CPM: Network Analysis

Chapter: 8

PERT Vs CPM

PERT (Program Evaluation and Review Technique) and CPM (Critical Path Method) are both project management tools used to plan and control project activities, but there are some differences between the two.

- **Approach:** The main difference between PERT and CPM is in their approach. PERT uses a *probabilistic* approach, while CPM uses a *deterministic* approach.

- **Focus:** PERT primarily focuses on time estimation and scheduling, while CPM focuses on cost control and resource allocation.
- **Activity Dependency:** PERT allows for activity dependency to be represented as a probability distribution, while CPM assumes activity dependency as fixed.
- **Probability Distribution:** PERT uses a probability distribution for estimating activity duration, while CPM uses a single value.

- **Use:** PERT is often used for research and development projects, where there is a high degree of uncertainty, while CPM is commonly used for construction and engineering projects, where the sequence of activities is well-defined.

In summary, PERT and CPM are both useful project management tools that serve different purposes. PERT is suitable for complex and uncertain projects, while CPM is more appropriate for projects with a clear path of activities.

CPM Process

CPM (Critical Path Method) is a project management technique used to identify the critical path of a project, which is the sequence of activities that determines the earliest completion date of the project. The CPM process involves the following steps:

- Define the scope of the project
- Identify the activities
- Determine the activity sequence
- Estimate the duration of each activity
- Determine the critical path
- Develop the project schedule
- Monitor the project

Overall, the CPM process is a systematic and structured approach to project management that helps ensure that projects are completed on time and within budget.

CPM: Networks

- CPM networks are ***activity oriented*** while the PERT networks are event-oriented.
- The arrows representing activities or jobs are labelled with some description of activity.
- Each activity is represented by an arrow
- The sequence of the activities is shown by the sequence of the arrows.
- However, both the activities and events can be labelled if so desired.

Numbering the Events

Though CPM networks are activity-oriented, the events constitute important control points. The events should, therefore, be so numbered that they reflect the logical sequence of the activities. This can be done by following the Fulkerson’s Rule as used in the case of PERT networks.

