Lecture 7

CPM and PERT

(Critical Path Method and Program Evaluation and Review Technique)

6.5 CPM AND PERT

CPM (Critical Path Method) and PERT (Program Evaluation and Review Technique) are network-based methods designed to assist in the planning, scheduling, and control of projects.

A project is defined as a collection of interrelated activities with each activity consuming time and resources. The objective of CPM and PERT is to provide analytic means for scheduling the activities. Figure 6.38 summarizes the steps of the techniques. First, we define the activities of the project, their precedence relationships, and their time requirements. Next, the precedence relationships among the activities are represented by a network. The third step involves specific computations to develop the time schedule for the project. During the actual execution of the project things may not proceed as planned, as some of the activities may be expedited or delayed. When this happens, the schedule must be revised to reflect the realities on the ground. This is the reason for including a feedback loop between the time schedule phase and the network phase, as shown in Figure 6.38.

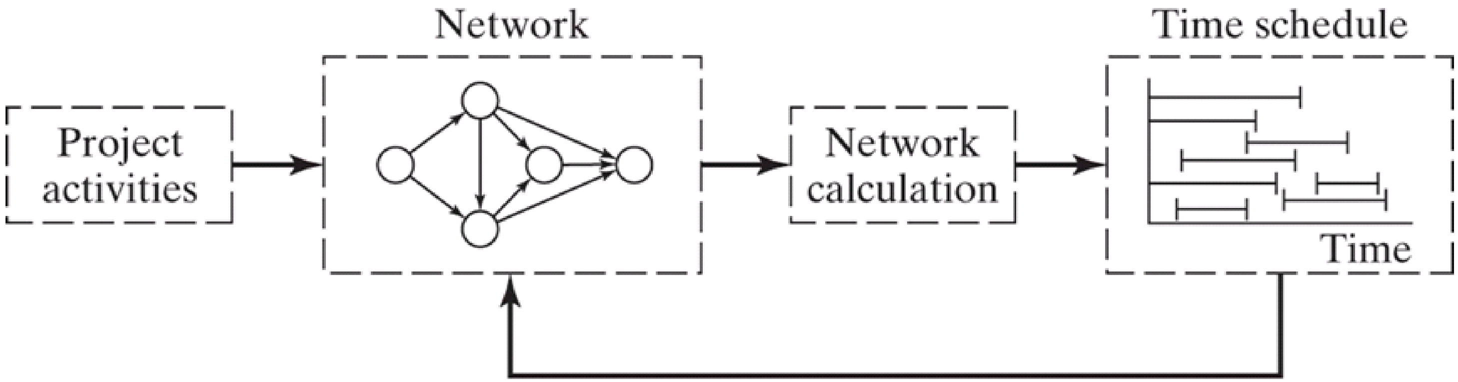


Figure 6.38 Phases for project planning with CPM-PERT

The two techniques, CPM and PERT, which were developed independently, differ in that CPM assumes deterministic activity durations and PERT assumes probabilistic durations. This presentation will start with CPM and then proceed with the details of PERT.

CPM and PERT have been successfully used in many applications, including:

1 Scheduling construction projects such as office buildings, highways, and swimming pools

2 Scheduling the movement of a 400-bed hospital from Portland, Oregon, to a suburban location

3 Developing a countdown and "hold" procedure for the launching of space flights

4 Installing a new computer system

5 Designing and marketing a new product

6 Completing a corporate merger

7 Building a ship

6.5.1 Network Representation

Each activity of the project is represented by an arc pointing in the direction of progress in the project. The nodes of the network establish the precedence relationships among the different activities.

Three rules are available for constructing the network.

Rule 1. *Each activity is represented by one, and only one, arc.*

Rule 2. *Each activity must be identified by two distinct end nodes.*

Figure 6.39 shows how a dummy activity can be used to represent two concurrent activities, *A* and *B*. By definition, a dummy activity, which normally is depicted by a dashed arc, consumes no time or resources. Inserting a dummy activity in one of the four ways shown in Figure 6.39, we maintain the concurrence of *A* and *B*, and provide unique end nodes for the two activities (to satisfy rule 2).

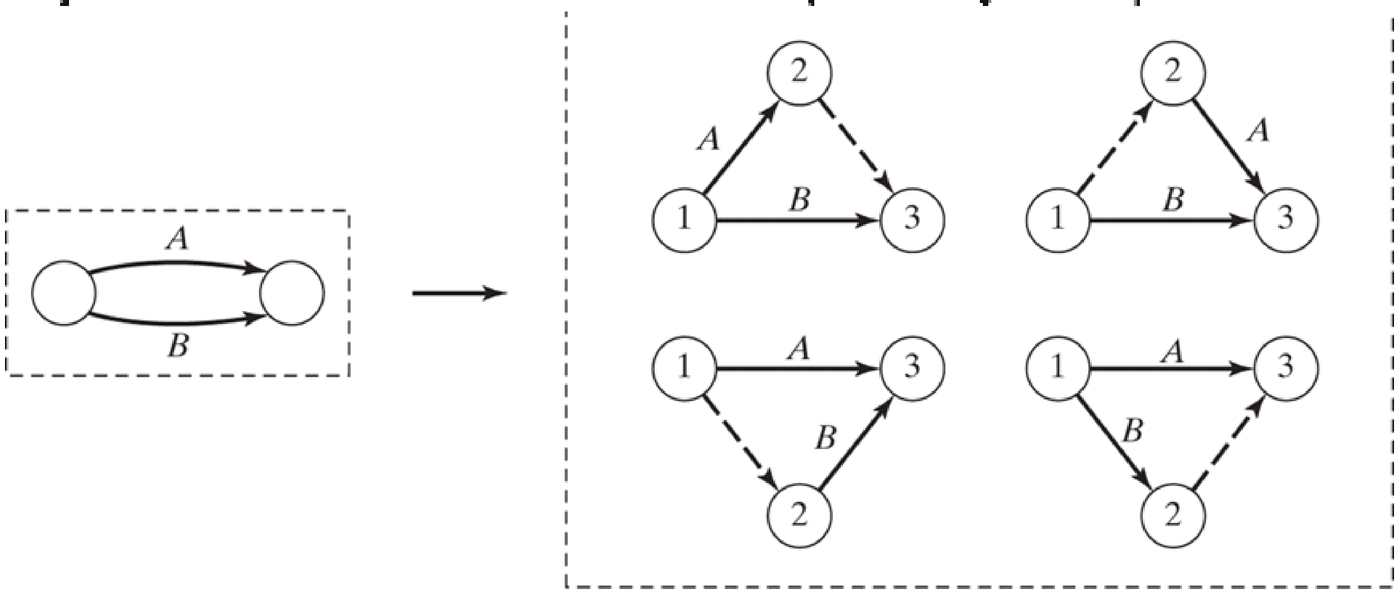


Figure 6.39
Use of dummy activity to produce unique representation of concurrent activities

Rule 3. *To maintain the correct precedence relationships, the following questions must be answered as each activity is added to the network:*

(a) *What activities must immediately precede the current activity?*

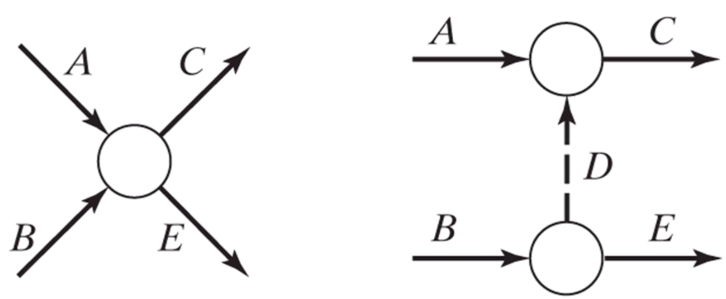
(b) *What activities must follow the current activity?*

(c) *What activities must occur concurrently with the current activity?*

The answers to these questions may require the use of dummy activities to ensure correct precedences among the activities. For example, consider the following segment of a project:

1. Activity C starts immediately after A and B have been completed.
2. Activity E starts only after B has been completed.

