Operation Research

Module 4

4.1 Introduction to CPM / PERT Techniques

CPM (Critical Path Method) was developed by Walker to solve project scheduling problems. **PERT (Project Evaluation and Review Technique)** was developed by team of engineers working on the polar's missile programme of US navy.

The methods are essentially **network-oriented techniques** using the same principle. PERT and CPM are basically time-oriented methods in the sense that they both lead to determination of a time schedule for the project. The significant difference between two approaches is that the time estimates for the different activities in CPM were assumed to be **deterministic** while in PERT these are described **probabilistically**. These techniques are referred as **project scheduling** techniques.

4.2 Applications of CPM / PERT

These methods have been applied to a wide variety of problems in industries and have found acceptance even in government organizations. These include

- Construction of a dam or a canal system in a region
- Construction of a building or highway
- Maintenance or overhaul of airplanes or oil refinery
- Space flight
- Cost control of a project using PERT / COST
- Designing a prototype of a machine
- Development of supersonic planes

4.3 Basic Steps in PERT / CPM

Project scheduling by PERT / CPM consists of four main steps

1. **Planning**

- The planning phase is started by splitting the total project in to small projects. These smaller projects in turn are divided into activities and are analyzed by the department or section.
- The relationship of each activity with respect to other activities are defined and established and the corresponding responsibilities and the authority are also stated.

- Thus the possibility of overlooking any task necessary for the completion of the project is reduced substantially.

2. Scheduling

- The ultimate objective of the scheduling phase is to prepare a time chart showing the start and finish times for each activity as well as its relationship to other activities of the project.
- Moreover the schedule must pinpoint the critical path activities which require special attention if the project is to be completed in time.
- For non-critical activities, the schedule must show the amount of slack or float times which can be used advantageously when such activities are delayed or when limited resources are to be utilized effectively.

3. Allocation of resources

- Allocation of resources is performed to achieve the desired objective. A resource is a physical variable such as labour, finance, equipment and space which will impose a limitation on time for the project.
- When resources are limited and conflicting, demands are made for the same type of resources a systematic method for allocation of resources become essential.
- Resource allocation usually incurs a compromise and the choice of this compromise depends on the judgment of managers.

4. Controlling

- The final phase in project management is controlling. Critical path methods facilitate the application of the principle of management by expectation to identify areas that are critical to the completion of the project.
- By having progress reports from time to time and updating the network continuously, a better financial as well as technical control over the project is exercised.
- Arrow diagrams and time charts are used for making periodic progress reports. If required, a new course of action is determined for the remaining portion of the project.

4.4 Network Diagram Representation

In a network representation of a project certain definitions are used

1. Activity

Any individual operation which utilizes resources and has an end and a beginning is called activity. An arrow is commonly used to represent an activity with its head indicating the direction of progress in the project. These are classified into four categories

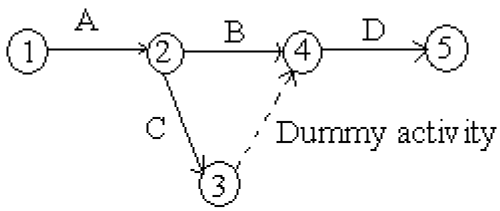
1. **Predecessor activity** – Activities that must be completed immediately prior to the start of another activity are called predecessor activities.
2. **Successor activity** – Activities that cannot be started until one or more of other activities are completed but immediately succeed them are called successor activities.
3. **Concurrent activity** – Activities which can be accomplished concurrently are known as concurrent activities. It may be noted that an activity can be a predecessor or a successor to an event or it may be concurrent with one or more of other activities.

4. **Dummy activity** – An activity which does not consume any kind of resource but merely depicts the technological dependence is called a dummy activity.

The dummy activity is inserted in the network to clarify the activity pattern in the following two situations

- To make activities with common starting and finishing points distinguishable
- To identify and maintain the proper precedence relationship between activities that is not connected by events.

For example, consider a situation where A and B are concurrent activities. C is dependent on A and D is dependent on A and B both. Such a situation can be handled by using a dummy activity as shown in the figure.

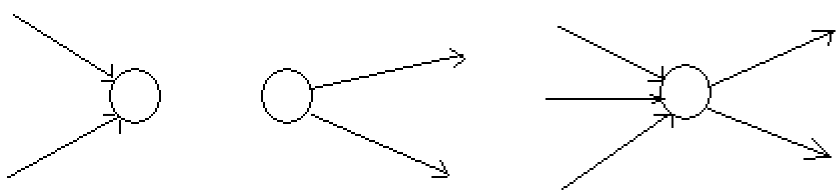


2. Event

An event represents a point in time signifying the completion of some activities and the beginning of new ones. This is usually represented by a circle in a network which is also called a node or connector.

