PERT SCHEDULING TECHNIQUES FOR ENGINEERING PROJECTS

INTRODUCTION AND HISTORICAL PERSPECTIVES

Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) are the two most frequently used project network analysis tools. Network analysis procedures originated from the traditional Gantt Chart, or bar chart, developed by Henry L. Gantt during World War I. There have been several mathematical techniques for scheduling activities, especially where resource constraints are a major factor. Unfortunately, the mathematical formulations are not generally practical due to the complexity involved in implementing them for realistically large projects. Even computer implementations of the mathematical techniques sometimes become too cumbersome for real-time managerial applications. It should be recalled that the people responsible for project schedules are the managers, who justifiably, prefer simple and quick decision aids. To a project scheduler, a complex mathematical procedure constitutes an impediment rather than an aid in the scheduling process. Nonetheless, the premise of the mathematical formulations rests on their applicability to small projects consisting of very few activities. Many of the techniques have been evaluated, applied, and reported in the literature.

A more practical approach to scheduling is the use of heuristics. If the circumstances of a problem satisfy the underlying assumptions, a good heuristic will yield schedules that are feasible enough to work with. A major factor in heuristic scheduling is to select a heuristic with assumptions that are widely applicable. A wide variety of scheduling heuristics exists for a wide variety of special cases. The procedure for using heuristics to schedule projects involves prioritizing activities in the assignment of resources and time slots in the project schedule. Many of the available priority rules consider activity durations and resource requirements in the scheduling process.

If all activities are assigned priorities at the beginning and then scheduled, the scheduling heuristic is referred to as a *serial method*. If priorities are assigned to the set of activities eligible for scheduling at a given instant and the schedule is developed concurrently, then the scheduling heuristic is referred to as a *parallel method*. In the serial method, the relative priorities of activities remain fixed. In the parallel methods, the priorities change with the current composition of activities. The techniques presented in this chapter are particularly useful for engineering projects because of the unique characteristics of such projects.

PERT/CPM TOOLS

The network of activities contained in a project provides the basis for scheduling the project. Although CPM and PERT are the two most popular techniques for project network analysis, the Precedence Diagramming Method (PDM) has gained in popularity in recent years because of its relevance for concurrent engineering applications. A project network is the graphical representation of the contents and objectives of a project. The basic project network analysis is typically implemented in three phases: network planning phase, network scheduling phase, and network control phase.

Network planning is sometimes referred to as activity planning. This involves the identification of the relevant activities for the project. The required activities and their precedence relationships are determined. Precedence requirements may be determined on the basis of technological, procedural, or imposed constraints. The activities are then represented in the form of a network diagram. The two popular models for network drawing are the activity-on-arrow (AOA) and the activity-on-node (AON) conventions. In the AOA approach, arrows are used to represent activities, while nodes represent starting and ending points of activities. In the AON approach, nodes represent activities, while arrows represent precedence relationships. Estimates of time, cost, and resource requirements are developed for each activity during the network planning phase. Time estimates may be based on historical records, time standards, forecasting, regression analysis, or other quantitative methods.

Network scheduling is performed by using forward pass and backward pass computational procedures. These computations give the earliest and latest starting and finishing times for each activity. The slack time or float associated with each activity is determined during the forward/backward network computations. The activity path with the minimum slack in the network is used to determine the critical activities. This path also determines project duration. Resource allocation and time-cost tradeoffs are other functions performed during network scheduling.

Network control involves tracking the progress of a project on the basis of the network schedule and taking corrective actions when needed. An evaluation of actual performance versus expected performance determines deficiencies in the project progress. The overall procedure is summarized in the steps below.

Step 1 (Planning Phase). *Activity planning*. The activities making up the project are defined and their interdependencies or precedence relationships are determined. Precedence requirements may be determined on the basis of technological, procedural, or imposed constraints. A tabulated presentation of the project data should be prepared in this step. The table should include activity description, code (if desired), estimated duration, predecessors, and resource requirements.

Step 2 (Planning Phase). *Activity network drawing*. The activities defined in Step 1 are represented in the form of a network diagram.

Step 3 (Scheduling Phase). *Basic scheduling*. The basic scheduling computations are performed through forward-pass and backward-pass rules. These computations yield the earliest and latest allowable start and finish times for each activity. The amount of slack or float associated with each activity is determined. The activity path with the minimum slack through the network is used to determine the critical activities.

Step 4 (Scheduling Phase). *Time/cost trade-offs*. Time cost trade-offs analysis may be performed if the project analyst is interested in determining the cost of reducing the project length.

Step 5(Scheduling Phase). *Constrained resource allocation*. Constrained resource allocation refers to the process of allocating limited resources to competing activities in the project. The feasibility of each schedule must be checked with respect to resource requirements and availability. This is where heuristic scheduling rules are used to determine which set of the competing activities gets resources first.

Step 6 (Scheduling Phase). *Resource leveling*. If desired, activity shifting or rearrangement may be performed to reduce period-to-period fluctuations in resource requirements. This is very beneficial if it is managerially unacceptable to change the size of the work force frequently.