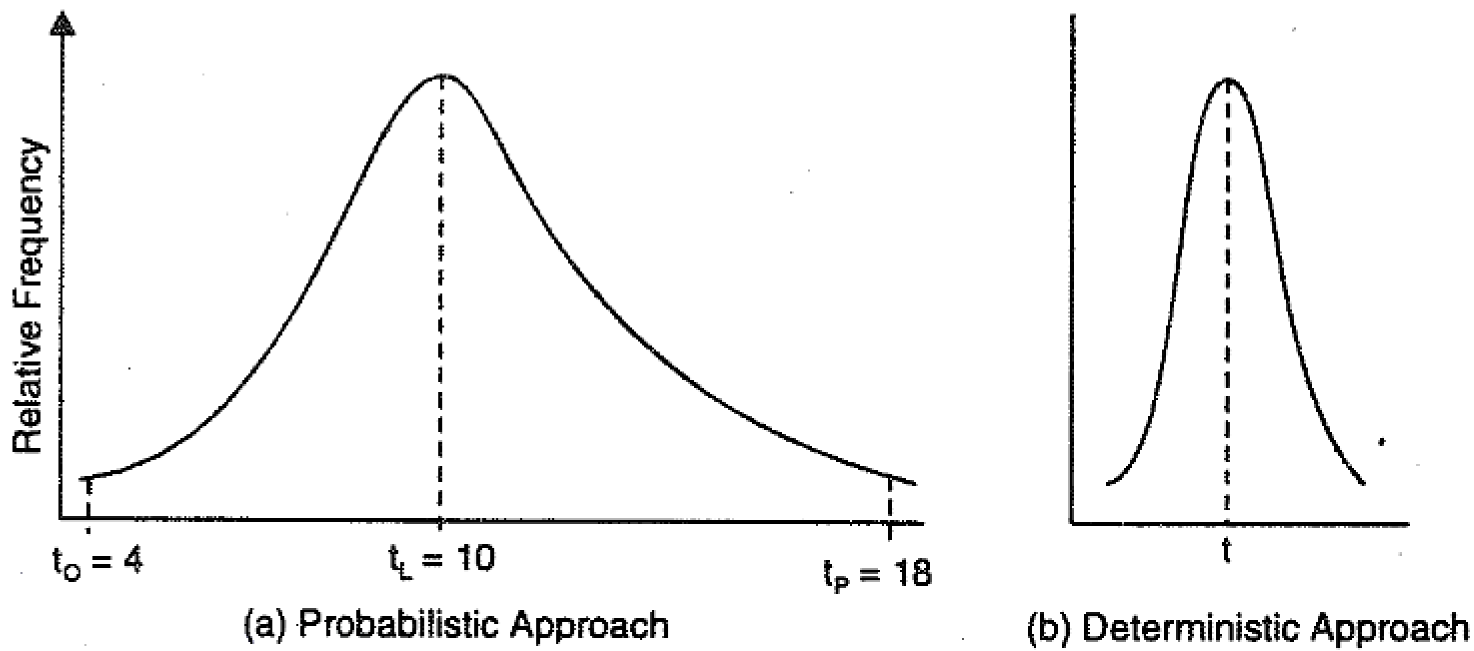
Activity Time Estimate

Two approaches may be used to estimate the duration of activity completion.

1. *Probabilistic Approach*
2. *Deterministic Approach*

- *In case of CPM networks*, deterministic approach is used, in which we assume that we know enough about each job or operation, so that a single time estimate is sufficiently accurate to give reasonable results.
- No uncertainties are taken into consideration.

- The range of variance is very narrow and we approach towards a more realistic model



ACTIVITY TIME ESTIMATE.

Earliest Event Time

The earliest occurrence time or earliest event time (T_E) is the earliest time at which an event can occur. *It is the time by which all the activities discharging into the event under consideration are completed.*

$$T_E^j = T_E^i + t^{ij}$$

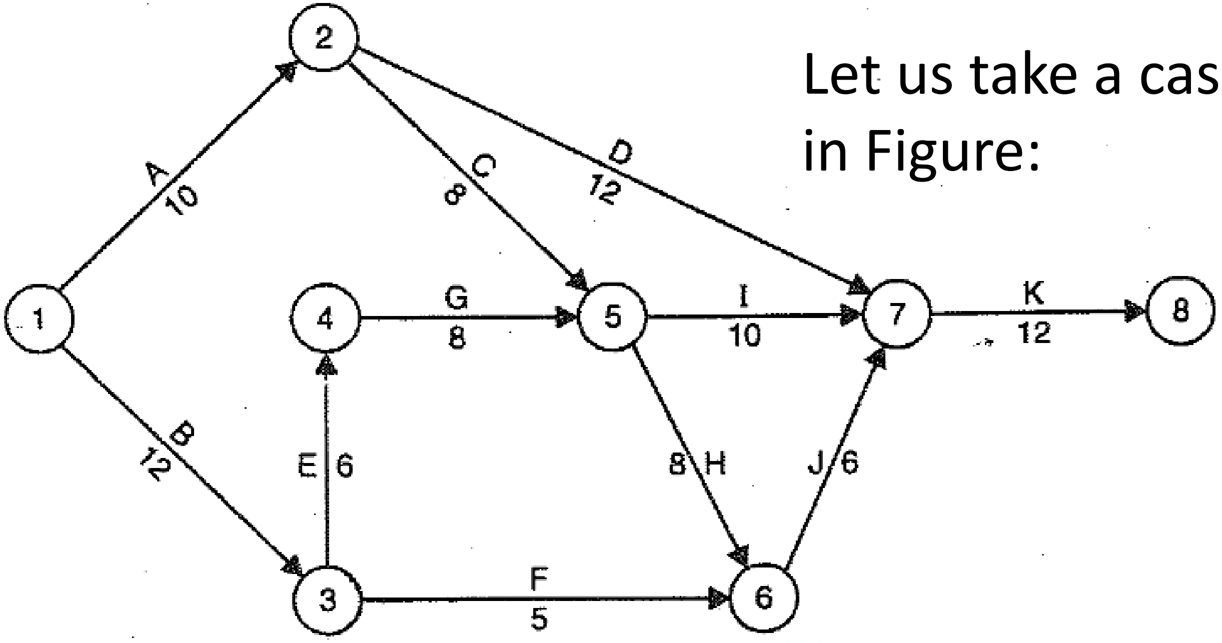
T_E^i = earliest occurrence time for the tail event

