Chapter

15 Network Scheduling by PERT/CPM

15.1 INTRODUCTION

Network scheduling is a technique used for planning and scheduling large projects, in the fields of construction, maintenance, fabrication and purchasing of computer systems, etc. It is a method of minimizing the trouble spots such as production, delays and interruptions, by determining critical factors and co-ordinating various parts of the overall job.

There are two basic planning and controlling techniques that utilize a network to complete a predetermined project or schedule. These are Programme Evaluation Review Technique (PERT) and Critical Path Method (CPM).

A project is defined as a combination of interrelated activities, all of which must be executed in a certain order for its completion.

The work involved in a project can be divided into three phases, corresponding to the management functions of planning, scheduling and controlling.

Planning: This phase involves setting the objectives of the project as well as the assumptions to be made. It also involves the listing of tasks or jobs that must be performed in order to complete a project under consideration. In this phase, in addition to the estimates of costs and duration of the various activities, the manpower, machines and materials required for the project are also determined.

Scheduling: This consists of laying the activities according to their order of precedence and determining the following:

(*i*) The start and finish times for each activity.
(*ii*) The critical path on which the activities require special attention.
(*iii*) The slack and float for the non-critical paths.

Controlling: This phase is exercised after the planning and scheduling. It involves the following:

(*i*) Making periodical progress reports
(*ii*) Reviewing the progress
(*iii*) Analyzing the status of the project
(*iv*) Making management decisions regarding updating, crashing and resource allocation, etc.

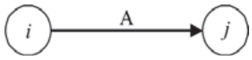
15.2 BASIC TERMS

To understand the network techniques, one should be familiar with a few basic terms of which both CPM and PERT are special applications.

Network: It is the graphic representation of logically and sequentially connected arrows and nodes, representing activities and events in a project. Networks are also called *arrow diagrams*.

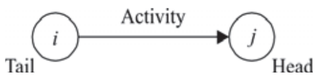
Activity: An activity represents some action and is a time consuming effort necessary to complete a particular part of the overall project. Thus, each and every activity has a point of time where it begins and a point where it ends.

It is represented in the network by an arrow,

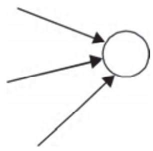


Here *A* is called the *activity*.

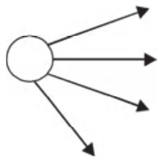
Event The beginning and end points of an activity are called *events or nodes*. Event is a point in time and does not consume any resources. It is represented by a numbered circle. The head event called the *jth event* always has a number higher than the tail event, which is also called the *ith event*.



Merge and burst events It is not necessary for an event to be the ending event of only one activity as it can be the ending event of two or more activities. Such an event is defined as a *merge event*.



If the event happens to be the beginning event of two or more activities, it is defined as a *burst event*.



Preceding, succeeding and concurrent activities Activities that must be accomplished before a given event can occur, are termed as *preceding activities.*

Activities that cannot be accomplished until an event has occurred, are termed as *succeeding activities.*

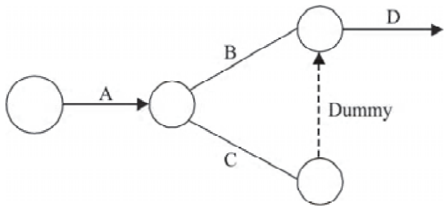
Activities that can be accomplished concurrently i.e., activities taking place at the same time or in the same location, are known as *concurrent activities.*

This classification is relative, which means that one activity can be preceding to a certain event, and the same activity can be succeeding to some other event or it may be a concurrent activity with one or more activities.

Dummy activity Certain activities, which neither consume time nor resources but are used simply to represent a connection or a link between the events are known as *dummies*. It is shown in the network by a dotted line. The purpose of introducing dummy activity is:

(*i*) to maintain uniqueness in the numbering system, as every activity may have a distinct set of events by which the activity can be identified.

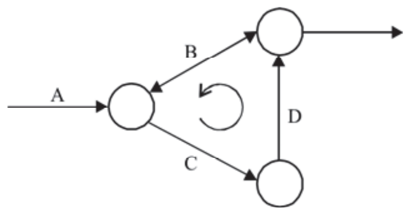
(*ii*) to maintain a proper logic in the network.



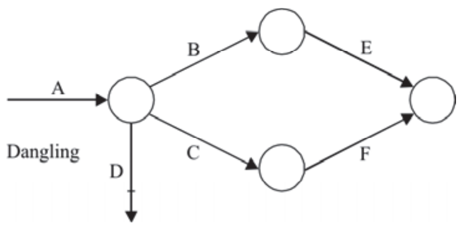
15.3 COMMON ERRORS

Following are the three common errors in a network construction:

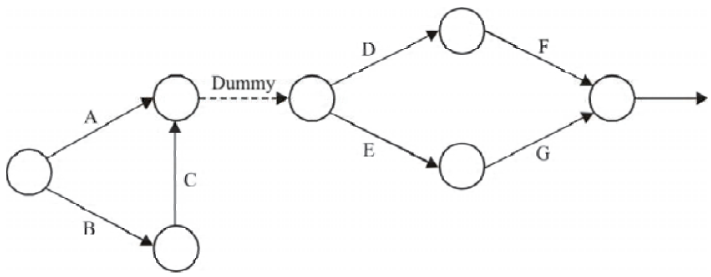
***Looping* (*cycling*)** In a network diagram, a looping error is also known as *cycling error*. Drawing an endless loop in a network is known as *error of looping*. A loop can be formed, if an activity is represented as going back in time.



Dangling To disconnect an activity before the completion of all the activities in a network diagram, is known as *dangling*.



Redundancy If a dummy activity is the only activity emanating from an event and can be eliminated, it is known as *redundancy*.



15.4 RULES OF NETWORK CONSTRUCTION

There are a number of rules in connection with the handling of events and activities of a project network that should be followed.