Part (a) of Figure 6.40 shows the incorrect representation of the precedence relationship because it requires both A and B to be completed before E can start. In part (b), the use of a dummy activity rectifies the situation.



(a) (b)

Figure 6.40
Use of dummy activity
to ensure
correct precedence
relationship

Example 6.5-1

A publisher has a contract with an author to publish a textbook. The (simplified) activities associated with the production of the textbook are given below. The author is required to submit to the publisher a hard copy and a computer file of the manuscript. Develop the associated network for the project.

Activity	Predecessor(s)	Duration (weeks)
A: Manuscript proofreading by editor	—	3
B: Sample pages preparation	—	2
C: Book cover design	—	4
D: Artwork preparation	—	3
E: Author's approval of edited manuscript and sample pages	*A*, *B*	2
F: Book formatting	*E*	4
G: Author's review of formatted pages	*F*	2
H: Author's review of artwork	*D*	1
I: Production of printing plates	*G*, *H*	2
J: Book production and binding	*C*, *I*	4

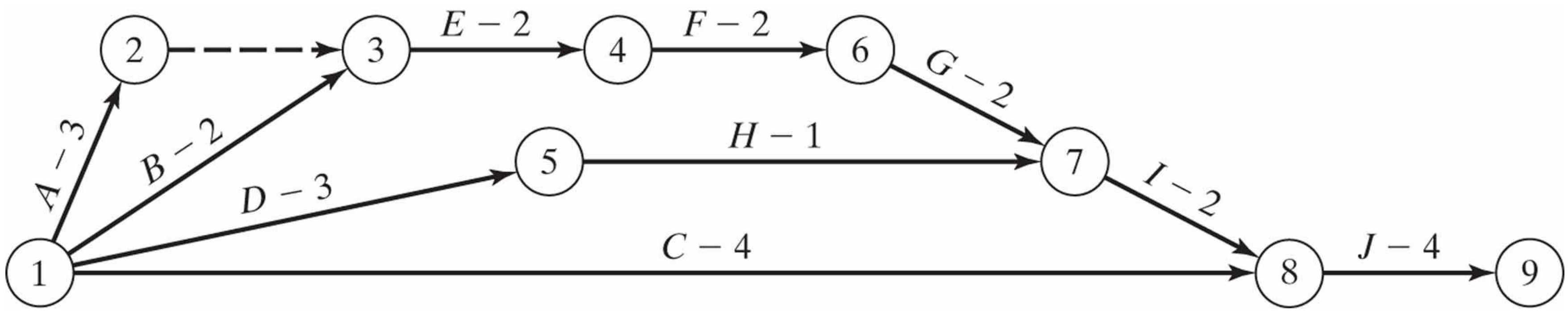


Figure 6.41 Project network for Example 6.5-1

Figure 6.41 provides the network describing the precedence relationships among the different activities. Dummy activity (2, 3) produces unique end nodes for concurrent activities A and B. It is convenient to number the nodes in ascending order in the direction of progress in the project.

PROBLEM SET 6.5A

1. Construct the project network comprised of activities *A* to *L* with the following precedence relationships:
   - **(a)** A, B, and *C*, the first activities of the project, can be executed concurrently.
   - **(b)** A and *B* precede *D*.
   - **(c)** B precedes *E*, *F*, and *H*.
   - **(d)** F and *C* precede *G*.
   - **(e)** E and *H* precede *I* and *J*.
   - **(f)** C, D, F, and *J* precede *K*.
   - **(g)** K precedes *L*.
   - **(h)** I, G, and *L* are the terminal activities of the project.

*3. The footings of a building can be completed in four consecutive sections. The activities for each section include (1) digging, (2) placing steel, and (3) pouring concrete. The digging of one section cannot start until that of the preceding section has been completed. The same restriction applies to pouring concrete. Develop the project network.

5. An opinion survey involves designing and printing questionnaires, hiring and training personnel, selecting participants, mailing questionnaires, and analyzing the data. Construct the project network, stating all assumptions.

6. The activities in the following table describe the construction of a new house. Construct the associated project network.

	Activity	Predecessor(s)	Duration (days)
A:	Clear site	—	1
B:	Bring utilities to site	—	2
C:	Excavate	*A*	1
D:	Pour foundation	*C*	2
E:	Outside plumbing	*B*, *C*	6
F:	Frame house	*D*	10
G:	Do electric wiring	*F*	3
H:	Lay floor	*G*	1
I:	Lay roof	*F*	1
J:	Inside plumbing	*E*, *H*	5
K:	Shingling	*I*	2
L:	Outside sheathing insulation	*F*, *J*	1
M:	Install windows and outside doors	*F*	2
N:	Do brick work	*L*, *M*	4
O:	Insulate walls and ceiling	*G*, *J*	2
P:	Cover walls and ceiling	*O*	2
Q:	Insulate roof	*I*, *P*	1
R:	Finish interior	*P*	7
S:	Finish exterior	*I*, *N*	7
T:	Landscape	*S*	3

6.5.2 Critical Path (CPM) Computations

The end result in CPM is the construction of the time schedule for the project (see Figure 6.38). To achieve this objective conveniently, we carry out special computations that produce the following information:

1. Total duration needed to complete the project.
2. Classification of the activities of the project as *critical* and *noncritical*.

An activity is said to be **critical** if there is no "leeway" in determining its start and finish times. A **noncritical** activity allows some scheduling slack, so that the start time of the activity can be advanced or delayed within limits without affecting the completion date of the entire project.

To carry out the necessary computations, we define an **event** as a point in time at which activities are terminated and others are started. In terms of the network, an event corresponds to a node. Define

$\square_j$ = Earliest occurrence time of event j (E_i)
Δ_j = Latest occurrence time of event j (L_j)
D_{ij} = Duration of activity (i, j)

The definitions of the *earliest* and *latest* occurrences of event j are specified relative to the start and completion dates of the entire project.

The critical path calculations involve two passes: The **forward pass** determines the *earliest* occurrence times of the events, and the **backward pass** calculates their *latest* occurrence times.

Forward Pass (Earliest Occurrence Times, □). The computations start at node 1 and advance recursively to end node n.

Initial Step. Set $□_1 = 0$ to indicate that the project starts at time 0.

General Step j. Given that nodes $p, q, \ldots,$ and v are linked *directly* to node j by incoming activities $(p, j), (q, j), \ldots,$ and (v, j) and that the earliest occurrence times of events (nodes) $p, q, \ldots,$ and v have already been computed, then the earliest occurrence time of event j is computed as

$$□_j = \max\{□_p + D_{pj}, □_q + D_{qj}, \ldots, □_v + D_{vj}\}$$

The forward pass is complete when $□_n$ at node n has been computed. By definition $□_j$ represents the longest path (duration) to node j.

