OPERATIONS RESEARCH

Chapter 7
Network Analysis

Prof. Bibhas C. Giri

Department of Mathematics

Jadavpur University

Kolkata, India

Email: bcgiri.jumath@gmail.com

MODULE - 1: Basic Components of Network and Critical Path Method (CPM)

1.0 Introduction

A project involves a large number of interrelated activities (or tasks) that must be completed on or before a specified sequence or order with specified quality and minimum cost of using resources such as personnel, money, materials, facilities and/or space. Example of projects include, construction of bridge, highway, power plant, repair or maintenance of an oil refinery or an airplane; design, development and marketing of a new product, research and development work, etc. Since project involves large number of interrelated activities, therefore, it is necessary to prepare a plan for scheduling and controlling these activities (or tasks). This approach will help in identifying bottlenecks and even discovering alternate work plan for the project.

Network analysis, network planning and network scheduling techniques are used for planning, scheduling and controlling large and complex projects. These techniques are based on the representation of the project as a network of activities. A network is a graphical presentation of arrows and nodes for showing the logical sequence of various activities to be performed to achieve project objectives. In this chapter, we shall discuss two of these well known techniques - PERT (Program Evaluation and Review Technique) and CPM (Critical Path Method).

PERT was developed in 1956-58 by a research team to help in the planning and scheduling of the US Navy's Polaris nuclear submarine missile project involving thousand of activities. The objective of the team was to efficiently plan and develop the Polaris missile system. This technique was proved to be useful for projects that have an element of uncertainty in the estimation of activity duration, as in the case with new types of projects which have been taken up before.

CPM was developed by E.I. DuPont company along with Remington Rand Corporation almost at the same time 1956-58. The objective of the company was to develop a technique to monitor the maintenance of its chemical plants. This technique has proved to be useful for developing time-cost trade-off for projects that involve activities of repetitive nature.

1.1 Basic Components of Network

A network is a graphic representation of a project's operations and is composed of activities and events that must be completed to reach the end objective of a project, showing the planning sequence of their accomplishments, their dependence and interrelationships. The basic components of a network are as follows:

1. **Activity:** An activity in the network diagram represent project operations (or task) to be conducted. Each activity except dummy activity requires resources and takes a certain amount of time for completion. An arrow is commonly used to represent an activity with its head indicating the direction of progress in the project. Activities are identified by the numbers of their starting (tail or initial)

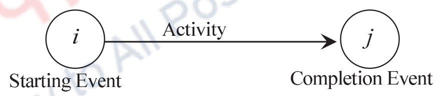


Fig. 1.1: Activity

event and ending (head, or terminal) event. For example, an arrow (i, j) between two events; the tail event i represents the start of the activity and the head event j represents the completion of the activity as shown in Fig. 1.1. The activities can be further classified into the following four categories:

(i) **Predecessor Activity** - An activity which must be completed before the start of one or more other activities is known as predecessor activity.

(ii) **Successor Activity** - An activity which starts immediately after the completion of one or more of other activities is known as successor activity.

(iii) **Concurrent Activity** - An activity which can be accomplished concurrently is known as concurrent activity. 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 the other activities.

(iv) **Dummy Activity** - An activity which does not consume either any resource and/or time is known as dummy activity.

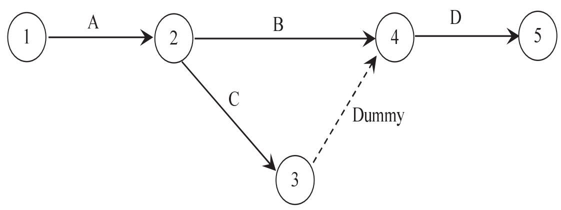


Fig. 1.2: Dummy activity

It can be noted that the dummy activity is inserted in the network to clarify the activity pattern in the following two situations:

(a) to make activities with common starting and finishing points distinguishable, and

(b) to identify and maintain the proper precedence relationship between activities that are not connected by events.