The events are classified in to three categories

1. **Merge event** – When more than one activity comes and joins an event such an event is known as merge event.
2. **Burst event** – When more than one activity leaves an event such an event is known as burst event.
3. **Merge and Burst event** – An activity may be merge and burst event at the same time as with respect to some activities it can be a merge event and with respect to some other activities it may be a burst event.



Merge event Burst event Merge and Burst event

3. Sequencing

The first prerequisite in the development of network is to maintain the precedence relationships. In order to make a network, the following points should be taken into considerations

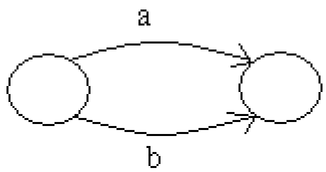
- What job or jobs precede it?
- What job or jobs could run concurrently?
- What job or jobs follow it?
- What controls the start and finish of a job?

Since all further calculations are based on the network, it is necessary that a network be drawn with full care.

4.5 Rules for Drawing Network Diagram

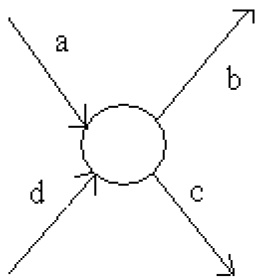
Rule 1

Each activity is represented by one and only one arrow in the network



Rule 2

No two activities can be identified by the same end events



Rule 3

In order to ensure the correct precedence relationship in the arrow diagram, following questions must be checked whenever any activity is added to the network

- What activity must be completed immediately before this activity can start?
- What activities must follow this activity?
- What activities must occur simultaneously with this activity?

In case of large network, it is essential that certain good habits be practiced to draw an easy to follow network

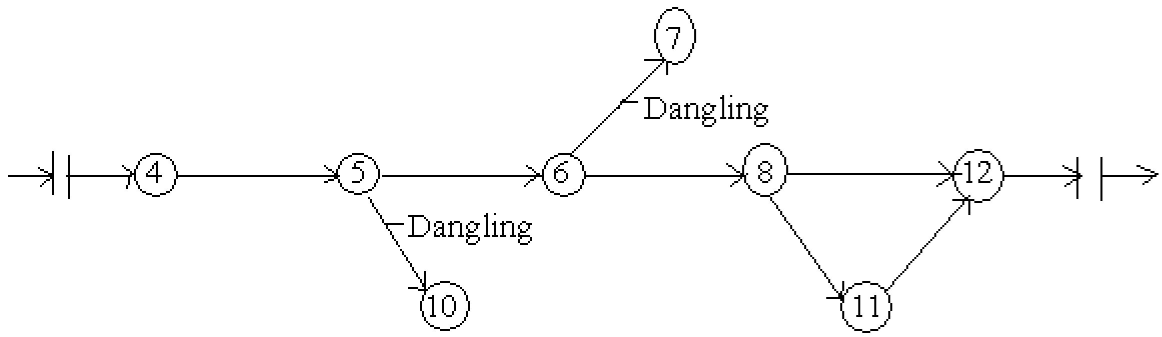
- Try to avoid arrows which cross each other
- Use straight arrows
- Do not attempt to represent duration of activity by its arrow length
- Use arrows from left to right. Avoid mixing two directions, vertical and standing arrows may be used if necessary.
- Use dummies freely in rough draft but final network should not have any redundant dummies.
- The network has only one entry point called start event and one point of emergence called the end event.

4.6 Common Errors in Drawing Networks

The three types of errors are most commonly observed in drawing network diagrams

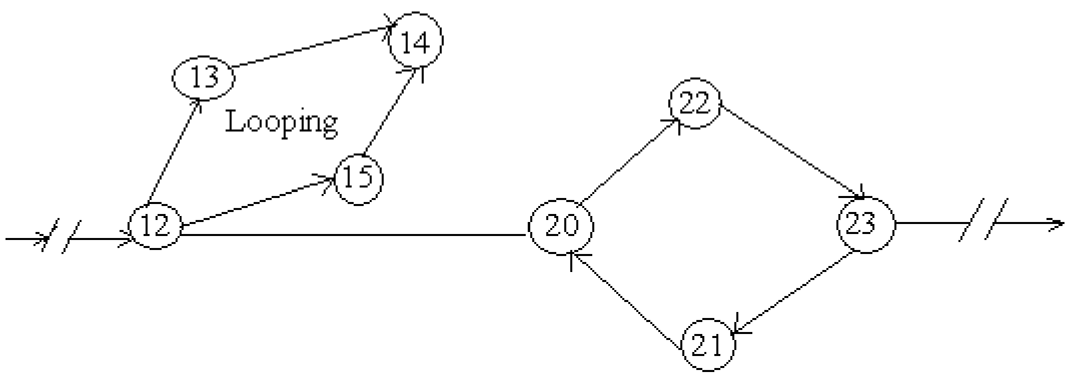
1. Dangling

To disconnect an activity before the completion of all activities in a network diagram is known as dangling. As shown in the figure activities (5 – 10) and (6 – 7) are not the last activities in the network. So the diagram is wrong and indicates the error of dangling