Backward Pass (Latest Occurrence Times, Δ). Following the completion of the forward pass, the backward pass computations start at node n and end at node 1.

Initial Step. Set $\Delta_n = \square_n$ to indicate that the earliest and latest occurrences of the last node of the project are the same.

General Step *j*. Given that nodes $p, q, \ldots,$ and v are linked *directly* to node j by *outgoing* activities $(j, p), (j, q), \ldots,$ and (j, v) and that the latest occurrence times of nodes $p, q, \ldots,$ and v have already been computed, the latest occurrence time of node j is computed as

$$\Delta_j = \min\{\Delta_p - D_{jp}, \Delta_q - D_{jq}, \ldots, \Delta_v - D_{jv}\}$$

The backward pass is complete when Δ_1 at node 1 is computed. At this point, $\Delta_1 = \square_1\ (= 0)$.

Based on the preceding calculations, an activity (i, j) will be *critical* if it satisfies three conditions.

1. $\Delta_i = \Box_i$
2. $\Delta_j = \Box_j$
3. $\Delta_j - \Delta_i = \Box_j - \Box_i = D_{ij}$

The three conditions state that the earliest and latest occurrence times of end nodes i and j are equal and the duration D_{ij} fits "tightly" in the specified time span. An activity that does not satisfy all three conditions is thus *noncritical*.

By definition, the critical activities of a network must constitute an uninterrupted path that spans the entire network from start to finish.

Example 6.5-2

Determine the critical path for the project network in Figure 6.42 . All the durations are in days.

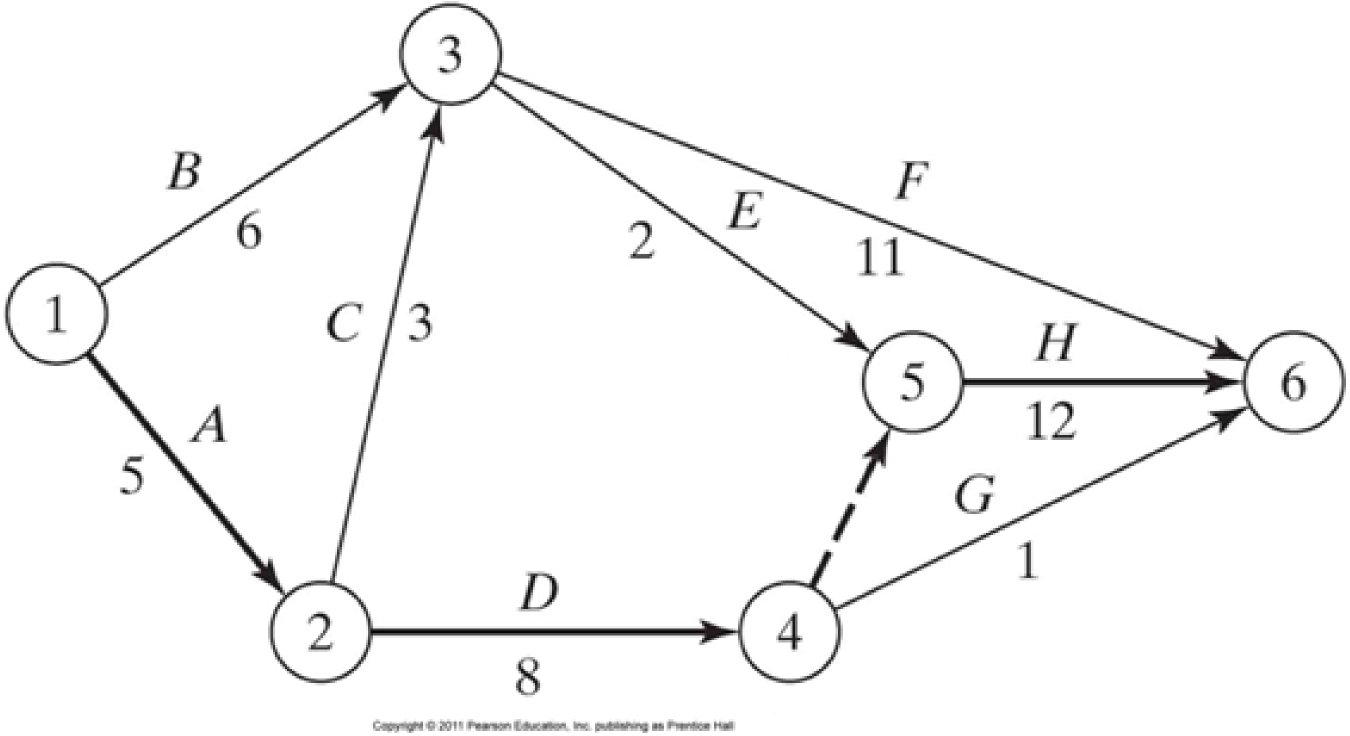


Figure 6.42 Project network for Example 6.5-2

Forward Pass

Node 1. Set $\square_1 = 0$

Node 2. $\square_2 = \square_1 + D_{12} = 0 + 5 = 5$

Node 3. $\square_3 = \max\{\square_1 + D_{13}, \square_2 + D_{23}\} = \max\{0 + 6, 5 + 3\} = 8$

Node 4. $\square_4 = \square_2 + D_{24} = 5 + 8 = 13$

Node 5. $\square_5 = \max\{\square_3 + D_{35}, \square_4 + D_{45}\} = \max\{8 + 2, 13 + 0\} = 13$

Node 6. $\square_6 = \max\{\square_3 + D_{36}, \square_4 + D_{46}, \square_5 + D_{56}\}$
$= \max\{8 + 11, 13 + 1, 13 + 12\} = 25$

The computations show that the project can be completed in 25 days.

Backward Pass

Node 6. Set $\Delta_6 = \square_6 = 25$

Node 5. $\Delta_5 = \Delta_6 - D_{56} = 25 - 12 = 13$

Node 4. $\Delta_4 = \min\{\Delta_6 - D_{46}, \Delta_5 - D_{45}\} = \min\{25 - 1, 13 - 0\} = 13$

Node 3. $\Delta_3 = \min\{\Delta_6 - D_{36}, \Delta_5 - D_{35}\} = \min\{25 - 11, 13 - 2\} = 11$

Node 2. $\Delta_2 = \min\{\Delta_4 - D_{24}, \Delta_3 - D_{23}\} = \min\{13 - 8, 11 - 3\} = 5$

Node 1. $\Delta_1 = \min\{\Delta_3 - D_{13}, \Delta_2 - D_2\} = \min\{11 - 6, 5 - 5\} = 0$

Correct computations will always end with $\Delta_1 = 0$.

