that can be pumped through each arc is shown in the table above (also called arc capacity).

For reasons that will become clear soon, an artificial arc called a_0 is added from the sink to the source. To formulate an LP about this problem first we should determine the decision variable.

X_{ij} = Millions of barrels of oil per hour that will pass through arc(i,j) of pipeline.

For a flow to be feasible it needs to be in the following range:

0 <= flow through each arc <= arc capacity

And

Flow into node i = Flow out from node i

Let X_0 be the flow through the artificial arc, the conservation of flow implies that X_0 = total amount of oil entering the sink. Thus, Sunco's goal is to maximize $X_{0.}$

Max Z= X_0

S.t. $X_{so,1}<=2$ *(Arc Capacity constraints)*

$X_{so,2}<=3$

$X_{12}<=3$

$X_{2,si}<=2$

$X_{13}<=4$

$X_{3,si}<=1$

$X_0=X_{so,1}+X_{so,2}$ *(Node so flow constraints)*

$X_{so,1}=X_{12}+X_{13}$ *(Node 1 flow constraints)*

$X_{so,2}+X_{12}=X_{2,si}$ *(Node 2 flow constraints)*

$X_{13}+X_{3,si}$ *(Node 3 flow constraints)*

$X_{3,si}+X_{2,si}=X_0$ *(Node si flow constraints)*

$X_{ij}>=0$

One optimal solution to this LP is Z=3, $X_{so,1}=2$, $X_{13}=1$, $X_{12}=1$, $X_{so,2}=1$, $X_{3,si}=1$, $X_{2,si}=2$, $X_{o}=3$.

5.4 CPM AND PERT

Network models can be used as an aid in the scheduling of large complex projects that consist of many activities.

CPM: If the duration of each activity is known with certainty, the Critical Path Method (CPM) can be used to determine the length of time required to complete a project.

PERT: If the duration of activities is not known with certainty, the Program Evaluation and Review Technique (PERT) can be used to estimate the probability that the project will be completed by a given deadline.

CPM AND PERT ARE USED IN MANY APPLICATIONS INCLUDING THE FOLLOWING:

- Scheduling construction projects such as office buildings, highways and swimming pools
- Developing countdown and “hold” procedure for the launching of space crafts
- Installing new computer systems
- Designing and marketing new products
- Completing corporate mergers
- Building ships

To apply CPM and PERT, we need a list of activities that make up the project. The project is considered to be completed when all activities have been completed. For each activity there is a set of activities (called the predecessors of the activity) that must be completed before the activity begins. A project network is used to represent the precedence relationships between activities. In the following discussions the activities will be represented by arcs and the nodes will be used to represent completion of a set of activities (Activity on arc (AOA) type of network).

Activity A must be completed before activity B starts

While constructing an AOA type of project diagram one should use the following rules:

- Node 1 represents the start of the project. An arc should lead from node 1 to represent each activity that has no predecessors.
- A node (called the finish node) representing the completion of the project should be included in the network.
- Number the nodes in the network so that the node representing the completion time of an activity always has a larger number than the node representing the beginning of an activity.
- An activity should not be represented by more than one arc in the network
- Two nodes can be connected by at most one arc.

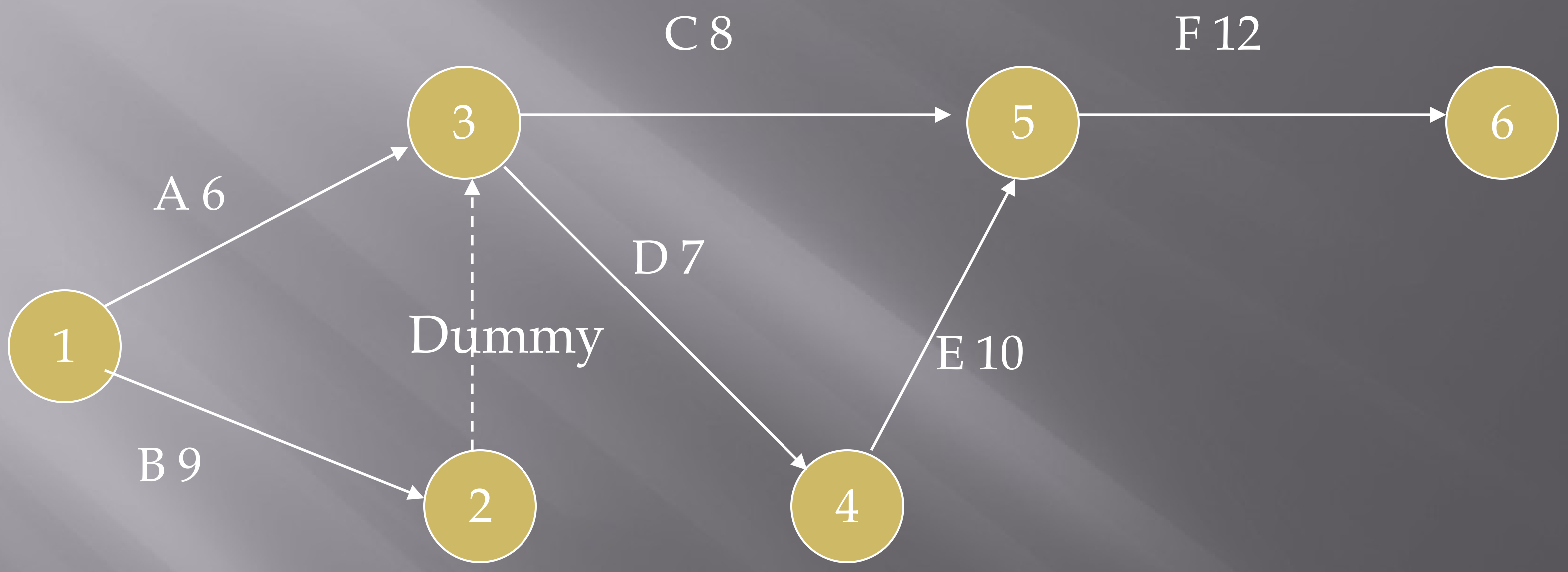
To avoid violating rules 4 and 5, it can be sometimes necessary to utilize a dummy activity that takes zero time.