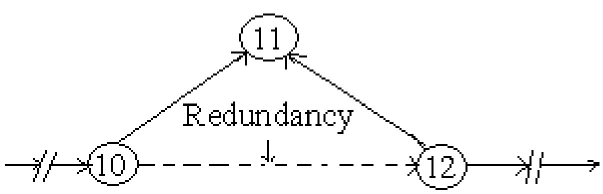
2. Looping or Cycling

Looping error is also known as cycling error in a network diagram. Drawing an endless loop in a network is known as error of looping as shown in the following figure.



3. Redundancy

Unnecessarily inserting the dummy activity in network logic is known as the error of redundancy as shown in the following diagram



4.7 Critical Path in Network Analysis

3.1.1 Basic Scheduling Computations

The notations used are
(i, j) = Activity with tail event i and head event j
E_i = Earliest occurrence time of event i
L_j = Latest allowable occurrence time of event j
D_{ij} = Estimated completion time of activity (i, j)
$(Es)_{ij}$ = Earliest starting time of activity (i, j)
$(Ef)_{ij}$ = Earliest finishing time of activity (i, j)
$(Ls)_{ij}$ = Latest starting time of activity (i, j)
$(Lf)_{ij}$ = Latest finishing time of activity (i, j)

The procedure is as follows

1. **Determination of Earliest time (E_j): Forward Pass computation**

- **Step 1**
  The computation begins from the start node and move towards the end node. For easiness, the forward pass computation starts by assuming the earliest occurrence time of zero for the initial project event.

- **Step 2**
  i. Earliest starting time of activity (i, j) is the earliest event time of the tail end event i.e. $(Es)_{ij} = E_i$
  ii. Earliest finish time of activity (i, j) is the earliest starting time + the activity time i.e. $(Ef)_{ij} = (Es)_{ij} + D_{ij}$ or $(Ef)_{ij} = E_i + D_{ij}$
  iii. Earliest event time for event j is the maximum of the earliest finish times of all activities ending in to that event i.e. $E_j = \max [(Ef)_{ij}$ for all immediate predecessor of (i, j)] or $E_j = \max [E_i + D_{ij}]$

2. **Backward Pass computation (for latest allowable time)**

- **Step 1**
  For ending event assume E = L. Remember that all E's have been computed by forward pass computations.

- **Step 2**
  Latest finish time for activity (i, j) is equal to the latest event time of event j i.e. $(Lf)_{ij} = L_j$

- **Step 3**
  Latest starting time of activity (i, j) = the latest completion time of (i, j) – the activity time or $(Ls)_{ij} = (Lf)_{ij} - D_{ij}$ or $(Ls)_{ij} = L_j - D_{ij}$

- **Step 4**
  Latest event time for event 'i' is the minimum of the latest start time of all activities originating from that event i.e. $L_i = \min [(Ls)_{ij}$ for all immediate successor of (i, j)] $= \min [(Lf)_{ij} - D_{ij}] = \min [L_j - D_{ij}]$

3. Determination of floats and slack times

There are three kinds of floats

- **Total float** – The amount of time by which the completion of an activity could be delayed beyond the earliest expected completion time without affecting the overall project duration time.
  Mathematically
  $(Tf)_{ij}$ = (Latest start – Earliest start) for activity (i – j)
  $(Tf)_{ij} = (Ls)_{ij} - (Es)_{ij}$ or $(Tf)_{ij} = (L_j - D_{ij}) - E_i$

- **Free float –** The time by which the completion of an activity can be delayed beyond the earliest finish time without affecting the earliest start of a subsequent activity.
  Mathematically
  $(Ff)_{ij}$ = (Earliest time for event j – Earliest time for event i) – Activity time for (i, j)
  $(Ff)_{ij} = (E_j - E_i) - D_{ij}$

- **Independent float** – The amount of time by which the start of an activity can be delayed without effecting the earliest start time of any immediately following activities, assuming that the preceding activity has finished at its latest finish time.
  Mathematically
  $(If)_{ij} = (E_j - L_i) - D_{ij}$
  The negative independent float is always taken as zero.