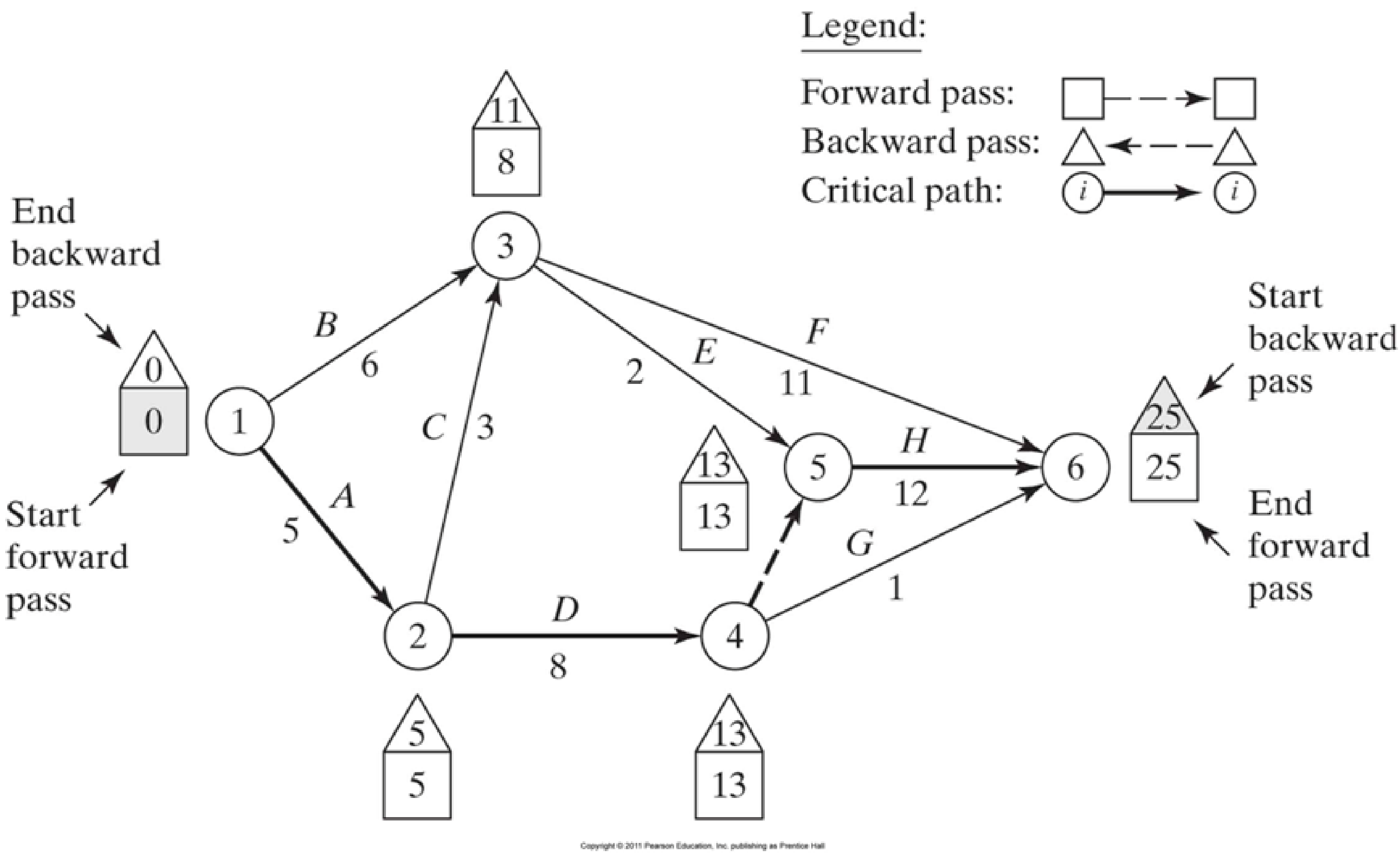


Figure 6.42 Project network for Example 6.5-2

The forward and backward pass computations can be made directly on the network as shown in Figure 6.42. Applying the rules for determining the critical activities, the critical path is $1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6$, which, as should be expected, spans the network from start (node 1) to finish (node 6). The sum of the durations of the critical activities [(1, 2), (2, 4), (4, 5), and (5, 6)] equals the duration of the project (= 25 days). Observe that activity (4, 6) satisfies the first two conditions for a critical activity ($\Delta_4 = \square_4 = 13$ and $\Delta_5 = \square_5 = 25$) but not the third ($\square_6 - \square_4 \neq D_{46}$). Hence, the activity is noncritical.

PROBLEM SET 6.5B

*1. Determine the critical path for the project network in Figure 6.43.