(*i*) Try to avoid the arrows that cross each other.
(*ii*) Use straight arrows.
(*iii*) No event can occur until every activity preceding it has been completed.
(*iv*) An event cannot occur twice, i.e., there must be no loops.
(*v*) An activity succeeding an event cannot be started until that event has occurred.
(*vi*) Use arrows from left to right. Avoid mixing two directions, vertical and standing arrows may be used, if necessary.
(*vii*) Dummies should be introduced only, if it is extremely necessary.
(*viii*) The network has only one entry point called the *start event* and one point of emergence called the *end or terminal event*.

15.5 NUMBERING THE EVENTS (FULKERSON'S RULE)

After the network is drawn in a logical sequence, every event is assigned a unique number. The number sequence must be such so as to reflect the flow of the network. In numbering the events, the following rules should be observed.

(*i*) Event numbers should be unique.
(*ii*) Event numbering should be carried out on a sequential basis, from left to right.
(*iii*) The initial event, which has all outgoing arrows with no incoming arrow is numbered as 1.
(*iv*) Delete all the arrows emerging from all the numbered events. This will create at least one new start event, out of the preceding events.
(*v*) Number all new start events 2, 3 and so on. Repeat this process until the terminal event without any successor activity is reached. Number the terminal node suitably.

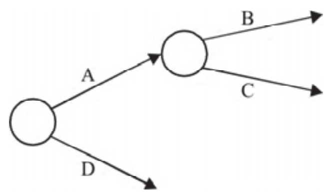
Note: The head of an arrow should always bear a number higher than the one assigned to the tail of the arrow.

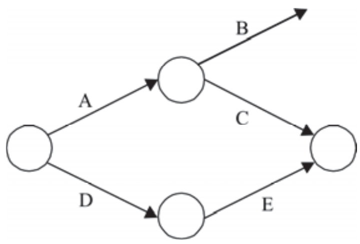
15.5.1 Construction of Network

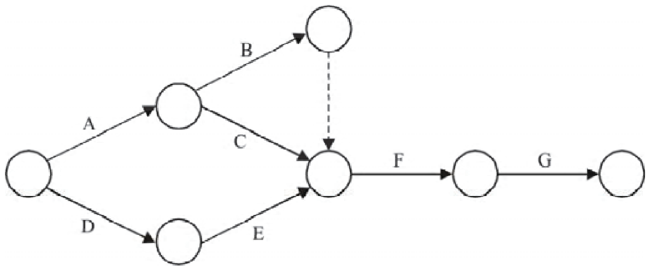
Example 15.1 Construct a network for the project whose activities and precedence relationships are as given below:

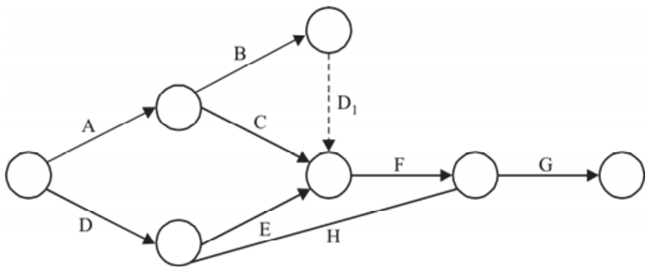
Activities	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*
Immediate Predecessor	–	*A*	*A*	–	*D*	*B*, *C*, *E*	*F*	*D*	*G*, *H*

Solution From the given constraints, it is clear that *A* and *D* are the starting activities and *I* the terminal activity. *B* and *C* are starting with the same event and are both the predecessors of the activity *F*. Also, *E* has to be the predecessor of both *F* and *H*. Hence, we have to introduce a dummy activity.



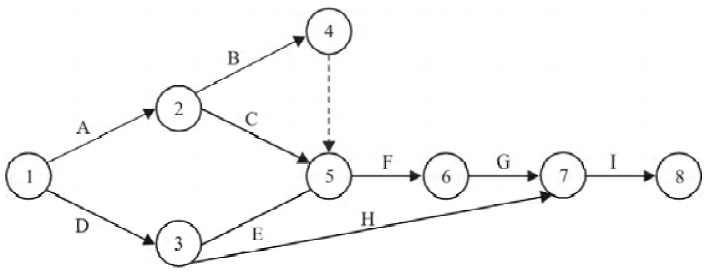






D_1 is the dummy activity.

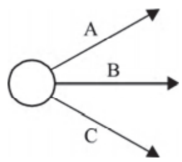
Finally, we have the following network.

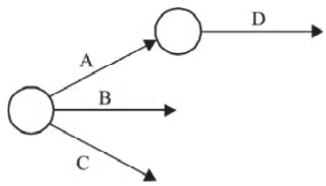


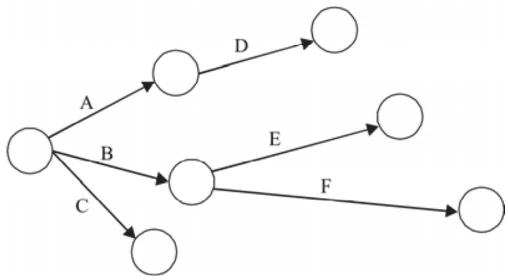
Example 15.2 Construct a network for each of the projects whose activities and their precedence relationships are given below.

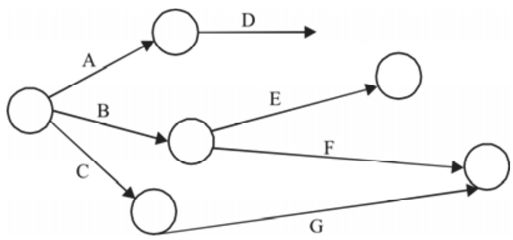
Activity	***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***	***J***	***K***
Predecessor	–	–	–	*A*	*B*	*B*	*C*	*D*	*E*	*H, I*	*F, G*

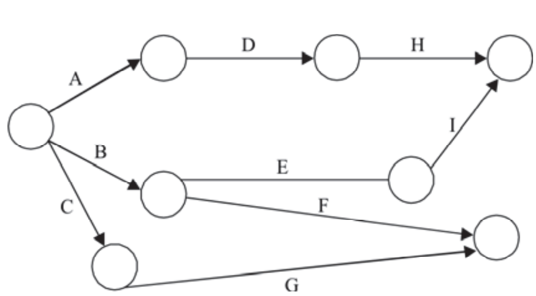
Solution A, B and C are the concurrent activities as they start simultaneously. B becomes the predecessor of activities E and F. Since the activities J and K have two preceding activities, a dummy may be introduced (if possible).