AN EXAMPLE FOR CPM

Widgetco is about to introduce a new product. A list of activities and the precedence relationships are given in the table below. Draw a project diagram for this project.

Activity	Predecessors	Duration(days)
A:train workers	-	6
B:purchase raw materials	-	9
C:produce product 1	A, B	8
D:produce product 2	A, B	7
E:test product 2	D	10
F:assemble products 1&2	C, E	12

PROJECT DIAGRAM FOR WIDGETCO



Node 1 = starting node
Node 6 = finish node

IMPORTANT DEFINITIONS FOR COMPUTATION

Early Event Time: The early event time for node i, represented by ET(i), is the earliest time at which the event corresponding to node i can occur.

Late Event Time: The late event time for node i, represented by LT(i), is the latest time at which the event corresponding to node i can occur without delaying the completion of the project.

COMPUTATION OF EARLY EVENT TIME

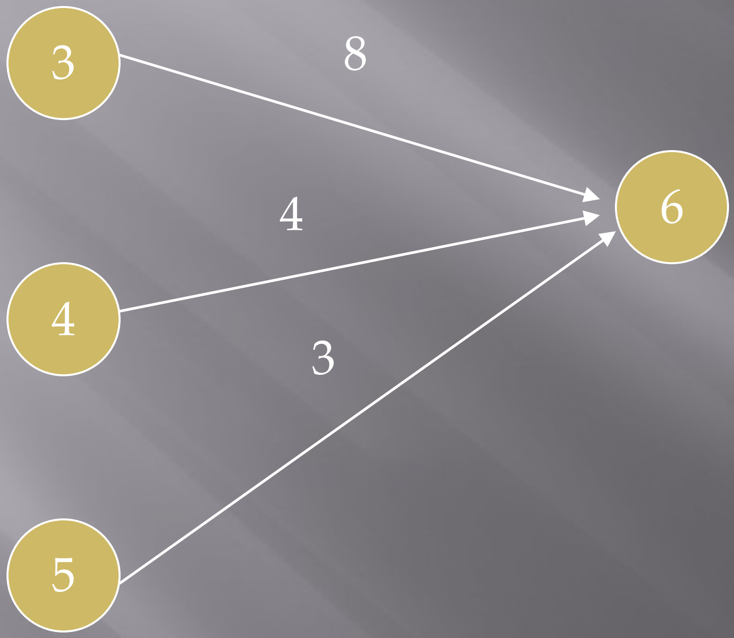
To find the early event time for each node in the project network we follow the steps below:

- Find each prior event to node *i* that is connected by an arc to node *i*. These events are immediate predecessors of node *i*.
- To the *ET* for each immediate predecessor of the node *i* add the duration of the activity connecting the immediate predecessor to node *i*.
- *ET(i)* equals the maximum of the sums computed in previous step.

Computing *ET(i)*. We start with the beginning node

ET(1)=0 (The starting node is always 0)

Let's say ET(3)=6
ET(4)=8, ET(5)=10



$$ET(6) = \max \begin{cases} ET(3) + 8 = 14 \\ ET(4) + 4 = 12 \\ ET(5) + 3 = 13 \end{cases}$$

ET(6)=14

COMPUTATION OF LATE EVENT TIME

To compute LT(i) we begin with the finish node and go backwards and we follow the steps below:

- Find each node that occurs after node *i* and is connected to node *i* by an arc. These events are immediate successors of node *i*.
- From the *LT* for each immediate successor to node *i* subtract the duration of the activity joining the successor the node *i*.
- *LT(i)* is the smallest of the differences determined in previous step.

Let's compute LT(I)'s for the Widgetco example:

From the graph we know that the late completion time for node 6 is 35. Since node 6 is the only immediate successor of node 6, LT(5)=LT(6)-12=26. Same way LT(4)= LT(5)-10=16.

Nodes 4 and 5 are the immediate successors of node 3. Thus;

$$LT(3) = \min \begin{cases} LT(4) - 7 = 9 \\ LT(5) - 8 = 18 \end{cases}$$

$$\mathrm{LT}(2) = \mathrm{LT}(3) - 0 = 9$$

$$LT(1) = \min \begin{cases} LT(2) - 9 = 0 \\ LT(3) - 6 = 3 \end{cases}$$

Total Float

- For an arbitrary arc representing activity (*i,j*), the total float, represented by *TF*(*i,j*), of the activity represented by (*i,j*) is the amount by which the starting time of activity (*i,j*) could be delayed beyond its earliest possible starting time without delaying the completion of the project (assuming no other activities are delayed).

TOTAL FLOAT

Before the project is begun, the duration of each activity is unknown, and the duration of each activity used to construct the project network is just an estimate of the activity's actual completion time. The concept of total float of an activity can be used as a measure of how important it is to keep each activity's duration from greatly exceeding our estimate of its completion time. Total float is represented by TF(i,j) and equals:

$$ET(I) + k + t_{ij} <= LT(j) \text{ or } k <= LT(j) - ET(i) - t_{ij}$$

$$=> TF(i,j)=LT(j)-ET(i)-t_{ij}$$

For Widgetco example ET(i)’s and LT(i)’s are as follows:

Node	ET(i)	LT(i)
1	0	0
2	9	9
3	9	9
4	16	16
5	26	26
6	38	38

According to the table on the previous slide the TF(i,j)'s are computed as follows:

Activity B: TF(1,2)=LT(2)-ET(1)-9=0

Activity A: TF(1,3)=LT(3)-ET(1)-6=3

Activity D: TF(3,4)=LT(4)-ET(3)-7=0

Activity C: TF(3,5)=LT(5)-ET(3)-8=9

Activity E: TF(4,5)=LT(5)-ET(4)-10=0

Activity F: TF(5,6)=LT(6)-ET(5)-12=0

Dummy activity: TF(2,3)=LT(3)-ET(2)-0=0

CRITICAL PATH

- An activity with a total float of zero is a **critical activity**
- A path from node 1 to the finish node that consists entirely of critical activities is called a **critical path**.