2. **Node:** The specific point in time at which an activity begins or ends is called a node. The starting and end points of an activity are thus described by two nodes, usually known as the *tail node* and *head node*, respectively. A node is generally represented by a circle, rectangle, hexagon or other geometric shapes. Numbers within the geometric shapes distinguish an activity from another one.

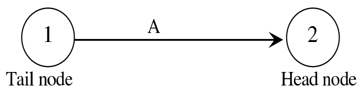


Fig. 1.3: Node

3. **Event:** An event represent the start (beginning) or completion (end) of some activity and as such it consumes no time. It has no time duration and does not consume any resource. An event is nothing but a node and is generally represented on the network by a circle, rectangle, hexagon or some other geometric shape.

The events can be further classified into following three categories:

(i) **Merge event**- When more than one activity comes and joins an event, such event is known as merge event.

(ii) **Burst event-** When more than one activity leaves an event, such event is known as a burst event.

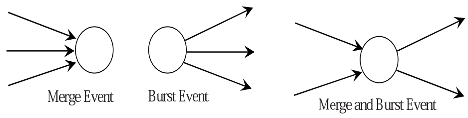


Fig. 1.4: Merge and burst event

(iii) **Merge and burst event-** An event may be merge and burst at the same time.

4. **Activity-on-Node (AON) network:** In this network, nodes represent the activities and the arrows indicate sequencing connection between different activities. There is no need of dummy activities to draw this network diagram.

5. **Activity-on-Arrow (AOA) network:** In this network, the activity is shown on the arrow and the nodes at the end of the arrow state the start and finish of the activity. Nodes are numbered sequentially from left to right. A node or event does not consume time or resource.

6. **Sequencing:** The first pre-requisite in the development of a network is to maintain the precedence relationship. In order to draw a network, one must know the answers of the following questions:

   (a) What job or jobs precede it?

   (b) What job or jobs could run concurrently?

   (c) What job or jobs follow it?

   (d) What controls the start and finish of a job or jobs?

1.2 Fulkerson's Rule for Numbering Events

1. Number the starting event by 1.

2. Delete all arrows emerging from all numbered events. This will create at least one new starting event out of the preceding ones.

3. Number all the new events by 2,3, and so on.

4. Repeat steps 2 and 3 until the final event is numbered.

1.3 Rules for Drawing Network Diagram

For the construction of a network, generally, the following rules are followed :

1. Each activity is represented by one and only one arrow.
2. Each activity must be identified by its starting and end node which implies that
   - (i) two activities should not be identified by the same completion event, and
   - (ii) activities must be represented either by their symbols or by the corresponding ordered pair of starting-completion events.
3. Nodes are numbered to identify an activity uniquely. Tail node (starting point) should be lower than the head node (end point) of an activity.
4. Between any pair of nodes, there should be one and only one activity; however, more than one activity can start from and terminate to a node.
5. Arrows should be kept straight and not curved or bent.
6. The logical sequence (or inter-relationship) between activities must follow the following rules :
   - (i) An event cannot occur until all the incoming activities into it have been completed.
   - (ii) Dummy activities should only be introduced if absolutely necessary.

1.4 Common Errors in Drawing Networks

Three types of errors are commonly observed in drawing network diagrams.
1. Dangling - If an activity is disconnected before completion of all activities in a network diagram then such an error is called dangling error. As shown in figure 1.5, activities (4-5) 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) - Endless loop in a network diagram is known as error of looping as shown in figure 1.6.
3. Redundancy - If a dummy activity is inserted in a network unnecessarily then that error is called redundancy error as shown in figure 1.7.