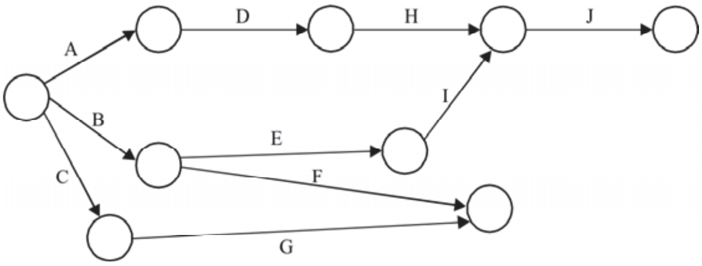




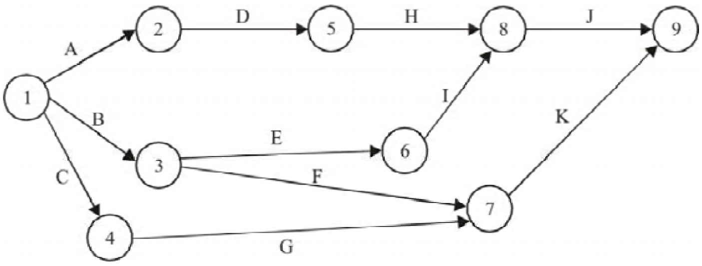








Finally we have,

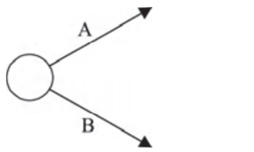


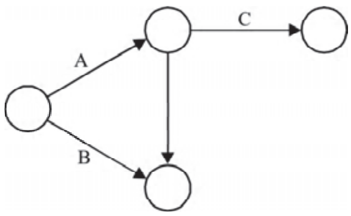
Example 15.3 $A < C, D, I; B < G, F; D < G, F; F < H, K; G, H < J; I, J, K < E$

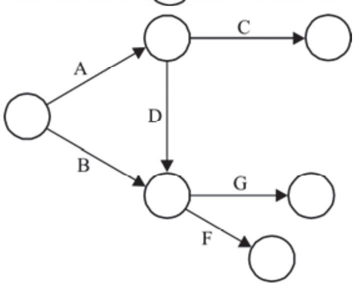
Solution Given $A < C$, which means that C cannot be started until A is completed. That is, A is the preceding activity to C. The above constraints can be given in the following table.

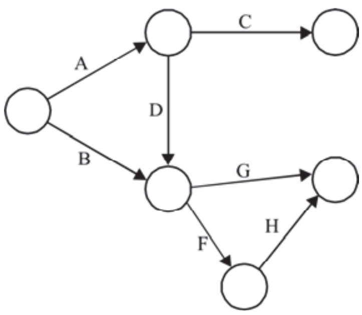
Activity	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*
Predecessor	–	–	*A*	*A*	*I, J, K*	*B, D*	*B, D*	*F*	*A*	*G, H*	*F*

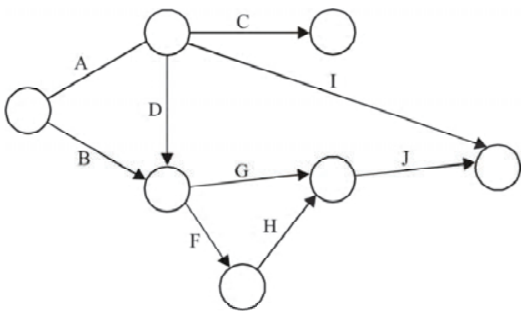
A and *B* are the starting activities, and *E* is the terminal activity.