For Widgetco example 1-2-3-4-5-6 is a critical path.

FREE FLOAT

Free float is the amount by which the starting time of the activity corresponding to arc(i,j) can be delayed without delaying the start of any later activity beyond the earliest possible starting time.

Suppose the occurrence of node *i*, or the duration of activity (*i,j*), is delayed by k units. Then the earlist that node j can occur is $ET(i) + t_{ij} + k$. Thus, if $ET(I) + t_{ij} + k <= ET(j)$, or $k <= ET(j) – ET(I) – t_{ij}$, then node *j* will not be delayed. If node j is not delayed, then no other activities will be delayed beyond their earliest possible starting times. Therefore,

$$FF(i,j)=ET(j)-ET(i)-t_{ij}$$

According to the table on the previous slide the FF(i,j)’s are computed as follows:

Activity B: FF(1,2)=9-0-9=0
Activity A: FF(1,3)=9-0-6=3
Activity D: FF(3,4)=16-9-7=0
Activity C: FF(3,5)=26-9-8=9
Activity E: FF(4,5)=26-16-10=0
Activity F: FF(5,6)=38-26-12=0

For example a delay up to 9 days in the start or duration of activity C will not delay the start of later activities.

USING LP TO FIND A CRITICAL PATH

Decision variable:

X_{ij}:the time that the event corresponding to node j occurs

Since our goal is to minimize the time required to complete the project, we use an objective function of:

$$Z=X_F-X_1$$

Note that for each activity (i,j), before j occurs , i must occur and activity (i,j) must be completed.

$\text{Min } Z = X_6 - X_1$

S.T.	$X_3 >= X_1 + 6$	(Arc (1,3) constraint)
	$X_2 >= X_1 + 9$	(Arc (1,2) constraint)
	$X_5 >= X_3 + 8$	(Arc (3,5) constraint)
	$X_4 >= X_3 + 7$	(Arc (3,4) constraint)
	$X_5 >= X_4 + 10$	(Arc (4,5) constraint)
	$X_6 >= X_5 + 12$	(Arc (5,6) constraint)
	$X_3 >= X_2$	(Arc (2,3) constraint)

The optimal solution to this LP is $Z=38$, $X_1=0$, $X_2=9$, $X_3=9$, $X_4=16$, $X_5=26$, $X_6=38$

Crashing the Project

- Widgetco wants to introduce product 3 within 25 days. Because the critical path in Example 6 has a length of 38 days, Widgetco will have to expend additional resources to meet the 25-day project deadline.
- Suppose that by allocating additional resources to an activity, Widgetco can reduce the duration of any activity by as many as 5 days. The cost per day of reducing the duration of an activity is shown below.

A	B	C	D	E	F
$10	$20	$3	$30	$40	$50

- To find the minimum cost of completing the project by the 25-day deadline, define variables A, B, C, D, E and F as follows:
- A…F = number of days by which duration of activity A…F is reduced
- x_j = time that the event corresponding to node j occurs

- Then Widgetco should solve the following LP:
- Min z = 10A + 20B+3C+30D+40E+50F S.t.
  A <= 5, B <=5, C <= 5, D <=5, E<=5, F<=5
  x2 >= x1 + 9 - B (Arc(1,2) constraint)
  x3 >= x1 + 6 - A (Arc(1,3) constraint)
  x5 >= x3 + 8 – C (Arc(3,5) constraint)
  x4 >= x3 + 7 – D (Arc(3,4) constraint)
  x5 >= x4 +10 – E (Arc(4,5) constraint)
  x6 >= x5 +12 – F (Arc(5,6) constraint)
  x3 >= x2 + 0 (Arc(2,3) constraint)
  x6 - x1 <= 25
  A, B, C, D, E, F >= 0

PERT

CPM assumes that the duration of each activity is known with certainty. For many projects, this is clearly not applicable. PERT is an attempt to correct this shortcoming of CPM by modeling the duration of each activity as a random variable. For each activity, PERT requires that the project manager estimate the following three quantities:

a : estimate of the activity's duration under the most favorable conditions

b : estimate of the activity's duration under the least favorable conditions

m : most likely value for the activity's duration

Let T_{ij} be the duration of activity (i,j). PERT requires the assumption that T_{ij} follows a beta distribution. It is important to realize that it can approximate a wide range of random variables, including many positively skewed, negatively skewed, and symmetric random variables. According to this assumption, it can be shown that the mean and variance of T_{ij} may be approximated by

$$E(T_{ij}) = \frac{a + 4m + b}{6}$$

$$\operatorname{var} T_{ij} = \frac{(b - a)^2}{36}$$

PERT requires the assumption that the durations of all activities are independent. Thus,

$$\sum_{(i,j)\in path} E(T_{ij})$$: expected duration of activities on any path

$$\sum_{(i,j)\in path} \operatorname{var} T_{ij}$$: variance of duration of activities on any path

Let CP be the random variable denoting the total duration of the activities on a critical path found by CPM. PERT assumes that the critical path found by CPM contains enough activities to allow us to invoke the Central Limit Theorem and conclude that the following is normally distributed:

$$CP = \sum_{(i,j) \in critical path} T_{ij}$$

A, B AND M FOR ACTIVITIES IN WIDGETCO

Activity	a	b	m
(1,2)	5	13	9
(1,3)	2	10	6
(3,5)	3	13	8
(3,4)	1	13	7
(4,5)	8	12	10
(5,6)	9	15	12