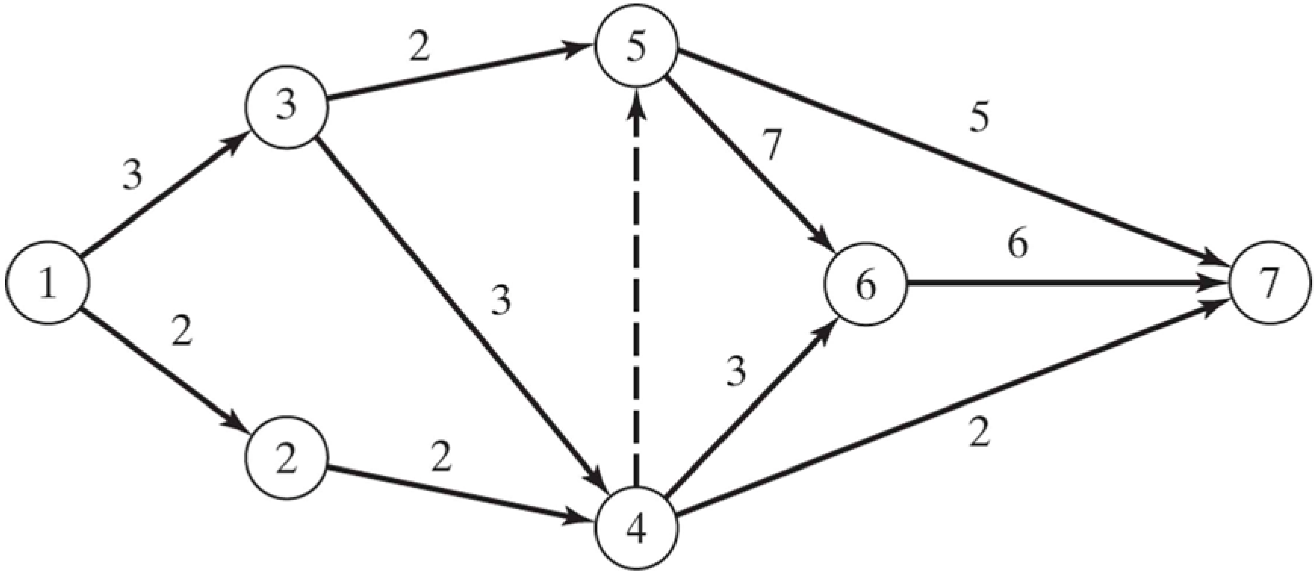


Figure 6.43 Project networks for Problem 1, Set 6.5b

2. Determine the critical path for the project networks in Figure 6.44.

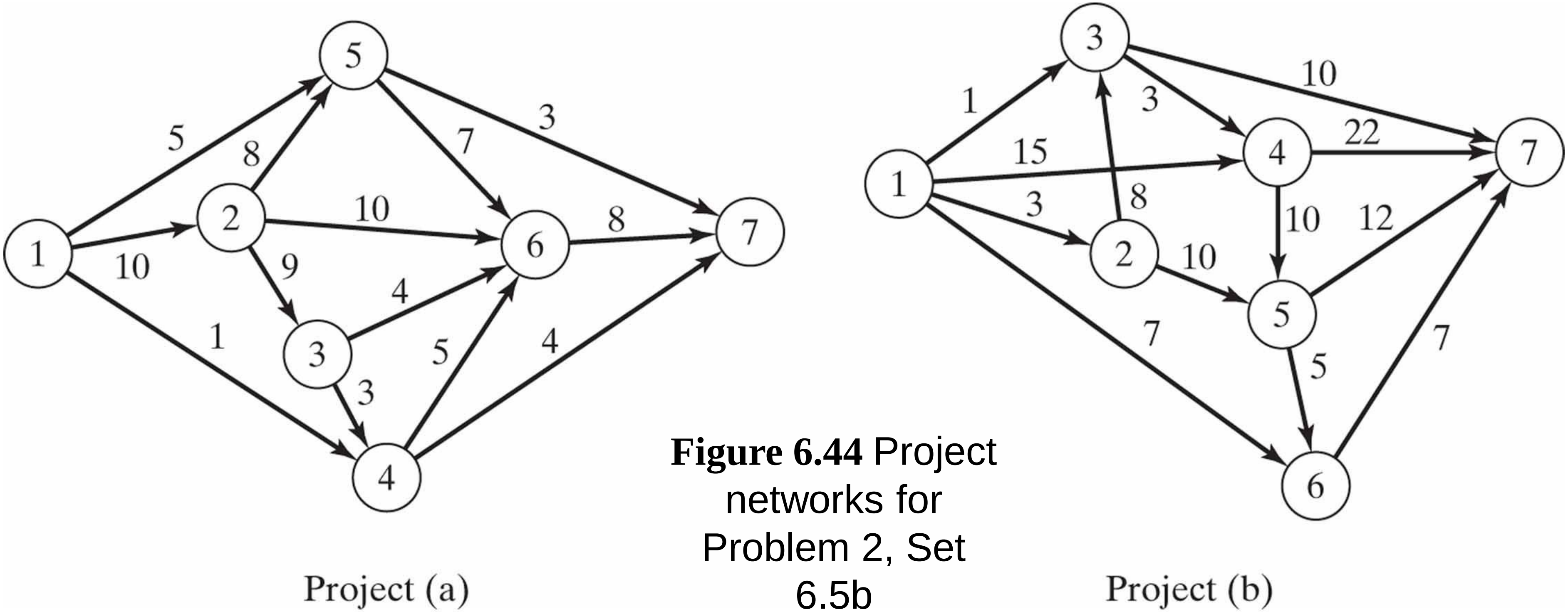


Figure 6.44 Project networks for Problem 2, Set 6.5b

6.5.3 Construction of the Time Schedule

This section shows how the information obtained from the calculations in Section 6.5.2 can be used to develop the time schedule. We recognize that for an activity (i, j), $\square_i$ represents the *earliest start time*, and Δ_j represents the *latest completion time*. This means that the interval $(\square_i, \Delta_j)$ delineates the (maximum) span during which activity (i, j) may be scheduled without delaying the entire project.

Construction of Preliminary Schedule. The method for constructing a preliminary schedule is illustrated by an example.

Example 6.5-3

Determine the time schedule for the project of Example 6.5-2 (Figure 6.42).

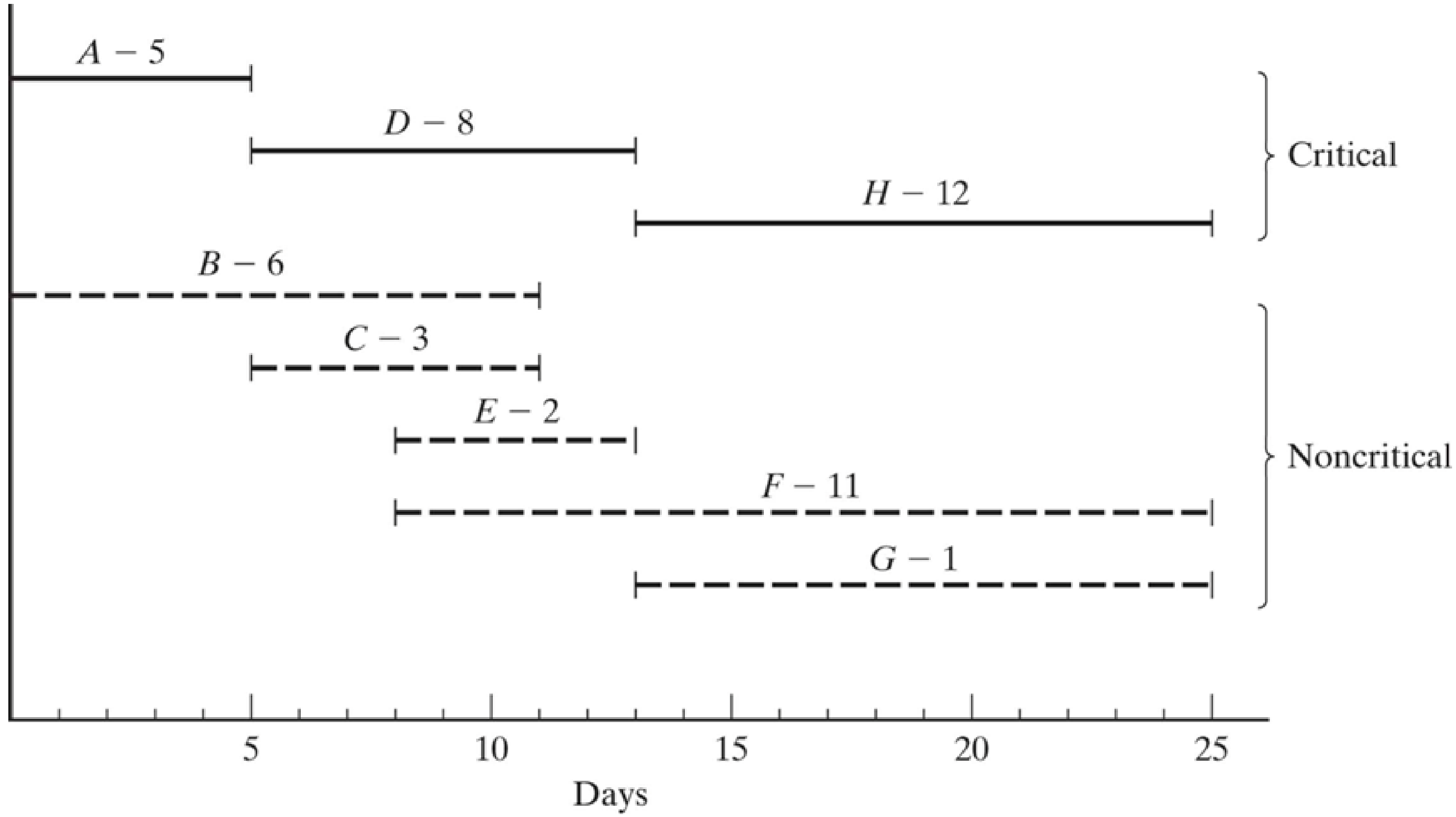


Figure 6.45 Preliminary schedule for the project of Example 6.5-2

We can get a preliminary time schedule for the different activities of the project by delineating their respective time spans as shown in Figure 6.45. Two observations are in order.

1. The critical activities (shown by solid lines) must be stacked one right after the other to ensure that the project is completed within its specified 25-day duration.
2. The noncritical activities (shown by dashed lines) have time spans that are larger than their respective durations, thus allowing slack (or "leeway") in scheduling them within their allotted time intervals.