- **Event slack** - It is defined as the difference between the latest event and earliest event times.
  Mathematically
  Head event slack = $L_j – E_j$, Tail event slack = $L_i - E_i$

4. Determination of critical path

- **Critical event** – The events with zero slack times are called critical events. In other words the event i is said to be critical if $E_i = L_i$

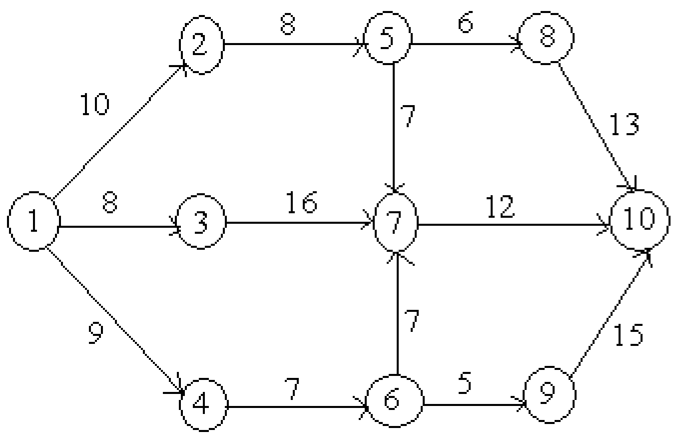
- **Critical activity** – The activities with zero total float are known as critical activities. In other words an activity is said to be critical if a delay in its start will cause a further delay in the completion date of the entire project.

- **Critical path** – The sequence of critical activities in a network is called critical path. The critical path is the longest path in the network from the starting event to ending event and defines the minimum time required to complete the project.

Worked Examples

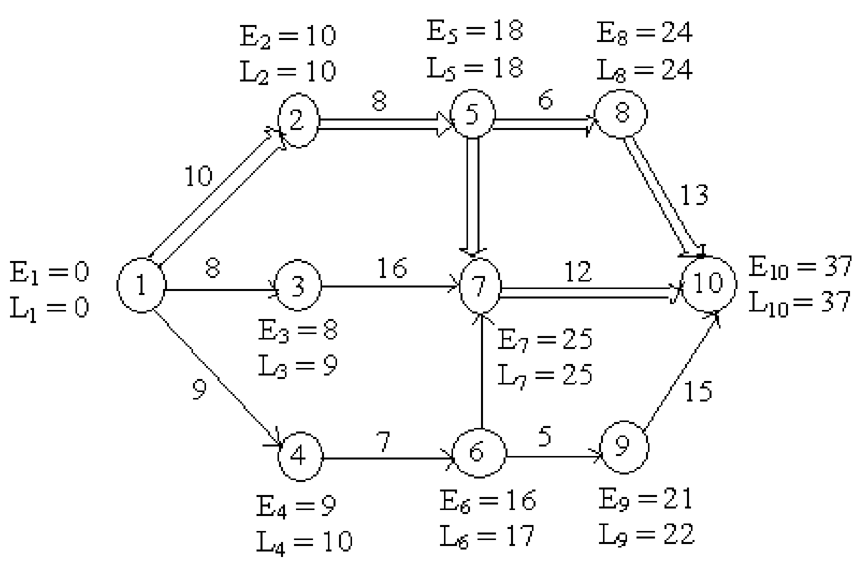
Example 1

Determine the early start and late start in respect of all node points and identify critical path for the following network.



Solution

Calculation of E and L for each node is shown in the network



Activity(i, j)	Normal Time (D_{ij})	Earliest Time		Latest Time		Float Time (L_i - D_{ij}) - E_i
		Start (E_i)	Finish (E_i + D_{ij})	Start (L_i - D_{ij})	Finish (L_i)	
(1, 2)	10	0	10	0	10	0
(1, 3)	8	0	8	1	9	1
(1, 4)	9	0	9	1	10	1
(2, 5)	8	10	18	10	18	0
(4, 6)	7	9	16	10	17	1
(3, 7)	16	8	24	9	25	1
(5, 7)	7	18	25	18	25	0
(6, 7)	7	16	23	18	25	2
(5, 8)	6	18	24	18	24	0
(6, 9)	5	16	21	17	22	1
(7, 10)	12	25	37	25	37	0
(8, 10)	13	24	37	24	37	0
(9, 10)	15	21	36	22	37	1

Network Analysis Table

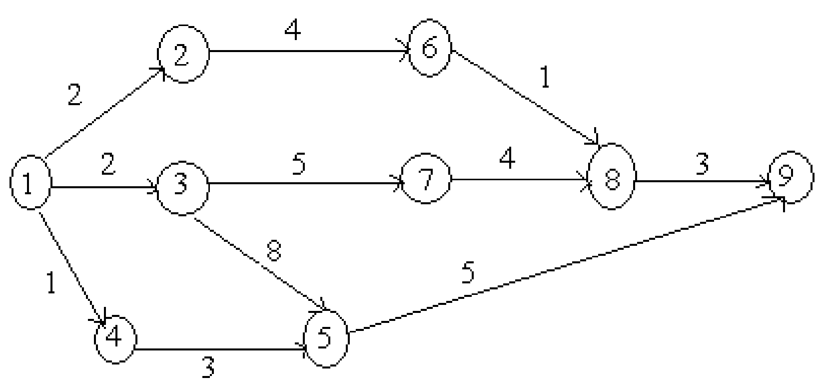
From the table, the critical nodes are (1, 2), (2, 5), (5, 7), (5, 8), (7, 10) and (8, 10)