According to the table on the previous slide:

$$E(T_{12}) = \frac{5+13+36}{6} = 9, \operatorname{var} T_{12} = \frac{(13-5)^2}{36} = 1.78$$

$$E(T_{13}) = \frac{2+10+24}{6} = 6, \operatorname{var} T_{13} = \frac{(10-2)^2}{36} = 1.78$$

$$E(T_{35}) = \frac{3+13+32}{6} = 8, \operatorname{var} T_{35} = \frac{(13-3)^2}{36} = 2.78$$

$$E(T_{34}) = \frac{1+13+28}{6} = 7, \operatorname{var} T_{34} = \frac{(13-1)^2}{36} = 4$$

$$E(T_{45}) = \frac{8+12+40}{6} = 10, \operatorname{var} T_{45} = \frac{(12-8)^2}{36} = 0.44$$

$$E(T_{56}) = \frac{9+15+48}{6} = 12, \operatorname{var} T_{56} = \frac{(15-9)^2}{36} = 1$$

Of course, the fact that arc (2,3) is a dummy arc yields

$$E(T_{23})=varT_{23}=0$$

The critical path was 1-2-3-4-5-6. Thus,

$E(CP)=9+0+7+10+12=38$

$varCP=1.78+0+4+0.44+1=7.22$

Then the standard deviation for CP is $(7.22)^{1/2}=2.69$

And

$$P(CP \le 35) = P(\frac{CP-38}{2.69} \le \frac{35-38}{2.69}) = P(Z \le -1.12) = 0.13$$

PERT implies that there is a 13% chance that the project will be completed within 35 days.

MINIMUM COST NETWORK FLOW PROBLEMS

The transportation, assignment, transshipment, shortest path, maximum flow, and CPM problems are all special cases of minimum cost network flow problems (MCNFP). Any MCNFP can be solved by a generalization of the transportation simplex called the network simplex.

To define MCNFP, let

X_{ij}= number of units of flow sent from node i to node j through arc(i,j)

b_i= net supply (outflow-inflow) at node i

c_{ij}= cost of transporting 1 unit of flow from node i to node j via arc(i,j)

L_{ij}= lower bound of flow through arc(i,j) (if there is no lower bound, let L_{ij}=0)

U_{ij}= upper bound of flow through arc(i,j) (if there is no upper bound, let U_{ij}=infinity)

Then the MCNFP can be modeled as follows:

Flow balance equations

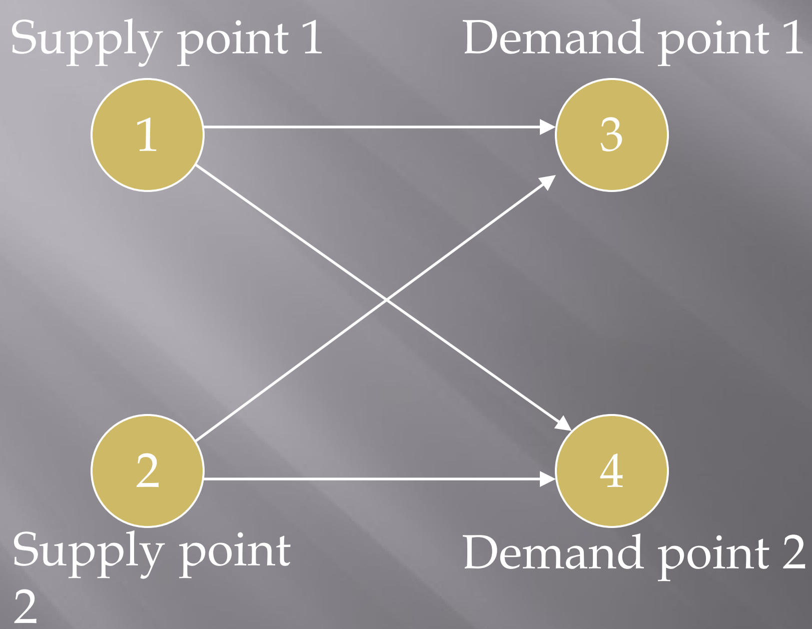
$$\min \sum_{all\,arcs} c_{ij} X_{ij}$$

$$s.t. \sum_{j} X_{ij} - \sum_{k} X_{ki} = b_i$$ (for each node i in the network)

$$L_{ij} \le X_{ij} \le U_{ij}$$ (for each arc in the network)

FORMULATING A TRANSPORTATION PROBLEM AS AN MCNFP

Consider the transportation problem below:



	1	2	4 (Node 1)
	3	4	5 (Node 2)
	6 (Node 3)	3 (Node 4)	

MCNFP Representation of the problem:

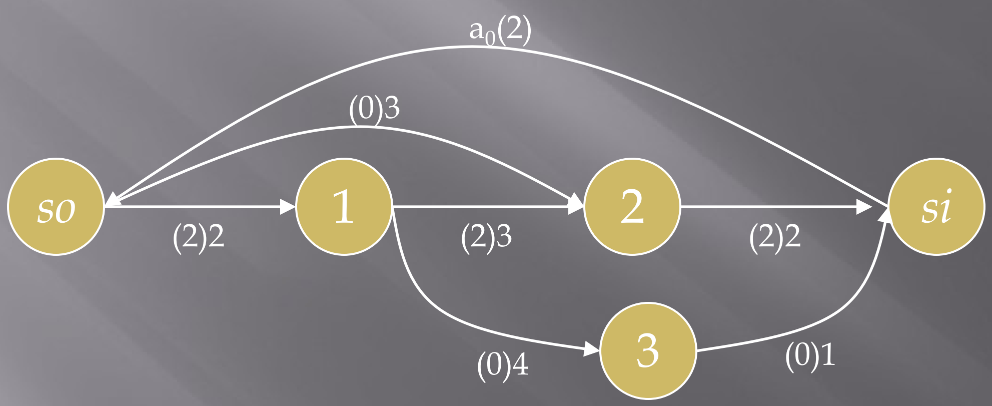
$$\min Z = X_{13} + 2X_{14} + 3X_{23} + 4X_{24}$$

X_{13}	X_{14}	X_{23}	X_{24}		rhs	Constraint
1	1	0	0	=	4	Node 1
0	0	1	1	=	5	Node 2
-1	0	-1	0	=	-6	Node 3
1	-1	0	-1	=	-3	Node 4

All Variables nonnegative

If the problem had not been balanced, we could not have formulated the problem as an MCNFP.