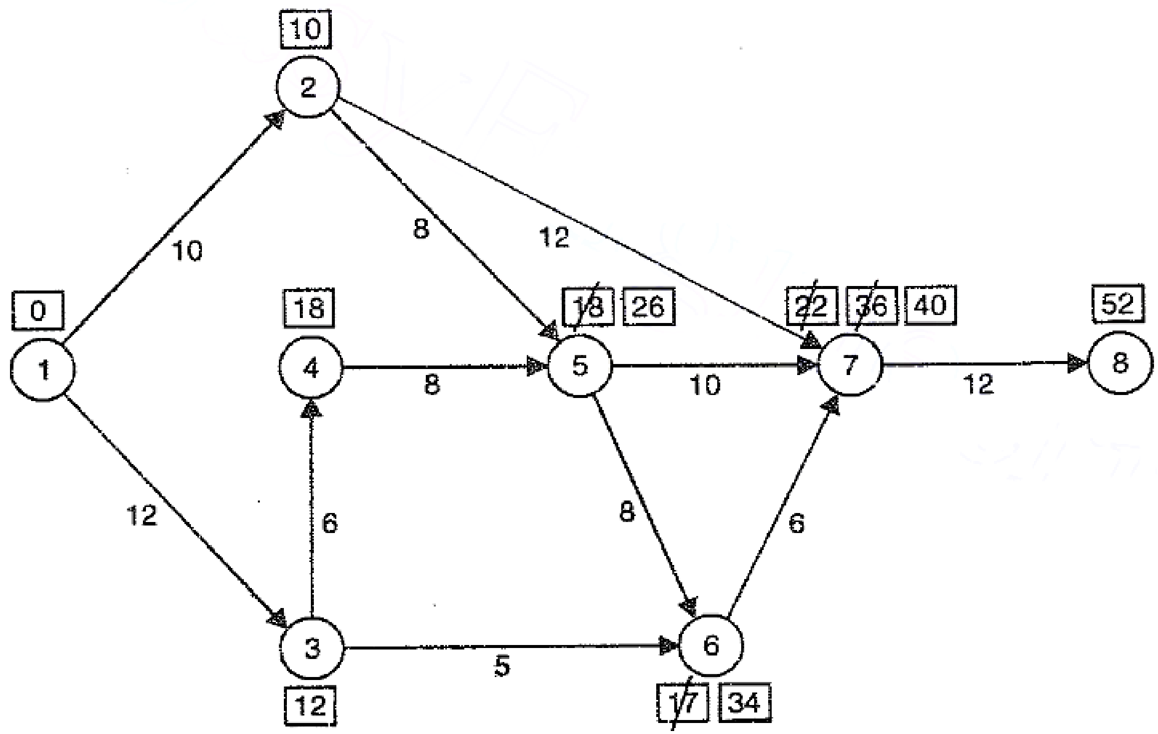
T_E^j = earliest occurrence time for the head event

ij = activity under consideration

t^{ij} = time of completion of activity ij

Let us take a case of a network shown in Figure:





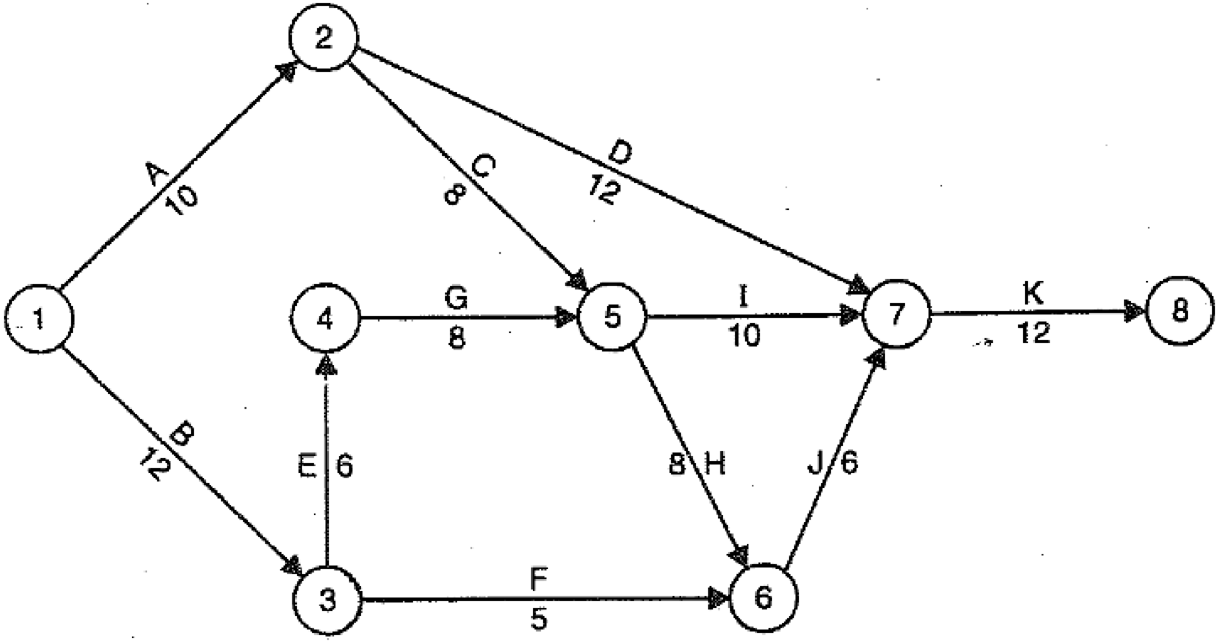
Computations for Earliest Event Time (T_E^j)

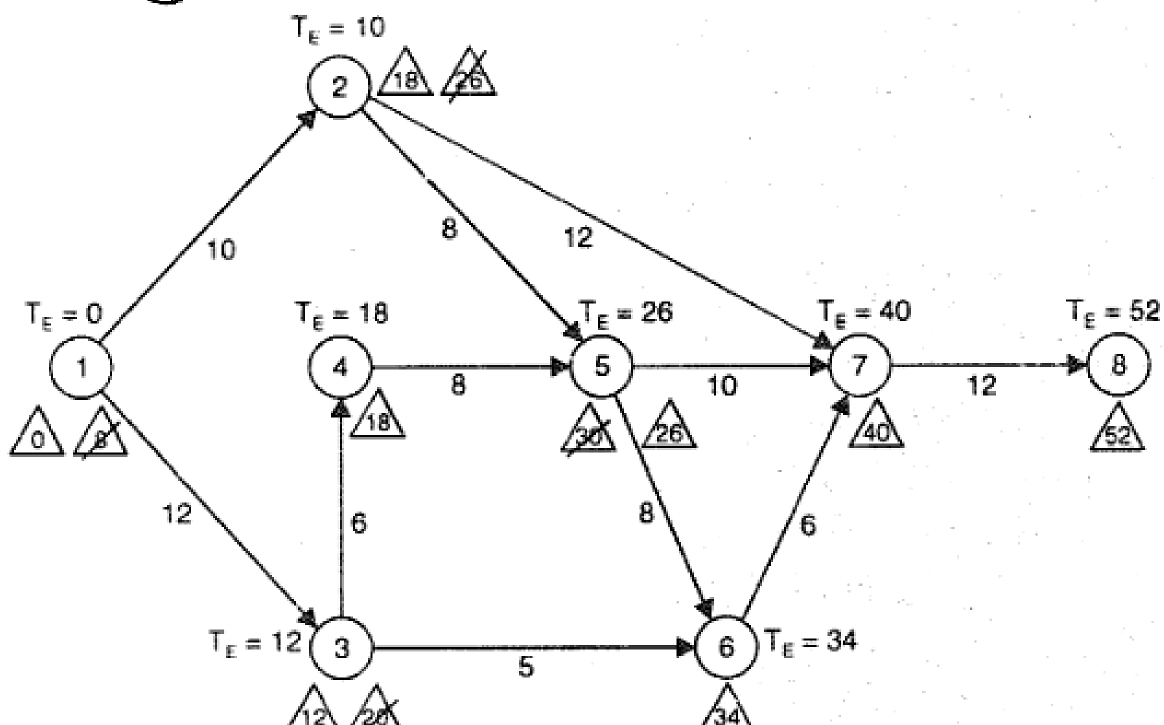
Successor event j	*Predecessor event i*	*Activity i—j*	t^{ij}	T_E^j
8	7	7—8	12	<u>52</u>
7	6	6—7	6	<u>40</u>
	5	5—7	10	36
	2	2—7	12	22
6	5	5—6	8	<u>34</u>
	3	3—6	5	27
5	4	4—5	8	<u>26</u>
	2	2—5	8	18
4	3	3—4	6	<u>18</u>
3	1	1—3	12	<u>12</u>
2	1	1—2	10	