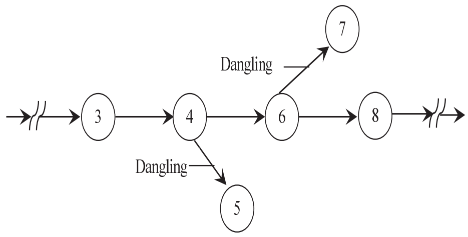


Fig. 1.5: Dangling error

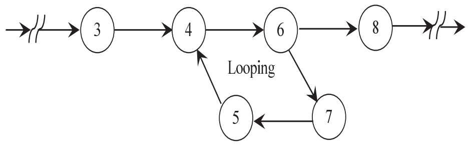


Fig. 1.6: Looping error

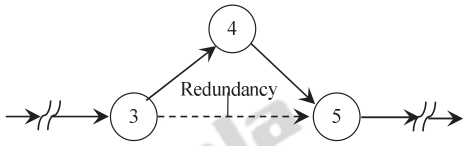


Fig. 1.7: Redundancy

1.5 Critical Path Method (CPM)

The purpose of analysis is to find the critical path, i.e., the sequence of activities with the longest duration, and to find the float associated with each non-critical activity. This helps in checking actual progress against the scheduled duration of the project. To achieve this objective, we carry out the special computations that produce the following information:

(a) Total duration needed for the completion of the project.

(b) Categorization of the activities of the project as being critical or non-critical.

An activity in a network diagram is said to be critical, it the delay in its start will further delay the project completion time. A non-critical activity allows some scheduling slack, so that the start time of the activity may be advanced or delayed within limits without affecting the completion date of the entire project.

The following notations are used for the purpose of calculating various times of events and activities :

E_i = earliest occurrence time of event i. It is the earliest time at which the event i can occur without affecting the total project time

L_i = latest occurrence time of event i. It is the latest time at which event i can occur without affecting the total project time

ES_{ij} = earliest start time for activity (i, j). It is the earliest time at which the activity (i, j) can start without affecting the total project time.

LS_{ij} = latest start time for activity (i, j). It is the latest possible time by which activity (i, j) must start without affecting the total project time.

EF_{ij} = earliest finish time for activity (i, j). It is the earliest possible time at which the activity (i, j) can be finished without affecting the total project time.

LF_{ij} = latest finish time for activity (i, j). It is the latest time by which the activity (i, j) must be completed without delaying the project completion.

t_{ij} = duration of activity (i, j).

The critical path calculations are done in the following two ways :
(a) Forward pass calculations (b) Backward pass calculations.

1.5.1 Forward pass calculations

We start from the initial node 1 with starting time of the project as zero. Proceed through the network visiting nodes in an increasing order of node number and end at the final node of the network. At each node, we calculate earliest start and finish times for each activity by considering E_i as the earliest occurrence of node i. The method may be summarized as follows:

Step 1. Set $E_1 = 0$; $i = 1$ (initial node)

Step 2. Set the earliest start time for each activity that begins at node i as $ES_{ij} = E_i$; for all activities (i, j) that start at node i.

Step 3. Compute the earliest finish time of each activity that begins at node i by adding the earliest start time of the activity to the duration of the activity. Thus $EF_{ij} = ES_{ij} + t_{ij} = E_i + t_{ij}$.

Step 4. Move on to next node, say node j $(j > i)$ and compute the earliest occurrence for node j, using $E_j = \max_i\{EF_{ij}\} = \max_i\{E_i + t_{ij}\}$, for all intermediate predecessor activities.

Step 5. If $j = n$ (final node), then the earliest finish time for the project is given by $E_n = \max\{EF_{ij}\} = \max\{E_{n-1} + t_{ij}\}$.

1.5.2 Backward pass calculations

We start from the final (last) node n of the network, proceed through the network visiting nodes in the decreasing order of node numbers and end at the initial node 1. At each node, we calculate the least finish and start times for each activity by considering L_j as the latest occurrence of node j. The method may be summerized below :

Step 1. $L_n = E_n$ for $j = n$.

Step 2. Set the latest finish time of each activity that ends at node j as $LF_{ij} = L_j$.

Step 3. Compute the latest occurrence times of all activities ending at j by subtracting the duration of each activity from the latest finish time of the activity. Thus $LS_{ij} = LF_{ij} - t_{ij} = L_j - t_{ij}$.

Step 4. Proceed backward to the node in the sequence, that decrease j by 1. Also compute the latest occurrence time of node i $(i < j)$ using $L_i = \min_j\{LS_{ij}\} = \min_j\{L_j - t_{ij}\}$ for all immediate successor activities.