Formulating a Maximum-Flow Problem as an MCNFP

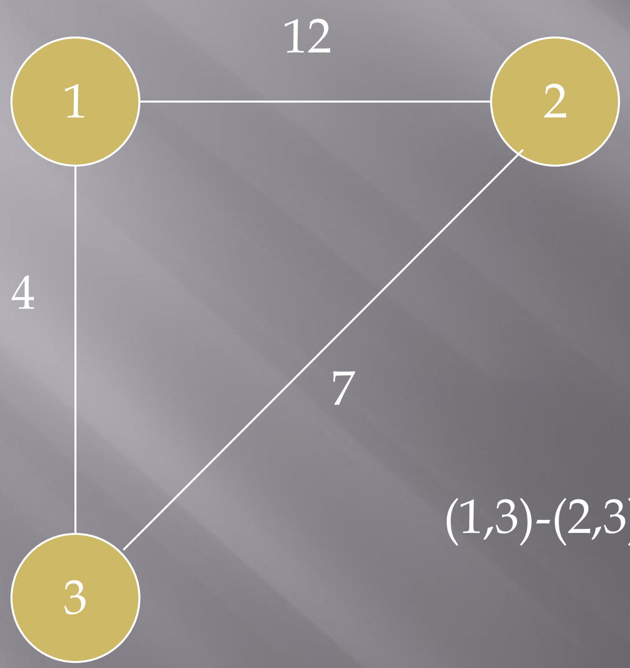


min z=x0									
Xso,1	Xso,2	X12	X12	X3,si	X2,si	X0		rhs	Constraint
1	1	0	0	0	0	-1	=	0	Node s0
-1	0	1	1	0	0	0	=	0	Node 1
0	-1	0	-1	0	1	0	=	0	Node 2
0	0	-1	0	1	0	0	=	0	Node 3
0	0	0	0	-1	-1	1	=	0	Node si
1	0	0	0	0	0	0	≦	2	Arc (s0,1)
0	1	0	0	0	0	0	≦	3	Arc (s0,2)
0	0	1	0	0	0	0	≦	4	Arc (1,3)
0	0	0	1	0	0	0	≦	3	Arc (1,2)
0	0	0	0	1	0	0	≦	1	Arc (3,si)
0	0	0	0	0	1	0	≦	2	Arc (2,si)
All variables non-negative									

5.6 MINIMUM SPANNING TREE PROBLEMS

Suppose that each arc (i,j) in a network has a length associated with it and that arc (i,j) represents a way of connecting node i to node j. For example, if each node in a network represents a computer in a computer network, arc(i,j) might represent an underground cable that connects computer i to computer j. In many applications, we want to determine the set of arcs in a network that connect all nodes such that the sum of the length of the arcs is minimized. Clearly, such a group of arcs contain no loop.

For a network with n nodes, a spanning tree is a group of n-1 arcs that connects all nodes of the network and contains no loops.



(1,2)-(2,3)-(3,1) is a loop

(1,3)-(2,3) is the minimum spanning tree

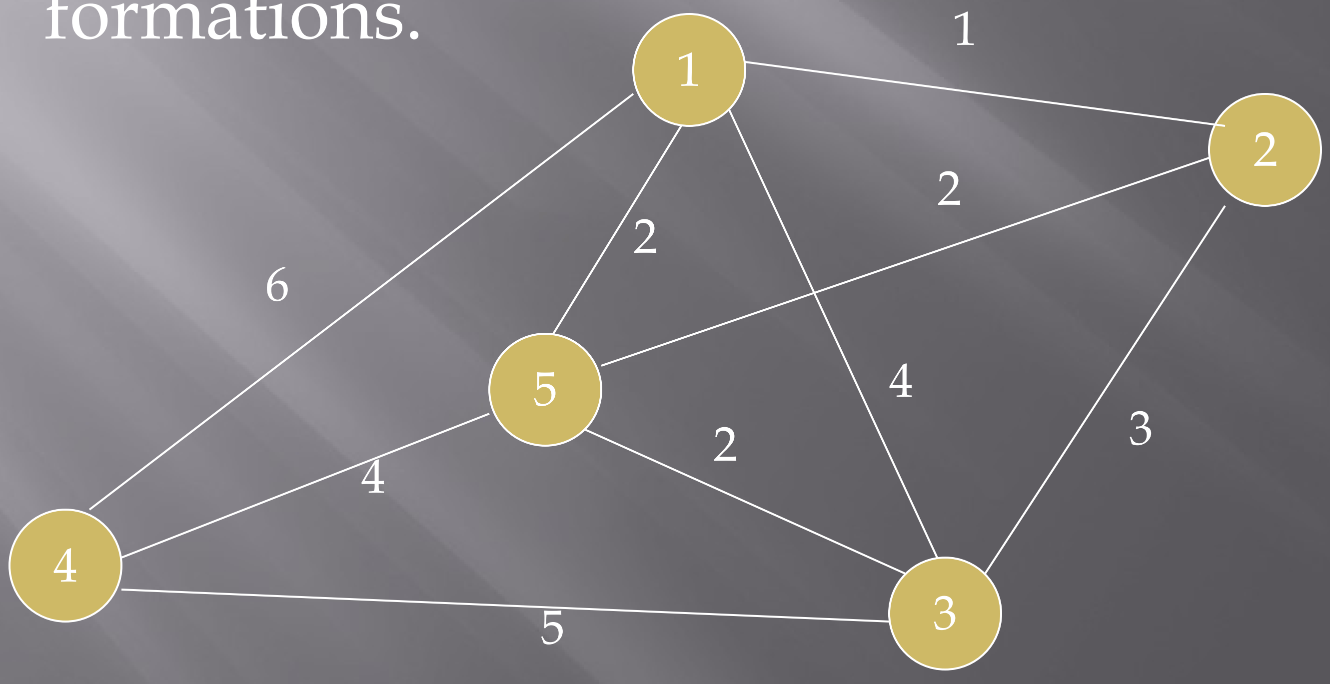
The following method (MST Algorithm) may be used to find a minimum spanning tree:

- Begin at any node i, and join node i to the node in the network (call it node j) that is closest to node i. The two nodes i and j now form a connected set of nodes C={i,j}, and arc (i,j) will be in the minimum spanning tree. The remaining nodes in the network (call them Ć) are referred to as the unconnected set of nodes.