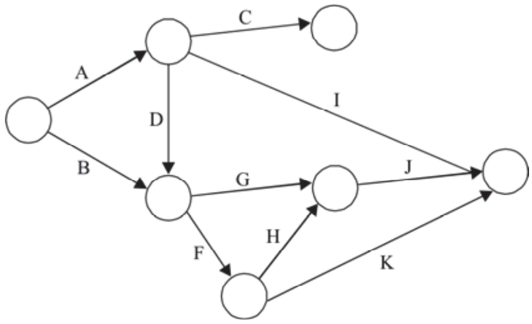




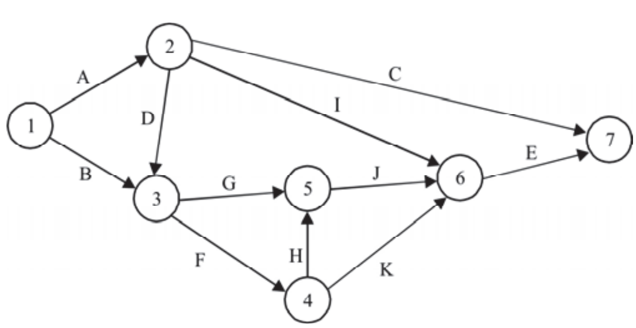








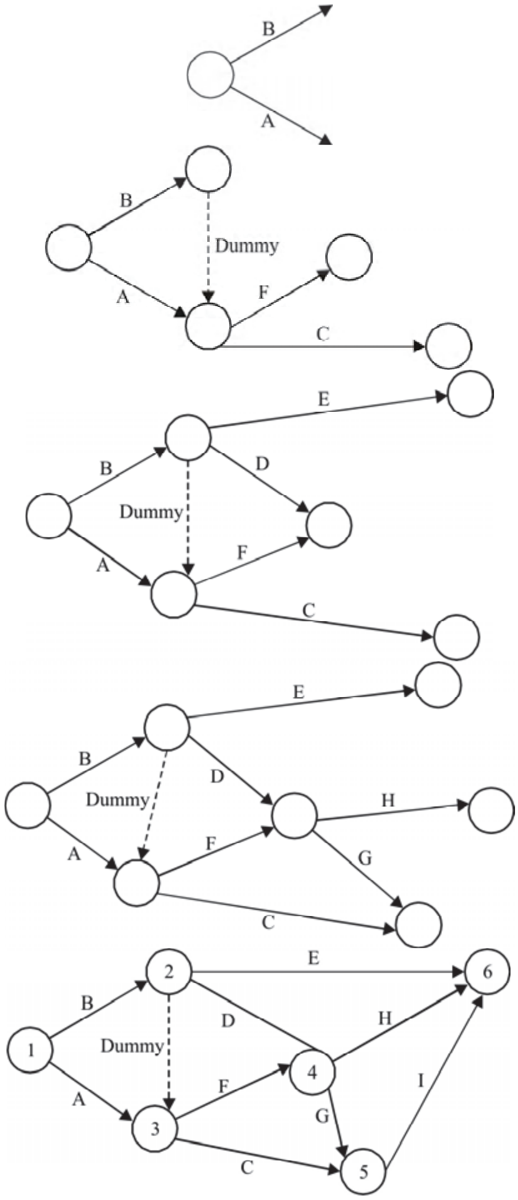
Finally, we have,



Example 15.4

Activities	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*
Immediate Predecessor	–	–	*A*, *B*	*B*	*B*	*A*, *B*	*F*, *D*	*F*, *D*	*C*, *G*

Solution *A* and *B* are concurrent activities as they start simultaneously. *I* is the terminal activity. Since the activities *C* and *F* are coming from both activities *A* and *B*, we need to introduce a dummy activity.

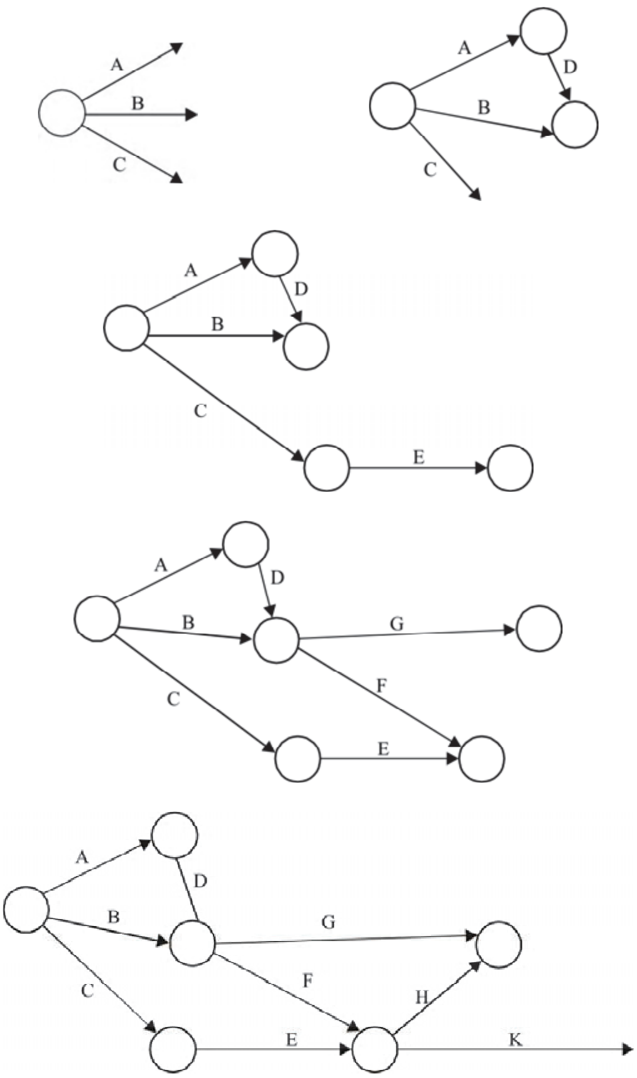


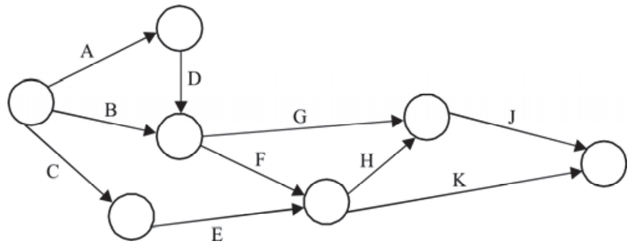
Example 15.5 A, B and C can start simultaneously

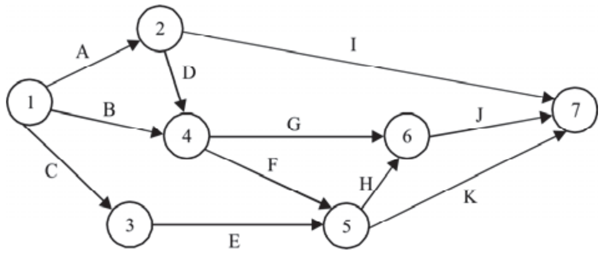
$A < D, I; B < G, F; D < G, F; C < E; E < H, K; F < H, K; G, H < J.$

Solution The above constraints can be formatted into a table.

Activity	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*
Predecessor Activity	–	–	–	*A*	*C*	*B*, *D*	*B*, *D*	*E*, *F*	*A*	*G*,	*E*, *F*







15.6 TIME ANALYSIS

Once the network of a project is constructed, the time analysis of the network becomes essential for planning various activities of the project. Activity time is a forecast of the time for an activity which is expected to take from its starting point to its completion (under normal conditions).