Latest Allowable Occurrence Time

The *latest allowable occurrence time* or the *latest event time* is the latest time by which an event must occur to keep the project on schedule. It is denoted by symbol T_L. If the *scheduled completion time* (T_S) of the project is given, the latest *event time* of the end event will be equal to T_S. If the scheduled completion time is not specified, then T_L is taken equal to the earliest event time T_E.

$$T_L^i = T_L^j - t^{ij} \text{ (for single path)}$$
$$T_L^i = (T_L^j - t^{ij})_{min} \text{ for multiple paths.}$$


1
2
3
4
5
6
7
8
A 10
B 12
C 8
D 12
E 6
F 5
G 8
H 8
I 10
J 6
K 12


$T_E = 0$
$T_E = 10$
$T_E = 12$
$T_E = 18$
$T_E = 26$
$T_E = 34$
$T_E = 40$
$T_E = 52$
1
2
3
4
5
6
7
8
10
12
8
12
6
8
5
10
8
6
12
0
8
18
26
12
20
18
30
26
34
40
52

Computation of Latest Event Occurrence Time (T_L)

Predecessor event i	*Successor event j*	*Activity i—j*	t^{ij}	T_L^i
7	8	7—8	12	<u>40</u>
6	7	6—7	6	<u>34</u>
5	7	5—7	10	30
	6	5—6	8	<u>26</u>
4	5	4—5	8	<u>18</u>
3	6	3—6	5	29
	4	3—4	6	<u>12</u>
2	7	2—7	12	28
	5	2—5	8	<u>18</u>
1	3	1—3	12	<u>0</u>
	2	1—2	10	8

Computations of T_E and T_L

Event No.	*Earliest event time* (↓)				*Latest event time* (↓)			
	Predecessor event (i)	t^{ij}	T_E^j	T_E	*Successor event (j)*	t^{ij}	T_L^i	T_L
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
1	—	—	0	0	2 3	10 12	8 0	0
2	1	10	10	10	5 7	8 12	18 28	18
3	1	12	12	12	4 6	6 5	12 29	12
4	3	6	18	18	5	8	18	18
5	2 4	8 8	18 26	26	6 7	8 10	26 30	26
6	3 5	5 8	17 34	34	7	7	34	34
7	2 5 6	12 10 6	22 36 40	40	8	12	40	40
8	7	12	52	52	—		52	52

Start and Finish Times of an Activity

Since CPM networks are activity oriented, the following activity times are useful for network computations:

- Earliest Start Time
- Earliest Finish Time
- Latest Start Time
- Latest Finish Time

Earliest Start Time

The *earliest start time* for an activity is the earliest time by which it can commence. This is naturally equal to the earliest event time associated with the tail of the activity arrow. It is abbreviated by EST.

Thus, EST = Earliest event time at its tail.