- Now choose a member of Ć (call it n) that is closest to some node in C. Let m represent the node in C that is closest to n. Then the arc(m,n) will be in the minimum spanning tree. Now update C and Ć. Since n is now connected to {i,j}, C now equals {i,j,n} and we must eliminate node n from Ć.

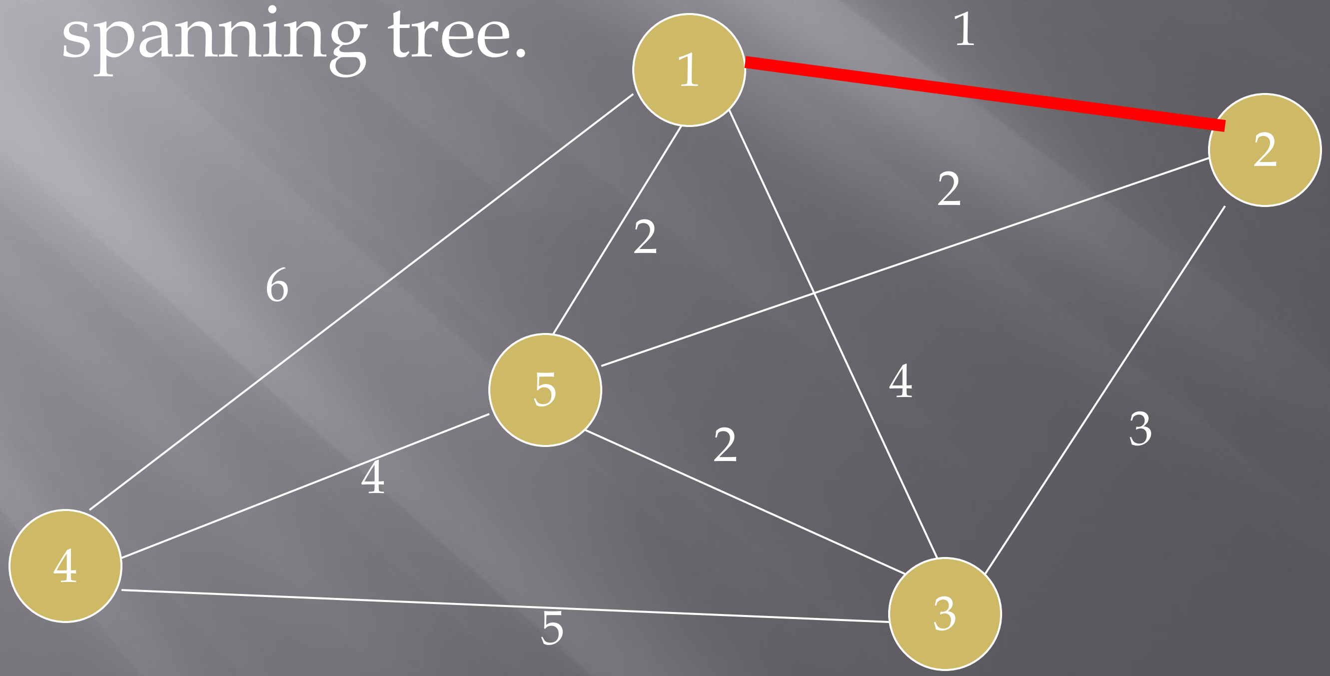
- Repeat this process until a minimum spanning tree is found. Ties for closest node and arc to be included in the minimum spanning tree may be broken arbitrarily.

Example: The State University campus has five computers. The distances between computers are given in the figure below. What is the minimum length of cable required to interconnect the computers? Note that if two computers are not connected this is because of underground rock formations.

