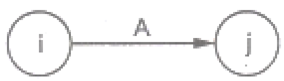
5.4 CRITICAL PATH METHOD

WORK ANALYSIS

Critical Path Method (CPM) was developed in the late 1950s as a method to resolve the issue of increased costs due to inefficient scheduling. Since then, CPM has become popular for planning projects and prioritizing tasks. It helps to break down complex projects into individual tasks and gain a better understanding of the project's flexibility. The Key Concept used by CPM is that a small set of activities, which make up the longest path through the activity network control the entire project. Such activity is called as critical activity. A critical path in project management is the longest sequence of activities that must be finished on time in order for the entire project to be complete. Any delays in critical tasks will delay the rest of the project. Non-critical activities can be re-planned, rescheduled and resources for them can be reallocated flexibly, without affecting the whole project. CPM revolves around discovering the most important tasks in the project timeline, identifying task dependencies, and calculating task durations. CPM has single time estimate which is assumed to be deterministic.

Basic Scheduling Computations in CPM



The basic notations used in CPM can be explained as follows:

For the given example,

(i,j) = Activity "A" with tail event "i" and head event "j"

E_i = Earliest occurrence time of event i

E_j = Latest al owable occurrence time of event j

D_{ij} = Estimated completion time of activity (i, j)

$(E_S)_U$ = Earliest starting time of activity (i, j)

$(E_F)_{ij}$ = Earliest finishing time of activity (i, j)

$(L_S)_{ij}$ = Latest starting time of activity (i, j)

$(L_f)_{ij}$ = Latest finishing time of activity (i, j)

(i) 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

Earliest starting time of activity (i, j) is the earliest event time of the tail end event

i.e. $(E_s)_{ij} = E_i$

Earliest finish time of activity (i, j) is the earliest starting time + the activity time

i.e. $(E_f)_{ij} = (E_s)_{ij} + D_{ij}$ **or** $(E_f)_{ij} = E_i + D_{ij}$

Earliest event time for event j is the maximum of the earliest finish times of all activities ending in to that event

i.e. $E_j = \max [(E_f)_{ij}$ **for all immediate predecessor of (i, j)] or**

$E_j = \max [E_i + D_{ij}]$

(ii) Backward Pass computation (for latest allowable time)

Step 1

For ending event assume E = L.

Also 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. $(L_f)_{ij} = L_j$

Step 3

Latest starting time of activity (i, j) = the latest completion time of (i, j) - the activity time bollo ons omisioak

i.e. $(L_s)_{ij} = (L_f)_{ij} - D_{ij}$ **or** $(L_s)_{ij} = L_j - D_{ij}$

Step 4

Latest event time for event " is the minimum of the latest start time of all activities originating from that event

i.e. $L_i = \min [(L_s)_{ij}$ **for all immediate successor of (i, j)]** $\min [(L_f)_{ij} - D_{ij}] = \min [L_j – D_{ij}]$

(iii) Determination of floats and slack times

There are three kinds of floats as follows:

❖ **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.

$(T_f)_{ij}$ Latest start - Earliest start) for activity (i –j)

i.e. = $(T_f)_{ij} = (L_s)_{ij} - (E_s)_{ij}$ **or** $(T_f)_{ij} = (T_L - T_E) - t_{ij}$

❖ **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.

$(F_f)_{ij}$ = Total float - Head event slack

i.e. $\mathbf{(F_f) = (E_j - E_i) - t_{ij}}$

❖ **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. The negative independent float is always taken as zero.

$(I_f)_{ij}$ = Free float - Tail event slack

i.e. $(I_f)_{ij} = (E_j - L_i) - t_{ij}$

❖ **Event slack -** It is defined as the difference between the latest event and earliest event times.

Head event slack = $L_j - E_j$

Tail event slack = $L_i - E_i$

(iv) 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.

5.4 -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.

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

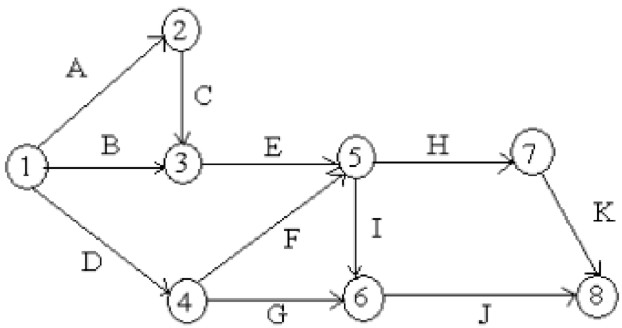
$$t_e = (t_0 + 4\, t_m + t_p) / 6$$

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

$$\sigma^2 = [(t_p - t_o) / 6]^2$$

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

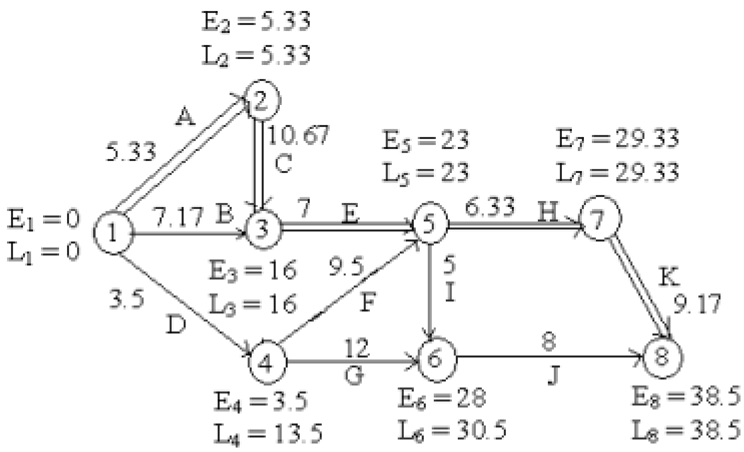
Find the earliest and latest expected time to each event and also critical path in the network.

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