From the table, there are two possible critical paths

i. 1 → 2 → 5 → 8 → 10
ii. 1 → 2 → 5 → 7 → 10

Example 2

Find the critical path and calculate the slack time for the following network

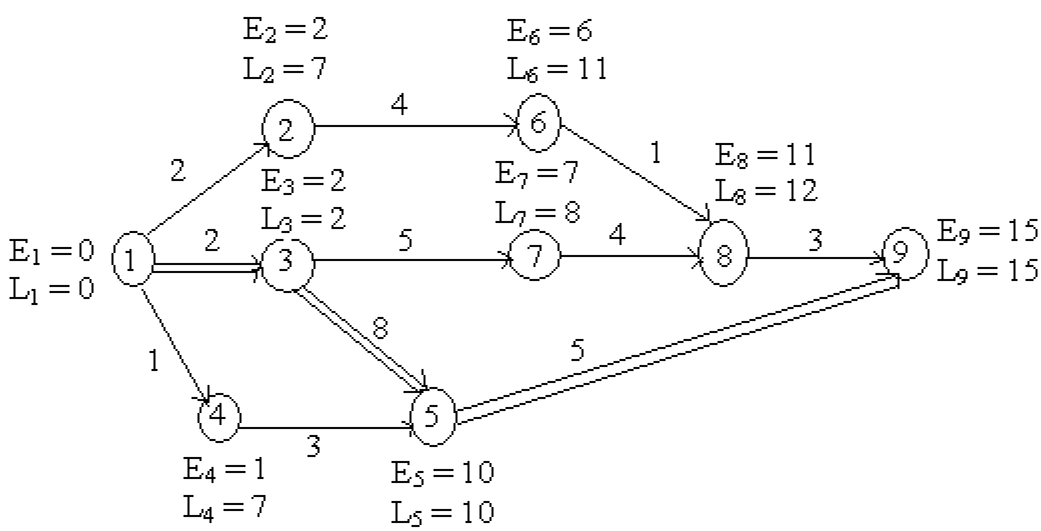


Solution

The earliest time and the latest time are obtained below

Activity(i, j)	Normal Time (D_{ij})	Earliest Time		Latest Time		Float Time (L_i - D_{ij}) - E_i
		Start (E_i)	Finish ($E_i + D_{ij}$)	Start ($L_i - D_{ij}$)	Finish (L_i)	
(1, 2)	2	0	2	5	7	5
(1, 3)	2	0	2	0	2	0
(1, 4)	1	0	1	6	7	6
(2, 6)	4	2	6	7	11	5
(3, 7)	5	2	7	3	8	1
(3, 5)	8	2	10	2	10	0
(4, 5)	3	1	4	7	10	6
(5, 9)	5	10	15	10	15	0
(6, 8)	1	6	7	11	12	5
(7, 8)	4	7	11	8	12	1
(8, 9)	3	11	14	12	15	1

From the above table, the critical nodes are the activities (1, 3), (3, 5) and (5, 9)



The critical path is 1 → 3 → 5 → 9

Example 3

A project has the following times schedule

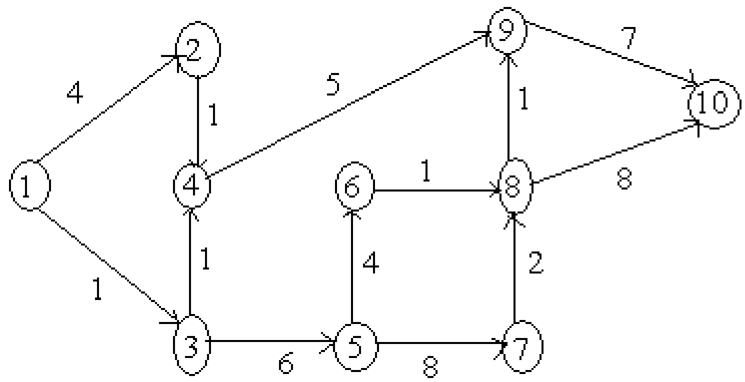
Activity	Times in weeks	Activity	Times in weeks
(1 – 2)	4	(5 – 7)	8
(1 – 3)	1	(6 – 8)	1
(2 – 4)	1	(7 – 8)	2
(3 – 4)	1	(8 – 9)	1
(3 – 5)	6	(8 – 10)	8
(4 – 9)	5	(9 – 10)	7
(5 – 6)	4		