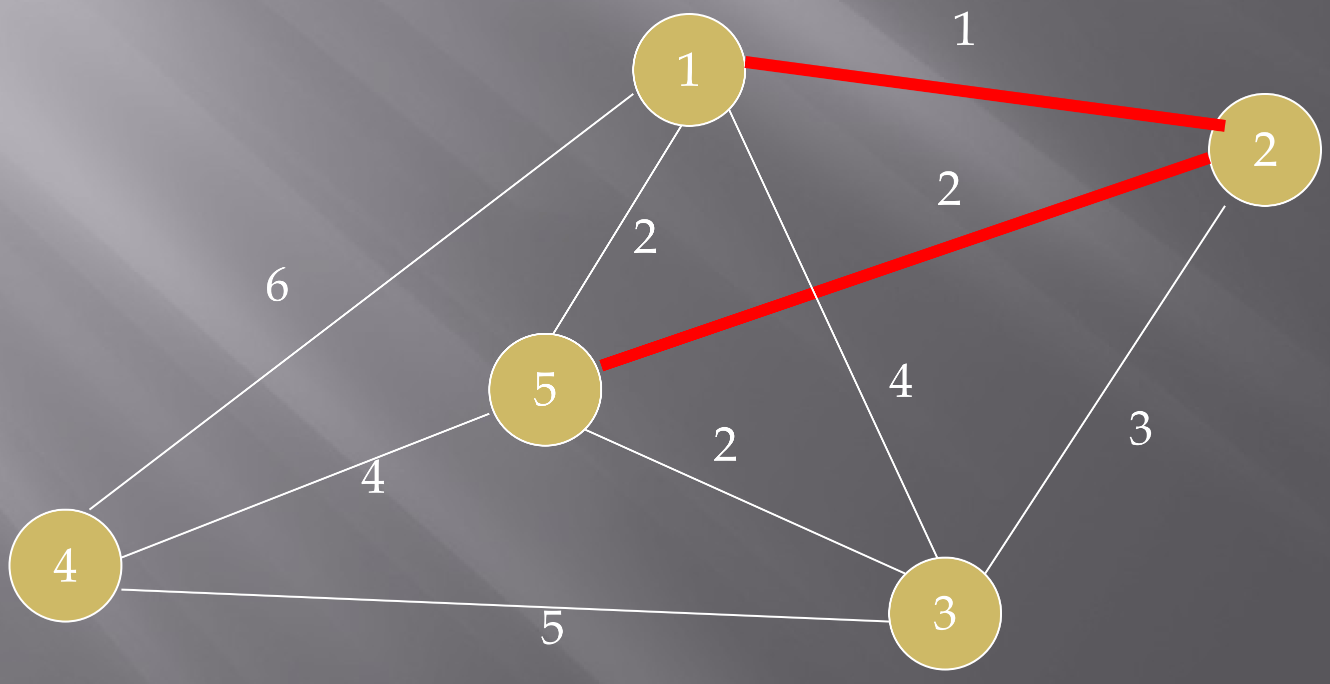


Solution: We want to find the minimum spanning tree.

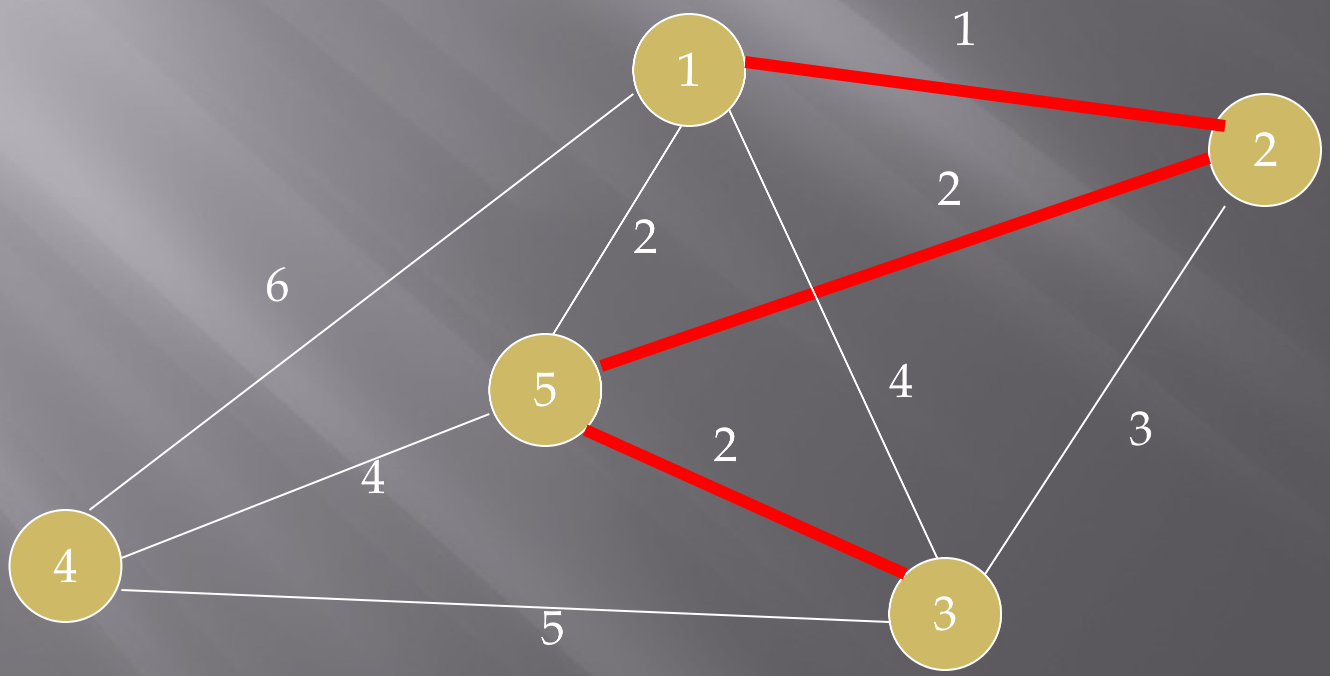
- Iteration 1: Following the MST algorithm discussed before, we arbitrarily choose node 1 to begin. The closest node is node 2. Now C={1,2}, Ć={3,4,5}, and arc(1,2) will be in the minimum spanning tree.



- Iteration 2: Node 5 is closest to C. since node 5 is two blocks from node 1 and node 2, we may include either arc(2,5) or arc(1,5) in the minimum spanning tree. We arbitrarily choose to include arc(2,5). Then C={1,2,5} and Ć={3,4}.