If the activity is denoted by $i—j$, and if the earliest event time at its tail is T_E^i, we have

$$EST = T_E^i$$

Earliest Finish Time

If an activity proceeds from its early start time and takes the estimated duration for completion, then it will have an early finish. Hence *earliest finish time* (EFT) for an activity is defined as the earliest time by which it can be finished. This is evidently equal to the earliest start time plus estimated duration of the activity :

$$\text{EFT} = \text{earliest start time} + \text{activity duration}$$

or

$$\text{EFT} = T_{\text{E}}^{i} + t^{ij}$$

Latest Finish Time

The latest finish time for an activity is the latest time by which an activity can be finished without delaying the completion of the project. Naturally, the latest finish time for an activity will be equal to the latest allowable occurrence time for the event at the head of the arrow. Hence

$$\text{LFT} = \text{Latest event time at the head of activity arrow}$$

or $$\text{LFT} = T_L^j$$

Latest Start Time

Latest start time for an activity is the *latest time* by which an activity can be started without delaying the completion of the project. For 'no delay' condition to be fulfilled it should be naturally equal to the latest finish time (LFT) minus the activity duration.

$$\therefore \qquad \text{LST} = \text{LFT} - \text{Activity duration.}$$

Since the Latest Finish Time (LFT) is equal to T_L^j

$$\text{LST} = T_L^j - t^{ij}$$

The above four activity times are generally written on the activity arrow, along with *activity name* and *activity time* (t^{ij}), as illustrated in the figure.

FLOAT

In CPM, "float" is another term for slack or the amount of time that an activity can be delayed without affecting the project completion time. It represents the amount of time that an activity can be delayed beyond its early start date without delaying the early start date of its successor activity.

Just as the ***Slack*** denotes the flexibility range within which an event can occur, ***Float*** denotes the range within which an activity start time or its finish time may fluctuates without affecting the completion of the project.

Floats are of the following types:

1. Total Float
2. Free Float
3. Independent Float
4. Interfering Float

Total Float

Total float is the amount of time an activity can be delayed without delaying the project completion date.

Sometimes there is a difference in the time available for an activity with respect to the actual time required to perform the activity. That difference is known as the ***Total Float.***

Consider an activity i—j. The time duration available for this activity is equal to the difference between its earliest start time (T_E^i) and the latest finish time (T_L^j) :

$\therefore$ Maximum time available = $T_L^j - T_E^i$

activity time required = t^{ij}

$\therefore$ Total float (F_T) = time available – time required

$\therefore$ Total float (F_T) = time available – time required

or $$F_T = (T_L^j - T_E^i) - t^{ij}$$

or $$F_T = T_L^j - (T_E^i + t^{ij})$$

Also $$F_T = (T_L^j - t^{ij}) - T_E^i$$

Now, if $T_E^j - t^{ij} = \text{LST}$

and And also, if $T_E^i = \text{EST}$

$\therefore$ $$F_T = \text{LST} - \text{EST}$$

Similarly $$F_T = \text{LFT} - \text{EFT}$$

Free Float

Free float is the amount of time an activity can be delayed without delaying the early start of any of its immediate successor activities. The concept of *free float* is based on the possibility that all the events occur at their earliest time (i.e. All activities start as early as possible).

To get a clear concept of the free float, consider activity i-j and its successor activity j-k. Events i and j has earliest occurrence times as T_E^i and T_E^j. Earliest start time for activity ij will be T_E^i, while EST for j-k will be T_E^j. However, if t^{ij} is the activity time, activity i-j will be complete by $(T_E^i + t^{ij})$ time, while activity j-k cannot start because its EST (T_E^j) is greater than $(T_E^i + t^{ij})$. The difference between the two is the free float for i-j.

$$\therefore \quad F_F \text{ for } i\text{-}j = T_E^j - (T_E^i + t^{ij})$$

1

But $(T_E^i + t^{ij})$ is the early finish time (EFT) of the activity i-j, while T_E^j is the early start time for activity j-k.

$$\therefore \quad F_F \text{ for } ij = \text{EST for successor activity}$$

$$- \text{EFT of present activity.}$$

or $$F_F \text{ for } ij = T_E^j - \text{EFT} \qquad 2$$

Again, from Eqn. 8.9 (a),

$$F_T = (T_L^j - T_E^i) - t^{ij}$$

$$\therefore \quad T_E^i + t^{ij} = T_L^j - F_T$$

Substituting in Eqn. 8.11,

$$F_F = T_E^j - (T_L^j - F_T)$$

or $$F_F = F_T - (T_L^j - T_E^j) = F_T - S_j \qquad 3$$

Equation 3 gives another method of calculating the free float. *It is the difference of the total float and the head event slack.* If head slack is zero, free float will be equal to the total float.