Construct the network and compute

1. T_E and T_L for each event
2. Float for each activity
3. Critical path and its duration

Solution

The network is

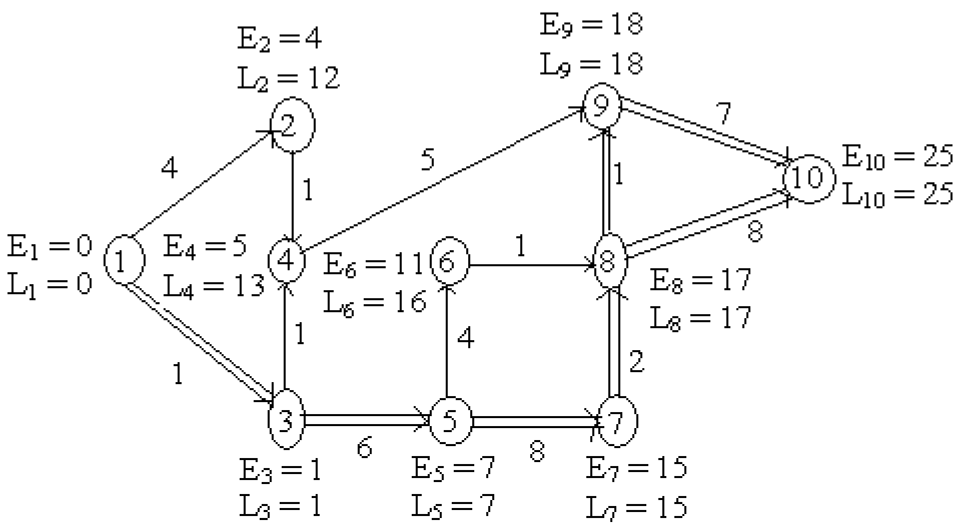


Event No.:	1	2	3	4	5	6	7	8	9	10
T_E:	0	4	1	5	7	11	15	17	18	25
T_L:	0	12	1	13	7	16	15	17	18	25

Float = T_L (Head event) – T_E (Tail event) – Duration

Activity	Duration	T_E (Tail event)	T_L (Head event)	Float
(1 – 2)	4	0	12	8
(1 – 3)	1	0	1	0
(2 – 4)	1	4	13	8
(3 – 4)	1	1	13	11
(3 – 5)	6	1	7	0
(4 – 9)	5	5	18	8
(5 – 6)	4	7	16	5
(5 – 7)	8	7	15	0
(6 – 8)	1	11	17	5
(7 – 8)	2	15	17	0
(8 – 9)	1	17	18	0
(8 – 10)	8	17	25	0
(9 – 10)	7	18	25	0

The resultant network shows the critical path



The two critical paths are

i. 1 → 3 → 5 →7 → 8 → 9 →10

ii. 1 → 3 → 5 → 7 → 8 →10

4.8 Project Evaluation and Review Technique (PERT)

The main objective in the analysis through PERT is to find out the completion for a particular event within specified date. The PERT approach takes into account the uncertainties. The three time values are associated with each activity

1. **Optimistic time** – It is the shortest possible time in which the activity can be finished. It assumes that every thing goes very well. This is denoted by t_0.
2. **Most likely time** – It is the estimate of the normal time the activity would take. This assumes normal delays. If a graph is plotted in the time of completion and the frequency of completion in that time period, then most likely time will represent the highest frequency of occurrence. This is denoted by t_m.
3. **Pessimistic time** – It represents the longest time the activity could take if everything goes wrong. As in optimistic estimate, this value may be such that only one in hundred or one in twenty will take time longer than this value. This is denoted by t_p.

In PERT calculation, all values are used to obtain the percent expected value.

1. **Expected time** – It is the average time an activity will take if it were to be repeated on large number of times and is based on the assumption that the activity time follows Beta distribution, this is given by

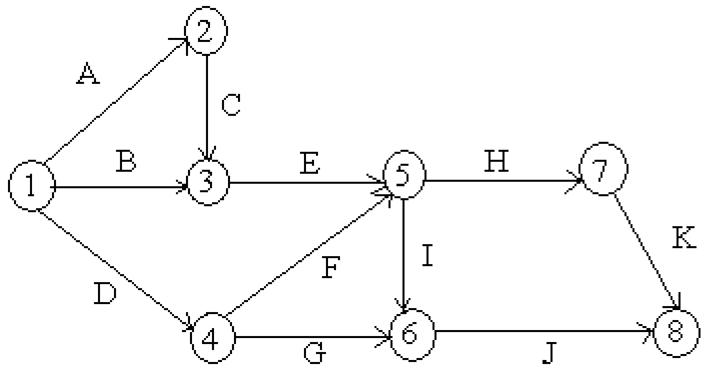
$$t_e = (t_0 + 4\, t_m + t_p) / 6$$

2. The **variance** for the activity is given by

$$\sigma^2 = [(t_p – t_o) / 6]^2$$

Worked Examples

Example 1
For the project



Task:	A	B	C	D	E	F	G	H	I	J	K
Least time:	4	5	8	2	4	6	8	5	3	5	6
Greatest time:	8	10	12	7	10	15	16	9	7	11	13
Most likely time:	5	7	11	3	7	9	12	6	5	8	9