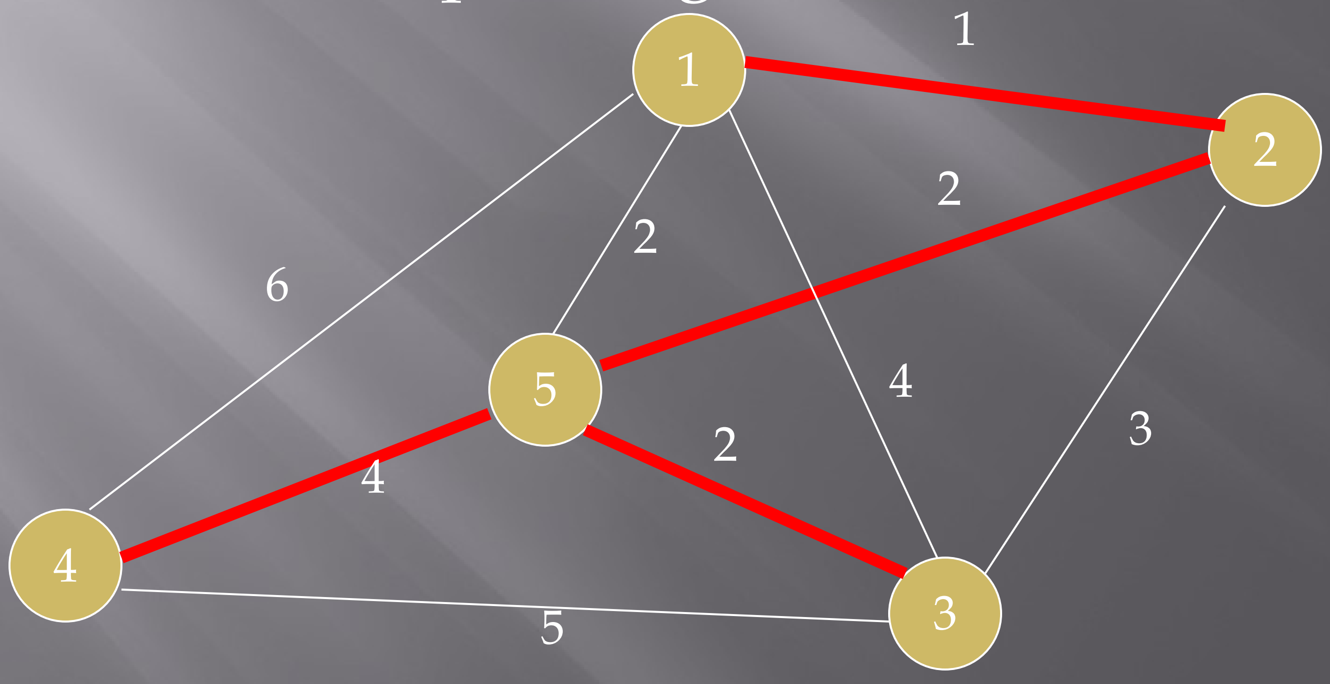


- Iteration 3: Since node 3 is two blocks from node 5, we may include arc(5,3) in the minimum spanning tree. Now C={1,2,5,3} and Ć={4}.



- Iteration 4: Node 5 is the closest node to node 4. Thus, we add arc(5,4) to the minimum spanning tree.

We now have a minimum spanning tree consisting of arcs(1,2), (2,5), (5,3), and (5,4). The length of the minimum spanning tree is 1+2+2+4=9 blocks.



CHAPTER 6

Optimization concept

Introduction to O.C.

- **Optimization** is the mathematical discipline which is concerned with finding the maxima and minima of functions, possibly subject to constraints

Where would we use optimization?

- Architecture
- Nutrition
- Electrical circuits
- Economics
- Transportation
- etc.

What do we optimize?

- A real function of n variables

$$f(x_1, x_2, \ldots, x_n)$$

- with or without constrains

Unconstrained optimization

$$\min \ f(x,y) = x^2 + 2y^2$$

Optimization with constraints

$$\min \ f(x,y) = x^2 + 2y^2$$

$$x > 0$$

or

$$\min \ f(x,y) = x^2 + 2y^2$$

$$-2 < x < 5, y \geq 1$$

or

$$\min \ f(x,y) = x^2 + 2y^2$$

$$x + y = 2$$

Writing the motivation and goals of an Optimization Problem

- Objective function
- Motivation
- Goals

Analytical Approach In Optimization

- Differential method of calculus used
- As shown above.
- Limited scope in practical application.

Subjective Optimization

- Instead of defining mathematical cost function, human behavior directly evaluates.
- Potential solutions are evaluated.
- Good or bad ones are selected
- Evaluation algorithms and other population based O.T. are used

Combinatorial optimization

- Combinatorial optimization is a subset of mathematical optimization that is related to operations research, algorithm theory, and computational complexity theory.
- It has important applications in several fields, including artificial intelligence, machine learning, mathematics, auction theory, and software engineering.
- Some research literature considers discrete optimization to consist of integer programming together with combinatorial optimization (which in turn is composed of optimization problem dealing with graph structures) although all of these topics have closely intertwined research literature. It often involves determining the way to efficiently allocate resources used to find solutions to mathematical problems.

Applications of C.O.

- Applications for combinatorial optimization include, but are not limited to:
- Developing the best airline network of spokes and destinations
- Deciding which taxis in a fleet to route to pick up fares
- Determining the optimal way to deliver packages
- Determining the right attributes of concept elements prior to concept testing

CHAPTER 6

Optimization concept

Introduction to O.C.

- **Optimization** is the mathematical discipline which is concerned with finding the maxima and minima of functions, possibly subject to constraints

Where would we use optimization?

- Architecture
- Nutrition
- Electrical circuits
- Economics
- Transportation
- etc.

What do we optimize?

- A real function of n variables

$$f(x_1, x_2, \ldots, x_n)$$

- with or without constrains

Unconstrained optimization

$$\min\ f(x,y) = x^2 + 2y^2$$

Optimization with constraints

$$\min \; f(x,y) = x^2 + 2y^2$$
$$x > 0$$

or

$$\min \; f(x,y) = x^2 + 2y^2$$
$$-2 < x < 5, y \geq 1$$

or

$$\min \; f(x,y) = x^2 + 2y^2$$
$$x + y = 2$$

Writing the motivation and goals of an Optimization Problem

- Objective function
- Motivation
- Goals

Analytical Approach In Optimization

- Differential method of calculus used
- As shown above.
- Limited scope in practical application.

Subjective Optimization

- Instead of defining mathematical cost function, human behavior directly evaluates.
- Potential solutions are evaluated.
- Good or bad ones are selected
- Evaluation algorithms and other population based O.T. are used

Combinatorial optimization

- Combinatorial optimization is a subset of mathematical optimization that is related to operations research, algorithm theory, and computational complexity theory.
- It has important applications in several fields, including artificial intelligence, machine learning, mathematics, auction theory, and software engineering.
- Some research literature considers discrete optimization to consist of integer programming together with combinatorial optimization (which in turn is composed of optimization problem dealing with graph structures) although all of these topics have closely intertwined research literature. It often involves determining the way to efficiently allocate resources used to find solutions to mathematical problems.

Applications of C.O.

- Applications for combinatorial optimization include, but are not limited to:
- Developing the best airline network of spokes and destinations
- Deciding which taxis in a fleet to route to pick up fares
- Determining the optimal way to deliver packages
- Determining the right attributes of concept elements prior to concept testing