We shall use the following notation for basic scheduling computations.

(i,j) = Activity (i, j) with tail event i and head event j
T_{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 basic scheduling computation can be put under the following three groups.

15.6.1 Forward Pass Computations (For Earliest Event Time)

Before starting computations, the occurrence time of the initial network event is fixed. The forward pass computation yields the earliest start and the earliest finish time for each activity (i, j) and indirectly the earliest occurrence time for each event namely E_i. This consists of the following three steps:

Step 1 The computations begin from the start node and move towards the end node. Let zero be the starting time for the project.

Step 2 Earliest starting time $(ES)_{ij} = E_i$ is the earliest possible time when an activity can begin, assuming that all of the predecessors are also started at their earliest starting time. Earliest finish time of activity (i, j) is the earliest starting time + the activity time.

$$(EF)_{ij} = (ES)_{ij} + t_{ij}$$

Step 3 Earliest event time for event j is the maximum of the earliest finish time of all the activities, ending at that event.

$$E_j = \underset{i}{\text{Max}}\,(E_i + t_{ij})$$

The computed 'E' values are put over the respective rectangles representing each event.

15.6.2 Backward Pass Computations (For Latest Allowable Time)

The latest event time (L) indicates the time by which all activities entering into that event must be completed without delaying the completion of the project. These can be calculated by reversing the method of calculations used for the earliest event time. This is done in the following steps.

Step 1 For ending event assume, $E = L$.

Step 2 Latest finish time for activity (i, j) is the target time for completing the project,

$$(LF_{ij}) = L_j$$

Step 3 Latest starting time of the activity (i, j) = latest completion time of (i, j) – the activity time

$$\begin{aligned} LS_{ij} &= LF_{ij} - t_{ij} \\ &= L_j - t_{ij} \end{aligned}$$

Step 4 Latest event time for event i is the minimum of the latest start time of all activities originating from the event.

$$\boxed{L_i = \underset{j}{\text{Min}}(L_j - t_{ij})}$$

The computed 'L' values are put over the respective triangles representing each event.

15.6.3 Determination of Floats and Slack Times

Float is defined as the difference between the latest and the earliest activity time.

Slack is defined as the difference between the latest and the earliest event time.

Hence, the basic difference between the slack and float is that slack is used for events only; whereas float is used for activities.

There are mainly three kinds of floats as given below.

Total float It refers to 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, the total float of an activity (i, j) is the difference between the latest start time and the earliest start time of that activity.

Hence, the total float for an activity (i, j) denoted by $(TF)_{ij}$ is calculated by the formula,

$(TF)_{ij}$ = (Latest start – Earliest start) for activity (i, j)

$$\boxed{\begin{aligned} \text{i.e., } (TF)_{ij} &= (LS)_{ij} - (ES)_{ij} \\ \text{or } (TF)_{ij} &= (L_j - E_i) - t_{ij} \end{aligned}}$$

where, E_i and L_j are the earliest time and latest time for the tail event i and head event j and t_{ij} is the normal time for the activity (i, j). This is the most important type of float as it concerns the overall project duration.

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 succeeding activity.

Mathematically, the free float for activity (i, j) denoted by $(FF)_{ij}$ can be calculated by the formula,

$$FF_{ij} = (E_j - E_i) - t_{ij}$$
$$(FF)_{ij} = \text{Total float} - \text{Head event slack}$$

Head event slack $= L_j - E_j$

This float is concerned with the commencement of the subsequent activity.

The free float can take values from zero up to total float, but it cannot exceed total float. This float is very useful for rescheduling an activity with minimum disruption in earlier plans.

Independent float The amount of time by which the start of an activity can be delayed, without affecting the earliest start time of any immediately following activities, assuming that the preceding activity has finished at its latest finish time.

Mathematically, independent float of an activity (i, j) denoted by $(IF)_{ij}$ can be calculated by the formula,

$$IF_{ij} = (E_j - L_i) - t_{ij}$$
or
$$(IF)_{ij} = \text{Free Float} - \text{Tail event slack}$$

where tail event slack is given by,

$$\text{Tail event slack} = L_i - E_i$$

The negative independent float is always taken as zero. This float is concerned with prior and subsequent activities.

$$IF_{ij} \leq FF_{ij} \leq TF_{ij}$$

Notes: (*i*) If the total float TF_{ij} for any activity (i, j) is zero, then such an activity is called *critical activity*.

(*ii*) The float can be used to reduce project duration. While doing this, the float of not only that activity, but that of other activities will also change.

Critical activity An activity is said to be critical, if a delay in its start cause a further delay in the completion of the entire project.

Critical path The sequence of critical activities in a network which determines the duration of a project is called the critical path. It is the longest path in the network, from the starting event to the ending event and defines the minimum time required to complete the project. In the network it is denoted by a double line and identifies all the critical activities of the project. Hence, for the activities (i, j) to lie on the critical path, following conditions must be satisfied.

(*a*) $ES_i = LF_i$

(*b*) $ES_j = LF_j$

(*c*) $ES_j - ES_i = LF_j - LF_i = t_{ij}$

ES_i and ES_j are the earliest start and finish time of the events i and j.

LF_i and LF_j are the latest start and finish time of the events i and j.

15.7 CRITICAL PATH METHOD (CPM)

The critical path method (CPM) is a step-by-step procedure for scheduling the activities in a project. It is an important tool related to effective project management. The iterative procedure of determining the critical path is as follows:

Step 1 List all the jobs and then draw an arrow (network) diagram. Each job is indicated by an arrow with the direction of the arrow showing the sequence of jobs. The length of the arrows has no significance. The arrows are placed based on the predecessor, successor and concurrent relation within the job.

Step 2 Indicate the normal time (t_{ij}) for each activity (i, j) above the arrow, which is deterministic.

Step 3 Calculate the earliest start time and the earliest finish time for each event and write the earliest time E_i for each event i in the $\square$. Also calculate the latest finish and latest start time. From this we calculate the latest time L_j for each event j and put it in the Δ.