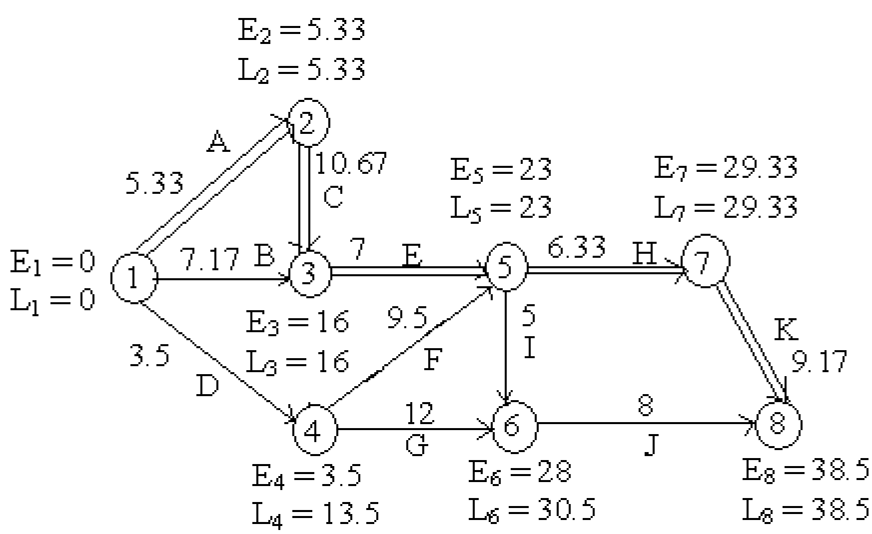
Find the earliest and latest expected time to each event and also critical path in the network.

Solution

Task	Least time(t_0)	Greatest time (t_p)	Most likely time (t_m)	Expected time (to + t_p + 4t_m)/6
A	4	8	5	5.33
B	5	10	7	7.17
C	8	12	11	10.67
D	2	7	3	3.5
E	4	10	7	7
F	6	15	9	9.5
G	8	16	12	12
H	5	9	6	6.33
I	3	7	5	5
J	5	11	8	8
K	6	13	9	9.17

Task	Expected time (t_e)	Start		Finish		Total float
		Earliest	Latest	Earliest	Latest	
A	5.33	0	0	5.33	5.33	0
B	7.17	0	8.83	7.17	16	8.83
C	10.67	5.33	5.33	16	16	0
D	3.5	0	10	3.5	13.5	10
E	7	16	16	23	23	0
F	9.5	3.5	13.5	13	23	10
G	12	3.5	18.5	15.5	30.5	15
H	6.33	23	23	29.33	29.33	0
I	5	23	25.5	28	30.5	2.5
J	8	28	30.5	36	38.5	2.5
K	9.17	29.33	29.33	31.5	38.5	0