How should we schedule the noncritical activities within their respective spans? Normally, it is preferable to start each noncritical activity as early as possible. In this manner, slack periods will remain opportunely available at the end of the allotted span where they can be used to absorb unexpected delays in the execution of the activity. It may be necessary, however, to delay the start of a noncritical activity past its earliest start time. For example, in Figure 6.45, suppose that each of the noncritical activities E and F requires the use of a bulldozer, and that only one is available. Scheduling both E and F as early as possible requires two bulldozers between times 8 and 10. We can remove the overlap by starting E at time 8 and pushing the start time of F to somewhere between times 10 and 14.

If all the noncritical activities can be scheduled as early as possible, the resulting schedule automatically is feasible. Otherwise, some precedence relationships may be violated if noncritical activities are delayed past their earliest time. Take for example activities C and E in Figure 6.45. In the project network (Figure 6.42), though C must be completed before E, the spans of C and E in Figure 6.45 allow us to schedule C between times 6 and 9, and E between times 8 and 10, which violates the requirement that C precede E. The need for a "red flag" that automatically reveals schedule conflict is thus evident. Such information is provided by computing the *floats* for the noncritical activities.

Determination of the Floats. Floats are the slack times available within the allotted span of the noncritical activity. The most common are the **total float and the free float.**

Figure 6.46 gives a convenient summary for computing the total float (TF_{ij}) and the free float (FF_{ij}) for an activity (i, j). The total float is the excess of the time span defined from the *earliest* occurrence of event i to the *latest* occurrence of event j over the duration of (i, j)—that is,

$$TF_{ij} = \Delta_j - \square_i - D_{ij}$$

The free float is the excess of the time span defined from the *earliest* occurrence of event i to the *earliest* occurrence of event j over the duration of (i, j)—that is,

$$FF_{ij} = \square_j - \square_i - D_{ij}$$

By definition, $FF_{ij} \leq TF_{ij}$.

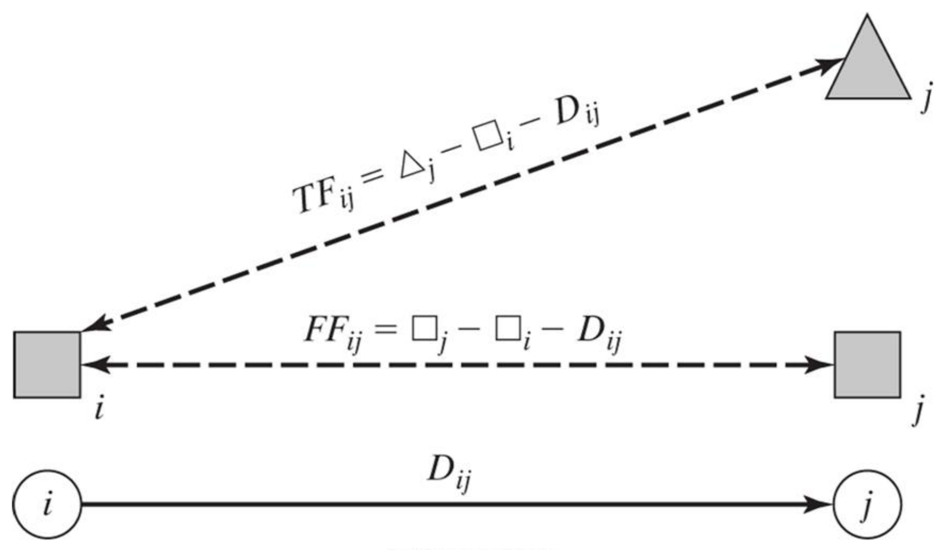


Figure 6.46
Computation of total and free floats

Red-Flagging Rule. *For a noncritical activity (i, j)*

(a) *If $FF_{ij} = TF_{ij}$, then the activity can be scheduled anywhere within its $(\square_i, \Delta_j)$ span without causing schedule conflict.*

(b) *If $FF_{ij} < TF_{ij}$, then the start of the activity can be delayed by at most FF_{ij} relative to its earliest start time $(\square_i)$ without causing schedule conflict. Any delay larger than FF_{ij} (but not more than TF_{ij}) must be coupled with an equal delay relative to $\square_j$ in the start time of all the activities leaving node j.*

The implication of the rule is that a noncritical activity (i, j) will be red-flagged if its $FF_{ij} < TF_{ij}$. This red flag is important only if we decide to delay the start of the activity past its earliest start time, $\square_i$, in which case we must pay attention to the start times of the activities leaving node j to avoid schedule conflicts.

Example 6.5-4

Compute the floats for the noncritical activities of the network in Example 6.5-2, and discuss their use in finalizing a schedule for the project.

The following table summarizes the computations of the total and free floats. It is more convenient to do the calculations directly on the network using the procedure in Figure 6.42.

Noncritical activity	Duration	Total float (TF)	Free float (FF)
$B(1,3)$	6	$11 - 0 - 6 = 5$	$8 - 0 - 6 = 2$
$C(2,3)$	3	$11 - 5 - 3 = 3$	$8 - 5 - 3 = 0$
$E(3,5)$	2	$13 - 8 - 2 = 3$	$13 - 8 - 2 = 3$
$F(3,6)$	11	$25 - 8 - 11 = 6$	$25 - 8 - 11 = 6$
$G(4,6)$	1	$25 - 13 - 1 = 11$	$25 - 13 - 1 = 11$

The computations red-flag activities B and C because their $FF < TF$. The remaining activities (E, F, and G) have $FF = TF$, and hence may be scheduled anywhere between their earliest start and latest completion times.

To investigate the significance of the red-flagged activities, consider activity B. Because its $TF = 5$ days, this activity can start as early as time 0 or as late as time 5 (see Figure 6.45). However, because its $FF = 2$ days, starting B anywhere between time 0 and time 2 will have no effect on the succeeding activities E and F. If, however, activity B must start at time $2 + d$ (≤ 5), then the start times of the immediately succeeding activities E and F must be pushed forward past their earliest start time (= 8) by at least d. In this manner, the precedence relationship between B and its successors E and F is preserved.

Turning to red-flagged activity C, we note that its $FF = 0$. This means that *any* delay in starting C past its earliest start time (= 5) must be coupled with at least an equal delay in the start of its successor activities E and F.

PROBLEM SET 6.5C

*3. For each of the following activities, determine the maximum delay in the starting time relative to its earliest start time that will allow all the immediately succeeding activities to be scheduled anywhere between their earliest and latest completion times.

(a) $TF = 10, FF = 10, D = 4$

(b) $TF = 10, FF = 5, D = 4$

(c) $TF = 10, FF = 0, D = 4$

*5. In the project of Example 6.5-2 (Figure 6.42), assume that the durations of activities B and F are changed from 6 and 11 days to 20 and 25 days, respectively.

(a) Determine the critical path.

(b) Determine the total and free floats for the network, and identify the red-flagged activities.

(c) If activity A is started at time 5, determine the earliest possible start times for activities C, D, E, and G.

(d) If activities F, G, and H require the same equipment, determine the minimum number of units needed of this equipment.