Step 4 Tabulate the various times, namely, normal time, earliest time and latest time on the arrow diagram.

Step 5 Determine the total float for each activity by taking the difference between the earliest start and the latest start time.

Step 6 Identify the critical activities and connect them with the beginning and the ending events in the network diagram by double line arrows. This gives the critical path.

Step 7 Calculate the total project duration.

Note: The earliest start and finish time of an activity, as well as the latest start and finish time of an activity are shown in the table. These are calculated by using the following hints.

To find the earliest time, we consider the tail event of the activity. Let the starting time of the project namely $ES_i = 0$. Add the normal time with the starting time, to get the earliest finish time. The earliest starting time for the tail event of the next activity is given by the maximum of the earliest finish time for the head event of the previous activity.

Similarly, to get the latest time, we consider the head event of the activity.

The latest finish time of the head event of the final activity is given by the target time of the project. The latest start time can be obtained by subtracting the normal time of that activity. The latest finish time for the head event of the next activity is given by the minimum of the latest start time for the tail event of the previous activity.

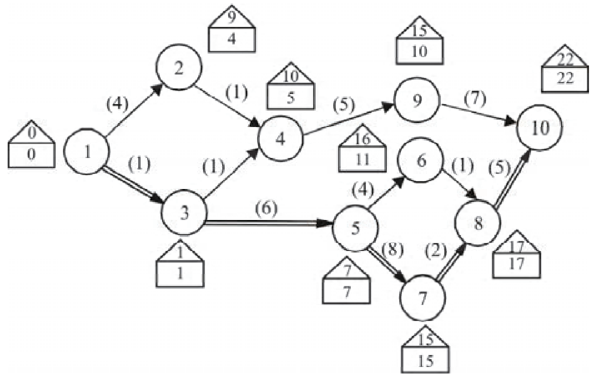
Example 15.6 A project schedule has the following characteristics.

Activity	*1–2*	*1–3*	*2–4*	*3–4*	*3–5*	*4–9*	*5–6*	*5–7*	*6–8*	*7–8*	*8–10*	*9–10*
Time (days)	4	1	1	1	6	5	4	8	1	2	5	7

From the above information, you are required to:

1. Construct a network diagram.
2. Compute the earliest event time and latest event time.
3. Determine the critical path and total project duration.
4. Compute total and free float for each activity.

Solution First we construct the network with the given constraints (here we get it by just connecting the event numbers).



The following table gives the critical path as well as total and free floats calculation.

Activity	*Normal time*	*Earliest*		*Latest*		*TF*	*FF*
		Start	*Finish*	*Start*	*Finish*		
1 – 2	4	0	4	5	9	5	5 – 5 = 0
1 – 3	1	0	1	0	1	0	0
2 – 4	1	4	5	9	10	5	0
3 – 4	1	1	2	9	10	8	3
3 – 5	6	1	7	1	7	0	0
4 – 9	5	5	10	10	15	5	0
5 – 6	4	7	11	12	16	5	0
5 – 7	8	7	15	7	15	0	0
6 – 8	1	11	12	16	17	5	0
7 – 8	2	15	17	15	17	0	0
8 –10	5	17	22	17	22	0	0
9 – 10	7	10	17	15	22	5	5

The earliest and latest calculations are shown below.

Forward pass calculation In this we estimate the earliest start (Es_i) and finish times (Es_j). The earliest time for the event i is given by,

$$E_i = \underset{i}{\text{Max}}\ (ES_i + t_{ij})$$
$$ES_1 = 0 = E_1 = 0$$
$$E_2 = ES_2 = ES_1 + t_{12} = 0 + 4 = 4$$
$$E_3 = ES_3 = ES_1 + t_{13} = 0 + 1 = 1$$
$$E_4 = ES_4 = \text{Max}\ (ES_3 + t_{34}, ES_2 + t_{24})$$
$$= \text{Max}\ (1 + 1, 4 + 1) = 5$$
$$E_5 = (E_3 + t_{35}) = 1 + 6 = 7$$
$$E_6 = E_5 + t_{56} = 7 + 4 = 11$$
$$E_7 = E_5 + t_{57} = 7 + 8 = 15$$

$$
\begin{aligned}
E_8 &= \text{Max}\,(E_6 + t_{68}, E_7 + t_{78}) \\
&= \text{Max}\,(11+1, 15+2) = 17 \\
E_9 &= E_4 + t_{49} = 5 + 5 = 10 \\
E_{10} &= \text{Max}\,(E_9 + t_{9,10}, E_8 + t_{8,\,10}) \\
&= \text{Max}\,(10+7, 17+5) = 22.
\end{aligned}
$$

Backward pass calculation In this we calculate the latest finish and the latest start time. The latest time L for an event i is given by $L_i = \underset{j}{\text{Min}}\ (LF_j - t_{ij})$,
where, LF_j is the latest finish time for the event j, t_{ij} is the normal time of the activity.

$$
\begin{aligned}
L_{10} &= 22 \\
L_9 &= L_{10} - t_{9,\,10} = 22 - 7 = 15 \\
L_8 &= L_{10} - t_{8,\,10} = 22 - 5 = 17 \\
L_7 &= L_8 - t_{7,\,8} = 17 - 2 = 15 \\
L_6 &= L_8 - t_{6,\,8} = 17 - 1 = 16 \\
L_5 &= \text{Min}\,(L_6 - t_{5,\,6}, L_7 - t_{5,\,7}) \\
&= \text{Min}\,(16-4, 15-8) = 7 \\
L_4 &= L_9 - t_{4,\,9} = 15 - 5 = 10 \\
L_3 &= \text{Min}\,(L_4 - t_{3,\,4}, L_5 - t_{3,\,5}) \\
&= \text{Min}\,(10-1, 7-6) = 1 \\
L_2 &= L_4 - t_{2,\,4} = 10 - 1 = 9 \\
L_1 &= \text{Min}\,(L_2 - t_{12}, L_3 - t_{13}) = \text{Min}\,(9-4, 1-1) = 0.
\end{aligned}
$$

These calculations are shown in the above table.