The network is



The critical path is A →C →E → H → K

Example 2

A project has the following characteristics

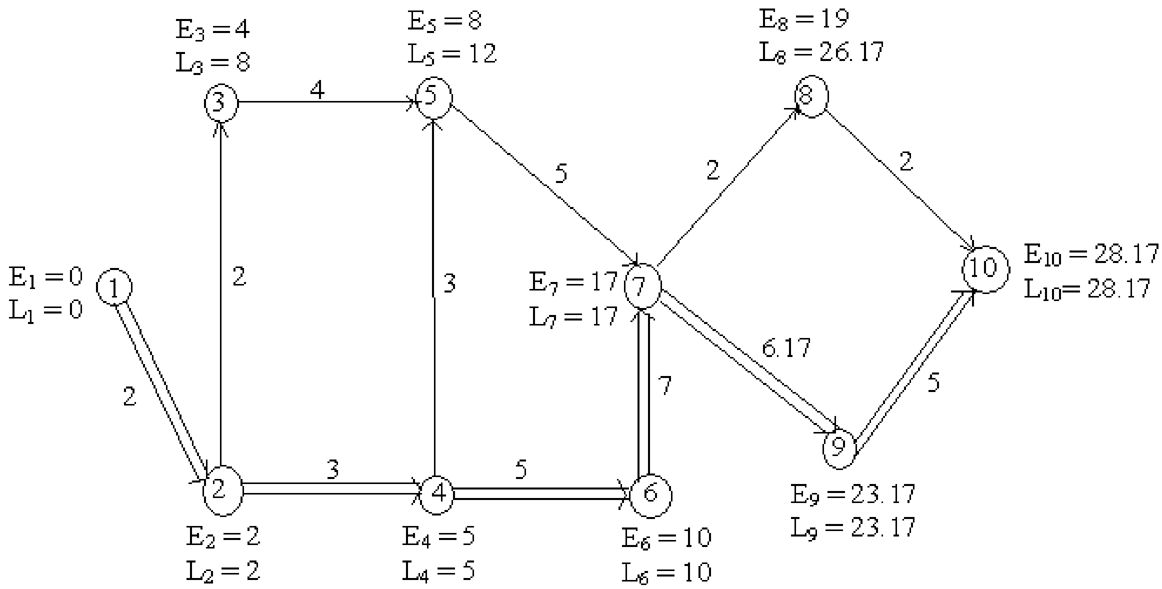
Activity	Most optimistic time (a)	Most pessimistic time (b)	Most likely time (m)
(1 – 2)	1	5	1.5
(2 – 3)	1	3	2
(2 – 4)	1	5	3
(3 – 5)	3	5	4
(4 – 5)	2	4	3
(4 – 6)	3	7	5
(5 – 7)	4	6	5
(6 – 7)	6	8	7
(7 – 8)	2	6	4
(7 – 9)	5	8	6
(8 – 10)	1	3	2
(9 – 10)	3	7	5

Construct a PERT network. Find the critical path and variance for each event.

Solution

Activity	(a)	(b)	(m)	(4m)	t_e $(a + b + 4m)/6$	v $[(b - a) / 6]^2$
(1 – 2)	1	5	1.5	6	2	4/9
(2 – 3)	1	3	2	8	2	1/9
(2 – 4)	1	5	3	12	3	4/9
(3 – 5)	3	5	4	16	4	1/9
(4 – 5)	2	4	3	12	3	1/9
(4 – 6)	3	7	5	20	5	4/9
(5 – 7)	4	6	5	20	5	1/9
(6 – 7)	6	8	7	28	7	1/9
(7 – 8)	2	6	4	16	4	4/9
(7 – 9)	5	8	6	24	6.17	1/4
(8 – 10)	1	3	2	8	2	1/9
(9 – 10)	3	7	5	20	5	4/9

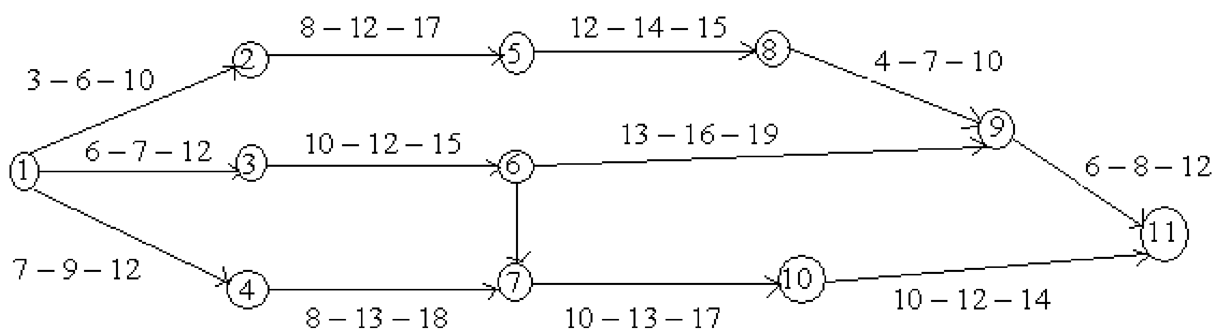
The network is constructed as shown below



The critical path = 1 → 2 → 4 → 6 → 7 →9 →10

Example 3

Calculate the variance and the expected time for each activity



Solution

Activity	(t_o)	(t_m)	(t_p)	t_e $(t_o + t_p + 4t_m)/6$	v $[(t_p - t_o) / 6]^2$
(1 – 2)	3	6	10	6.2	1.36
(1 – 3)	6	7	12	7.7	1.00
(1 – 4)	7	9	12	9.2	0.69
(2 – 3)	0	0	0	0.0	0.00
(2 – 5)	8	12	17	12.2	2.25
(3 – 6)	10	12	15	12.2	0.69
(4 – 7)	8	13	19	13.2	3.36
(5 – 8)	12	14	15	13.9	0.25
(6 – 7)	8	9	10	9.0	0.11
(6 – 9)	13	16	19	16.0	1.00
(8 – 9)	4	7	10	7.0	1.00
(7 – 10)	10	13	17	13.2	1.36
(9 – 11)	6	8	12	8.4	1.00
(10 – 11)	10	12	14	12.0	0.66