Step 5. If $j = 1$ (initial node), then $L_1 = \min\{LS_{ij}\} = \min\{L_2 - t_{ij}\}$. Based on the above calculations, an activity (i, j) will be critical if it satisfies the following conditions:

(i) $E_i = L_i$ and $E_j = L_j$

(ii) $E_j - E_i = L_j - L_i = t_{ij}$

An activity that does not satisfy the above conditions is termed as non-critical.

Critical Path: The critical activities of a network that constitute an uninterrupted path which spans the entire network from start to finish is known as critical path.

1.5.3 Float (or slack) of an activity and event

The float of an activity is the amount of time by which it is possible to delay its completion time without affecting the total project completion time.

1. **Event float** - The float of an event is the difference between its latest time (L_i) and its earliest time (E_i). That is, $Event\ float = L_i - E_i$.

   It is a measure of how much later than expected a particular event could occur without delaying the completion of the entire project.

2. **Activity float** - It is the float in the activity time estimates. There are mainly three types of activity floats as discussed below :

   (i) ***Total float*** - The total float may be defined as the amount of time by which an activity can be delayed without delay in the project completion date. In other words, it refers to the amount of the free time associated with an activity which can be used before, during or after the performance of this activity. Total float is the positive difference between the earliest finish time and the latest finish time, or the positive difference between the earliest start time and the latest start time of an activity depending upon which way it is defined.

   Thus, for each activity (i, j), the total float are computed as follows :

$$\begin{aligned} \text{Total float } (TF_{ij}) &= L_j - (E_i + t_{ij}) = LF_{ij} - EF_{ij} \\ &= (L_j - t_{ij}) - E_i = LS_{ij} - ES_{ij} \end{aligned}$$

   (ii) ***Free float*** - Free float is that portion of the total float within which an activity can be manipulated without affecting the float of subsequent activities. It is computed for an activity by subtracting the head event slack from its total float. The free float indicates the value by which an activity in question can be delayed without causing any delay in its immediate successor activities.

   Free float values for each activity (i, j) are computed as follows:

$$\begin{aligned} \text{Free float } (FF_{ij}) &= (E_j - E_i) - t_{ij} \\ &= E_j - (E_1 + t_{ij}) = \min\{ES_{ij}\} - EF_{ij}, \quad (i < j). \end{aligned}$$

   (iii) ***Independent float*** - If the float neither affects the predecessor nor the successor activities, then it is called an independent float and is given by

$$\begin{aligned} \text{Independent float } (IF_{ij}) &= (ES_{ij} - LS_{ij}) - t_{ij} \\ &= (E_j - L_i) - t_{ij}. \end{aligned}$$

   (iv) ***Interfering float*** - Utilization of the float of an activity can affect the float of subsequent activities in the network. Thus, interfering float can be defined as that part of the total float which causes a reduction in the float of the successor activities. In other words, it can be defined as the difference between the latest finish time of the activity under consideration and the earliest start time of the following activity, or zero, whichever is larger. Thus interfering float refers to that portion of the activity float which cannot be consumed without affecting adversely the float of the subsequent activity or activities.

Example 1.1: Consider the following network where nodes have been numbered according to the Fulkerson's rule. Numbers along various activities represent the normal time t_{ij} required to finish that activity. Find the time of completion of the project and the critical jobs.

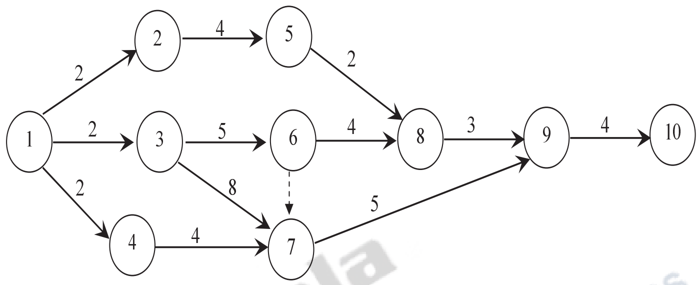


Fig. 1.8:

Solution: Forward pass computations - We have $E_1 = 0$. The earliest occurrence time E_j for event j is given by $E_j = \max_i\{E_i + t_{ij}\}$ where i the collection of all events that precede event j. Therefore,

$$E_2 = \max_i\{E_i + t_{i2}\} = E_1 + t_{12} = 0 + 2 = 2$$

Similarly,

$$\begin{aligned}
E_3 &= E_1 + t_{13} = 0 + 2 = 2 \\
E_4 &= E_1 + t_{14} = 0 + 2 = 2 \\
E_5 &= E_2 + t_{25} = 2 + 4 = 6 \\
E_6 &= E_3 + t_{36} = 2 + 5 = 7 \\
E_7 &= \max\begin{cases} E_3 + t_{37} &= 2 + 8 = 10 \\ E_4 + t_{47} &= 2 + 4 = 6 \\ E_6 + t_{67} &= 7 + 0 = 7 \end{cases} \\
&= 10
\end{aligned}$$

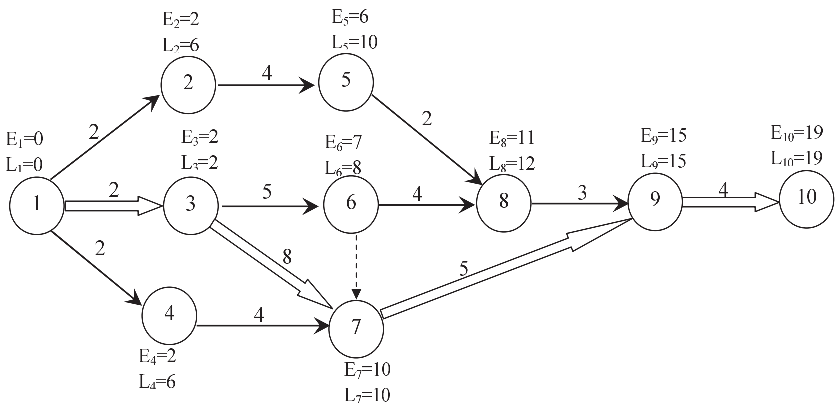
$$
\begin{aligned}
E_8 &= \max\begin{cases} E_5 + t_{58} = 6+2=8 \\ E_6 + t_{68} = 7+4=11 \end{cases} \\
&= 11 \\
E_9 &= \max\begin{cases} E_8 + t_{89} = 11+3=14 \\ E_6 + t_{68} = 10+5=15 \end{cases} \\
&= 15 \\
\text{and } E_{10} &= E_9 + t_{9,10} = 15+4=19
\end{aligned}
$$

Hence the project will be completed in 19 days.

Backward pass computations - We set $L_{10} = E_{10} = 19, \quad j = N = 10$. To calculate the latest occurrence time of event $i(i < j)$, we use the formula $L_i = \min_j\{L_j - t_{ij}\}$. Therefore,

$$
L_9 = \min_j\{L_{10} - t_{9,10}\} = 19-4=15
$$

$$
\begin{aligned}
L_8 &= L_9 - t_{89} = 15-3=12 \\
L_7 &= L_9 - t_{79} = 15-5=10 \\
L_6 &= \min\begin{cases} 12-4 = 8 \\ 10-0 = 10 \end{cases} \\
&= 8 \\
L_5 &= L_8 - t_{58} = 12-2=10 \\
L_4 &= L_7 - t_{47} = 10-4=6 \\
L_3 &= \min\begin{cases} L_6 - t_{36} = 8-5=3 \\ L_7 - t_{37} = 10-8=2 \end{cases} \\
&= 2 \\
L_2 &= L_5 - t_{25} = 10-4=6 \\
L_1 &= \min\begin{cases} 6-2 = 4 \\ 2-2 = 0 \\ 6-2 = 4 \end{cases} \\
&= 0 \\
\text{and } E_{10} &= E_9 + t_{9\ 10} = 15+4=19
\end{aligned}
$$



In this problem, the path 1 – 3 – 7 – 9 – 10 is the critical path. Along this path, the latest and earliest times of occurrence are the same implying that any activity along this path can not be delayed without affecting the duration of the project time.