To find the TF (Total Float) Considering the activity 1 – 2, *TF* of (1 – 2) = Latest start–Earliest start
$= 5 - 0 = 5$

Similarly $TF\,(2-4) = LS - ES$
$= 9 - 4 = 5$

Free float = *TF* – Head event slack.

Consider the activity 1 – 2

FF of (1 – 2) = *TF* of (1 – 2) – Slack for the head event 2
= 5 – (9 – 4) (from the figure for event 2)
$= 5 - 5 = 0$

FF of (2 – 4) = *TF* of (2 – 4) – Slack for the head event 4
$= 5 - (10 - 5) = 5 - 5 = 0$

Like this we calculate the *TF* and *FF* for the remaining activities.

From the above table we observe that the activities 1 – 3, 3 – 5, 5 – 7, 7 – 9, 8 – 10 are the critical activities as their total float is 0.

Hence, we have the following critical path.

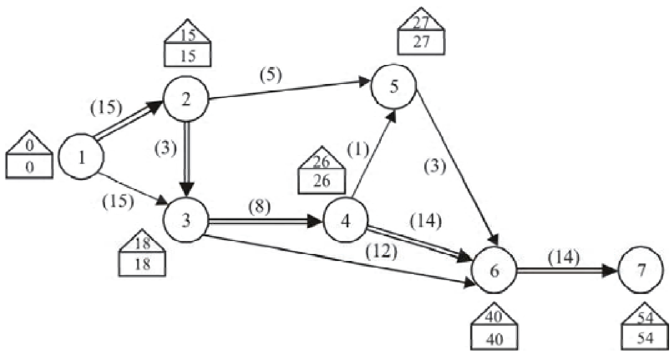
1 – 3 – 5 – 7 – 8 – 10, with the total project duration of 22 days.

Example 15.7 A small maintenance project consists of the following jobs, whose precedence relationships are given below.

Job	*1–2*	*1–3*	*2–3*	*2–5*	*3–4*	*3–6*	*4–5*	*4–6*	*5–6*	*6–7*
Duration (days)	15	15	3	5	8	12	1	14	3	14

1. Draw an arrow diagram representing the project.
2. Find the total float for each activity.
3. Find the critical path and the total project duration.

Solution



Forward pass calculation In this we estimate the earliest start and the earliest finish time ES_j given by,

$ES_j = \underset{i}{\text{Max}}\ (ES_i + t_{ij})$ where, ES_i is the earliest start time and t_{ij} is the normal time for the activity (i, j).

$$ES_1 = 0$$
$$ES_2 = ES_1 + t_{15} = 0 + 15 = 15$$
$$ES_3 = \text{Max}\ (ES_2 + t_{23}, ES_1 + t_{13})$$
$$= \text{Max}\ (15 + 3, 0 + 15) = 18$$
$$ES_4 = ES_3 + t_{34} = 18 + 8 = 26$$
$$ES_5 = \text{Max}\ (ES_2 + t_{25}, ES_4 + t_{45})$$
$$= \text{Max}\ (15 + 5, 26 + 1) = 27$$
$$ES_6 = \text{Max}\ (ES_3 + t_{36}, ES_4 + t_{46}, ES_5 + t_{56})$$
$$= \text{Max}\ (18 + 12, 26 + 14, 27 + 3)$$
$$= 40$$
$$ES_7 = ES_6 + t_{67} = 40 + 14 = 54.$$

Backward pass calculation In this we calculate the latest finish and latest start time LF_i, given by $LF_i = \underset{i}{\text{Min}}\ (LF_j - t_{ij})$ where, LF_j is the latest finish time for the event j

$$LF_7 = 54$$
$$LF_6 = LF_7 - t_{67} = 54 - 14 = 40$$

$$LF_5 = LS_6 - t_{56} = 40 - 3 = 37$$
$$LF_4 = \text{Min } (LS_5 - t_{45}, LS_6 - t_{46})$$
$$= \text{Min } (37 - 1, 40 - 14) = 26$$
$$LF_3 = \text{Min } (LF_4 - t_{34}, LF_6 - t_{36})$$
$$= \text{Min } (26 - 8, 40 - 12) = 18$$
$$LF_2 = \text{Min } (LF_5 - t_{25}, LF_3 - t_{23})$$
$$= \text{Min } (37 - 5, 18 - 3) = 15$$
$$LF_1 = \text{Min } (LF_3 - t_{13}, LF_2 - t_{12})$$
$$= \text{Min } (18 - 15, 15 - 15) = 0$$

The following table gives the calculations for critical path and total float.

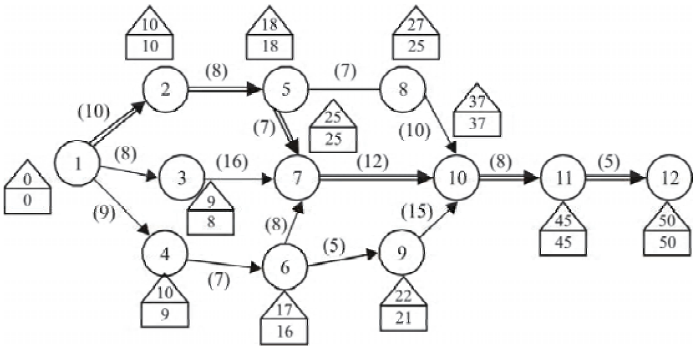
Activity	*Normal time*	*Earliest*		*Latest*		*Total float*
		Start	*Finish*	*Start*	*Finish*	$LF_j - ES_j$ *or* $LF_i - ES_i$
		ES_i	ES_j	LF_i	LF_j	
1 – 2	15	0	15	0	15	0
1 – 3	15	0	15	3	18	3
2 – 3	3	15	18	15	18	0
2 – 5	5	15	20	32	37	17
3 – 4	8	18	26	18	26	0
3 – 6	12	18	30	28	40	10
4 – 5	1	26	27	36	37	10
4 – 6	14	26	40	26	40	0
5 – 6	3	27	30	37	40	10
6 – 7	14	40	54	40	54	0

From the above table, we observe that the activities 1—2, 2—3, 3—4, 4—6, 6—7 are the critical activities and the critical path is given by, 1—2—3—4—6—7.
The total time taken for project completion is 54 days.