6.5.4 Linear Programming Formulation of CPM

A CPM problem can thought of as the opposite of the shortest-route problem (Section 6.3), in the sense that we are interested in finding the *longest* route of a unit flow entering at the start node and terminating at the finish node. We can thus apply the shortest route LP formulation in Section 6.3.3 to CPM in the following manner. Define

$$x_{ij} = \text{Amount of flow in activity } (i, j), \text{ for all defined } i \text{ and } j$$
$$D_{ij} = \text{Duration of activity } (i, j), \text{ for all defined } i \text{ and } j$$

Thus, the objective function of the linear program becomes

$$\text{Maximize } z = \sum_{\substack{\text{all defined}\\ \text{activities } (i, j)}} D_{ij}x_{ij}$$

(Compare with the shortest route LP formulation in Section 6.3.3 where the objective function is minimized.) For each node, there is one constraint that represents the conservation of flow:

$$\text{Total input flow} = \text{Total output flow}$$

All the variables, x_{ij}, are nonnegative.

Example 6.5-5

The LP formulation of the project of Example 6.5-2 (Figure 6.42) is given below. Note that nodes 1 and 6 are the start and finish nodes, respectively.

	A	B	C	D	E	F	Dummy	G	H	
	x_{12}	x_{13}	x_{23}	x_{24}	x_{35}	x_{36}	x_{45}	x_{46}	x_{56}	
Maximize $z =$	6	6	3	8	2	11	0	1	12	
Node 1	-1	-1								$=-1$
Node 2	1		-1	-1						$=0$
Node 3		1	1		-1	-1				$=0$
Node 4				1			-1	-1		$=0$
Node 5					1		1		-1	$=0$
Node 6						1		1	1	$=1$

The optimum solution is

$$z = 25, x_{12}(A) = 1, x_{24}(D) = 1, x_{45}(\text{Dummy}) = 1, x_{56}(H) = 1, \text{all others} = 0$$

The solution defines the critical path as $A \rightarrow D \rightarrow$ Dummy $\rightarrow H$, and the duration of the project is 25 days. The LP solution is not complete, because it determines the critical path, but does not provide the data needed to construct the CPM chart.

6.5.5 PERT Networks

PERT differs from CPM in that it bases the duration of an activity on three estimates:

1. **Optimistic time,** a, which occurs when execution goes extremely well.
2. **Most likely time,** m, which occurs when execution is done under normal conditions.
3. **Pessimistic time,** b, which occurs when execution goes extremely poorly.

The range (a, b) encloses all possible estimates of the duration of an activity. The estimate m lies somewhere in the range (a, b). Based on the estimates, the average duration time, $\overline{D}$, and variance, v, are approximated as:

$$\overline{D} = \frac{a + 4m + b}{6} \qquad v = \left(\frac{b - a}{6}\right)^2$$

CPM calculations given in Sections 6.5.2 and 6.5.3 may be applied directly, with $\overline{D}$ replacing the single estimate D.

It is possible now to estimate the probability that a node j in the network will occur by a prespecified scheduled time, S_j. Let e_j be the earliest occurrence time of node j. Because the durations of the activities leading from the start node to node j are random variables, e_j also must be a random variable. Assuming that all the activities in the network are statistically independent, we can determine the mean, $E\{e_j\}$, and variance, $\text{var}\{e_j\}$, in the following manner. If there is only one path from the start node to node j, then the mean is the sum of expected durations, $\overline{D}$, for all the activities along this path and the variance is the sum of the variances, v, of the same activities. On the other hand, if more than one path leads to node j, then it is necessary first to determine the statistical distribution of the duration of the longest path. This problem is rather difficult because it is equivalent to determining the distribution of the maximum of two or more random variables. A simplifying assumption thus calls for computing the mean and variance, $E\{e_j\}$ and $\text{var}\{e_j\}$, as those of the path to node j that has the largest sum of *expected* activity durations. If two or more paths have the same mean, the one with the largest variance is selected because it reflects the most uncertainty and, hence, leads to a more conservative estimate of probabilities.

Once the mean and variance of the path to node j, $E\{e_j\}$ and $\text{var}\{e_j\}$, have been computed, the probability that node j will be realized by a preset time S_j is calculated using the following formula:

$$P\{e_j \leq S_j\} = P\left\{\frac{e_j - E\{e_j\}}{\sqrt{\text{var}\{e_j\}}} \leq \frac{S_j - E\{e_j\}}{\sqrt{\text{var}\{e_j\}}}\right\} = P\{z \leq K_j\}$$

where

$$z = \text{Standard normal random variable}$$

$$K_j = \frac{S_j - E\{e_j\}}{\sqrt{\text{var}\{e_j\}}}$$

The standard normal variable z has mean 0 and standard deviation 1 (see Section 12.4.4). Justification for the use of the normal distribution is that e_j is the sum of independent random variables. According to the *central limit theorem* (see Section 12.4.4), e_j is approximately normally distributed.

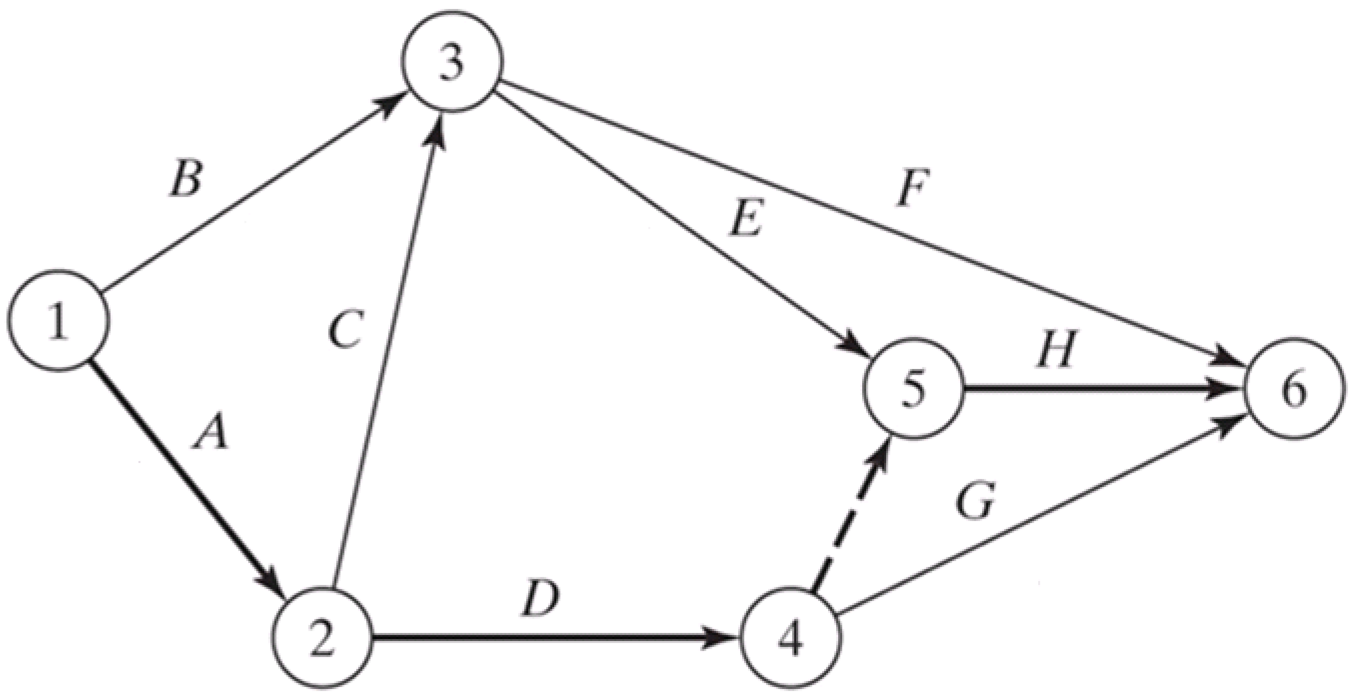
Example 6.5-6

Consider the project of Example 6.5-2. To avoid repeating critical path calculations, the values of a, m, and b in the table below are selected such that $\overline{D_{ij}} = D_{ij}$ for all i and j in Example 6.5-2.