Example 15.8 The following table shows the jobs of a project with their duration in days. Draw the network and determine the critical path. Also calculate all the floats.

Jobs	*1–2*	*1–3*	*1–4*	*2–5*	*3–7*	*4–6*	*5–7*	*5–8*	*6–7*	*6–9*	*7–10*	*8–10*	*9–10*	*10–11*	*11–12*
Duration	10	8	9	8	16	7	7	7	8	5	12	10	15	8	5

Solution First we construct the network as shown below:



Forward pass calculation In this we calculate the earliest start and the earliest finish time for the activity and the earliest time for each event.

$$
\begin{aligned}
ES_1 &= 0 \\
ES_2 &= ES_1 + t_{12} = 0 + 10 = 10 \\
ES_3 &= ES_1 + t_{13} = 0 + 8 = 8 \\
E_4 &= E_1 + t_{14} \\
&= 0 + 9 = 9 \\
E_5 &= E_2 + t_{25} = 10 + 8 = 18 \\
E_6 &= E_4 + t_{46} = 9 + 7 = 16 \\
E_7 &= \text{Max}\,(E_3 + t_{37}, E_5 + t_{57}, E_6 + t_{67}) \\
&= \text{Max}\,(8 + 16, 16 + 8, 18 + 7) = 25 \\
E_8 &= E_5 + t_{58} = 18 + 7 = 25 \\
E_9 &= E_6 + t_{69} = 16 + 5 = 21 \\
E_{10} &= \text{Max}\,(E_7 + t_{7,\,10}, E_8 + t_{8,\,10}, E_9 + t_{9,\,10} \\
&= \text{Max}\,(25 + 12, 25 + 10, 21 + 15) = 37 \\
E_{11} &= E_{10} + t_{10,\,11} = 37 + 8 = 45 \\
E_{12} &= E_{11} + t_{11,\,12} = 45 + 5 = 50
\end{aligned}
$$

Backward pass calculation In this we calculate the latest finish and the latest start time LF_i, given by,

$$
\begin{aligned}
LF_i &= \underset{i}{\text{Min}}\,(LF_j - t_{ij}) \\
L_{12} &= E_{12} = 50 \text{ (the target completion time)} \\
L_{11} &= L_{12} - t_{11,\,12} = 50 - 5 = 45 \\
L_{10} &= L_{11} - t_{10,\,11} = 45 - 8 = 37 \\
L_9 &= L_{10} - t_{9.10} = 37 - 15 = 22 \\
L_8 &= L_{10} - t_{8.10} = 37 - 10 = 27 \\
L_7 &= L_{10} - t_{7.10} = 37 - 12 = 25 \\
L_6 &= \text{Min}\,(L_9 - t_{69}, L_7 - t_7 - t_{67}) \\
&= \text{Min}\,(22 - 5, 25 - 8) = 17 \\
L_5 &= \text{Min}\,(L_8 - t_{58}, L_7 - t_{57}) \\
&= \text{Min}\,(27 - 7, 25 - 7) = 18
\end{aligned}
$$

$$L_4 = L_6 - t_{46} = 17 - 7 = 10$$
$$L_3 = L_7 - t_{37} = 25 - 16 = 9$$
$$L_2 = L_5 - t_{25} = 18 - 10 = 10$$
$$L_1 = \text{Min}\,(L_4 - t_{14}, L_3 - t_{13}, L_2 - t_{12})$$
$$= \text{Min}\,(10 - 9, 9 - 8, 10 - 10) = 0.$$

Computations of the critical path and all the floats are given in the following table:

Activity	*Normal time*	*Earliest*		*Latest*		*Floats*		
		Start	*Finish*	*Start*	*Finish*	*TF*	*FF*	*IF*
1–2	10	0	10	0	10	0	0	0
1–3	8	0	8	1	9	1	0	0
1–4	9	0	9	1	10	1	0	0
2–5	8	10	18	10	18	0	0	0
3–7	16	8	24	9	25	1	1	0
4–6	7	9	16	10	17	1	0	–1= 0
5–7	7	18	25	18	25	0	0	0
5–8	7	18	25	20	27	2	0	0
6–7	8	16	24	17	25	1	1	0
6–9	5	16	21	17	22	1	0	–1= 0
7–10	12	25	37	25	37	0	0	0
8–10	10	25	35	27	37	2	2	0
9–10	15	21	36	22	37	1	1	0
10–11	8	37	45	37	45	0	0	0
11–12	5	45	50	45	50	0	0	0

TF = Total float = $LS - ES$ or $LF - EF$

FF = Free float = TF – Head event slack = $TF - (L_j - E_j)$

IF = Independent float = FF – Tail event slack = $FF - (L_i - E_i)$

From the above calculation, we observe that the activity 1–2, 2–5, 5–7, 7–10, 10–11, 11–12 are the critical activities as their total float is 0. Hence, we have the critical path 1–2–5–7–10–11–12, with the total project duration as 50 days.

Example 15.9 A project consists of a series of tasks labelled A, B... H, I with the following constraints, $A < D, E$; $B, D < F$; $C < G$; $B < H$; $F, G < I$; $W < X, Y$ means X and Y cannot start until W is completed. You are required to construct a network using this notation. Also find the minimum time of completion of the project when the time of completion of each task is given as follows:

Task	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*
Time (days)	23	8	20	16	24	18	19	4	10