Activity	i–j	(a, m, b)	Activity	i–j	(a, m, b)
A	1–2	(3, 5, 7)	E	3–5	(1, 2, 3)
B	1–3	(4, 6, 8)	F	3–6	(9, 11, 13)
C	2–3	(1, 3, 5)	G	4–6	(1, 1, 1)
D	2–4	(5, 8, 11)	H	5–6	(10, 12, 14)

The mean $\overline{D_{ij}}$ and variance v_{ij} for the different activities are given in the following table. Note that for a dummy activity $(a, m, b) = (0, 0, 0)$, hence its mean and variance also equal zero.

Activity	i–j	$\overline{D}_{ij}$	V_{ij}	Activity	i–j	$\overline{D}_{ij}$	V_{ij}
A	1–2	5	.444	E	3–5	2	.111
B	1–3	6	.444	F	3–6	11	.444
C	2–3	3	.444	G	4–6	1	.000
D	2–4	8	1.000	H	5–6	12	.444


1
2
3
4
5
6
A
B
C
D
E
F
G
H

The next table gives the longest path from node 1 to the different nodes, together with their associated mean and standard deviation.

Node	Longest path based on mean durations	Path mean	Path standard deviation
2	1–2	5.00	0.67
3	1–2–3	8.00	0.94
4	1–2–4	13.00	1.20
5	1–2–4–5	13.00	1.20
6	1–2–4–5–6	25.00	1.37

Finally, the following table computes the probability that each node is realized by time S_j specified by the analyst.

Node j	Longest path	Path mean	Path standard deviation	S_j	K_j	$P\{z \leq K_j\}$
2	1–2	5.00	0.67	5.00	0	.5000
3	1–2–3	8.00	0.94	11.00	3.19	.9993
4	1–2–4	13.00	1.20	12.00	−.83	.2033
5	1–2–4–5	13.00	1.20	14.00	.83	.7967
6	1–2–4–5–6	25.00	1.37	26.00	.73	.7673

PROBLEM SET 6.5E

1. Consider Problem 2, Set 6.5b. The estimates (a, m, b) are listed below. Determine the probabilities that the different nodes of the project will be realized without delay.

Project (a)				Project (b)			
Activity	(a, m, b)	Activity	(a, m, b)	Activity	(a, m, b)	Activity	(a, m, b)
1-2	(5, 6, 8)	3-6	(3, 4, 5)	1-2	(1, 3, 4)	3-7	(12, 13, 14)
1-4	(1, 3, 4)	4-6	(4, 8, 10)	1-3	(5, 7, 8)	4-5	(10, 12, 15)
1-5	(2, 4, 5)	4-7	(5, 6, 8)	1-4	(6, 7, 9)	4-7	(8, 10, 12)
2-3	(4, 5, 6)	5-6	(9, 10, 15)	1-6	(1, 2, 3)	5-6	(7, 8, 11)
2-5	(7, 8, 10)	5-7	(4, 6, 8)	2-3	(3, 4, 5)	5-7	(2, 4, 8)
2-6	(8, 9, 13)	6-7	(3, 4, 5)	2-5	(7, 8, 9)	6-7	(5, 6, 7)
3-4	(5, 9, 19)			3-4	(10, 15, 20)		

Cost analysis in project management
Crashing the project

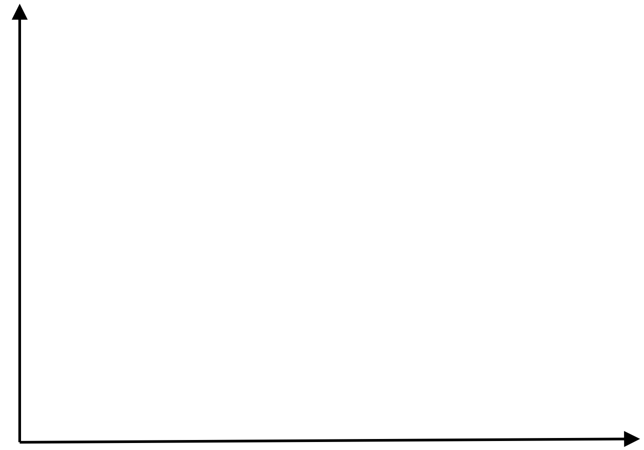
- In many situations, the project manager must complete the project in a time that is less than the length of the critical path.
- So, it is necesseary to determine the allocation of resources that minimizes the cost of meeting the project deadline.
- In practice, Linear programming is commonly used while crashing the Project.
- <u>The parameters:</u>

D_n : Normal completion time

C_n : The cost of completion of activity in normal completion time (D_n)

D_h : Fastest completion time

C_h : The cost of completion of activity in fastest completion time (D_h)



- Suppose that the critical path is determined.
- If we aim to shorten the total duration of project, it is clear that we should select the critical activity or a group of critical activity which has total minimal slope (unit cost).
- The crashing amount is determined by two factors: (1) maximal crashing amount of selected critical activities (2) Free floats of *alternative noncritical paths.*

Alternative noncritical paths

- Starts from the same or backward event with the selected critical activities
- Leads to the critical path(s).

After crashing the project, it is possible to have more than one critical path.

When there doesn't exist any candidate activity to shorten or a critical path is reached its crashing limit, it is not possible to shorten the total project duration, that is crashing process is over.

It should also be noted that a project may have initially more than one critical path.

Example1: Table on below gives the activities of a project, their predecessor(s), durations and costs.

a) Draw the network for the project.
b) Determine the critical path.
c) Construct the Time Schedule.
d) Obtain programs with minimum costs.

Activities	Predecessor(s)	Durations		Costs	
		Normal	Crashed	Normal	Crashed
A	---	4	2	100	160
B	---	5	5	200	200
C	---	7	4	250	310
D	A	7	5	180	210
E	B	6	6	120	120
F	C	8	4	150	250
G	C	9	6	200	320
H	E, F	4	3	160	220
I	E, F	6	5	240	300
J	E, F, D	10	5	150	350
K	H, G	5	5	250	250

Example2: Table on below gives the activities of a project, their predecessor(s), durations and costs.

a) Draw the network for the project.

b) Determine the critical path.

c) Obtain programs with minimum costs.

Activities	Predecessor(s)	Durations		Costs	
		Normal	Crashed	Normal	Crashed
A	---	5	3	100	150
B	---	4	2	180	200
C	---	6	3	170	200
D	A	7	5	200	260
E	B, C	4	4	120	120
F	B, C	7	7	150	150
G	C	5	3	180	250
H	D, F	8	6	200	300
I	E, G	9	6	100	160

References
[1] *Ahlatçıoğlu, M. , Tiryaki, F., Kantitatif Karar Verme Teknikleri, YTÜ Yayın No: YTÜ.FE.DK-98.0349, İstanbul-1998.*
[2] *H.A.Taha, Operations Research: An Introduction,* Prentice Hall; 9th edition, *Singapore, 2010.*
[3] *Winston, W., Operations Research and Applications and Algorithms. 1990.*