Step 7 (Control Phase). When the project network plan and schedule have been developed and found to be acceptable to management, they are prepared in a final form for field implementation. The project progress and performance are monitored by comparing the actual project status to the prevailing schedule. Monitoring permits frequent reviews and revisions of the project plan. If needed, corrective actions are taken to bring the project back in line with the plan and schedule. Thus, monitoring serves as the progress review of a project while corrective action serves as the control. The advantages of project network analysis are presented here.

- Advantages for communication:
  - It clarifies project objectives.
  - It establishes the specifications for project performance.
  - It provides a starting point for more detailed task analysis.
  - It presents a documentation of the project plan.
  - It serves as a visual communication tool.
- Advantages for control:
  - It presents a measure for evaluating project performance.
  - It helps determine what corrective actions are needed.
  - It gives a clear message of what is expected.
  - It encourages team interactions.
- Advantages for team interaction:
  - It offers a mechanism for a quick introduction to the project.
  - It specifies functional interfaces on the project.
  - It facilitates ease of application.

Figure 1 shows the graphical representation for the AON network. The components of the network are explained here.

1. Node. A node is a circular representation of an activity.
2. Arrow. An arrow is a line connecting two nodes and having an arrowhead at one end. The arrow implies that the activity at the tail of the arrow precedes the one at the head of the arrow.
3. Activity. An activity is a time-consuming effort required to perform a part of the overall project. An activity is represented by a node in the AON system or by an arrow in the AOA system. The job the activity represents may be indicated by a short phrase or symbol inside the node or along the arrow.
4. Restriction. A restriction is a precedence relationship that established the sequence of activities. When one activity must be completed before another activity can begin, the first is said to be a predecessor of the second.
5. Dummy. A dummy is used to indicate one event of a significant nature (e.g., milestone). It is denoted by a dashed circle and treated as an activity with zero time duration. A dummy is not required in the AON method. However, it may be included for convenience, network clarification, or to represent a milestone in the progress of the project.
6. Predecessor activity. A predecessor activity is one that immediately precedes the one being considered.
7. Successor activity. A successor activity is one which immediately follows the one being considered.
8. Descendant activity. A descendant is any activity restricted by the one under consideration.
9. Antecedent activity. An antecedent activity is any activity that must precede the one being considered. Activities In Figure 1, A and B are antecedents of D. Activity A is antecedent of B and A has no antecedent.
10. Merge point. A merge point exists when two or more activities are predecessors to a single activity. All activities preceding the merge point must be completed before the merge activity can commence.
11. Burst point. A burst point exists when two or more activities have a common predecessor. None of the activities emanating from the same predecessor activity can be started until the burst point activity is completed.
12. Precedence diagram. A precedence diagram is a graphical representation of the activities making up a project and the precedence requirements needed to complete the project. Time is conventionally shown to be from left to right, but no attempt is made to make the size of the nodes or arrow proportional to time.

PRECEDENCE STRUCTURE

Precedence relationships in a CPM network fall into the three major categories listed here:

1. Technical precedence
2. Procedural precedence

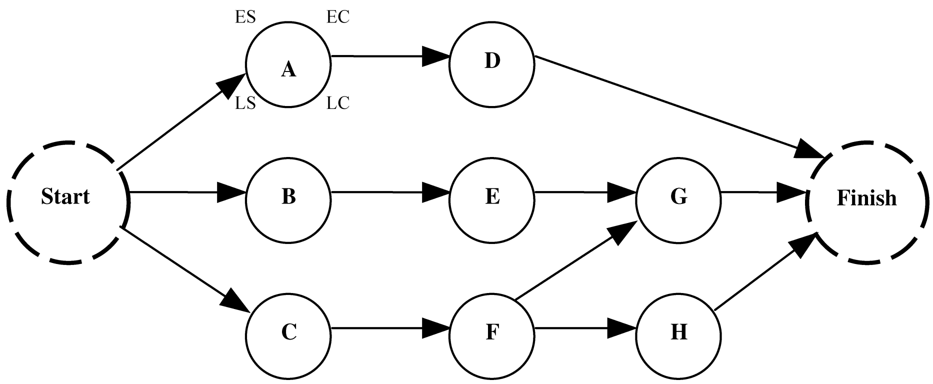

Figure 1. Graphical representation of AON network.

3. Imposed precedence

Technical precedence requirements are caused by the technical relationships among activities in a project. For example, in conventional construction, walls must be erected before the roof can be installed. Procedural precedence requirements are determined by policies and procedures. Such policies and procedures are often subjective, with no concrete justification. Imposed precedence requirements can be classified as resource-imposed, state-imposed, or environment-imposed. For example, resource shortages may require that one task precede another. The current status of a project (e.g. percent completion) may determine that one activity be performed before another. The environment of a project, for example, weather changes or the effects of concurrent projects, may determine the precedence relationships of the activities in a project.

The primary goal of a CPM analysis of a project is the determination of the *critical path*. The critical path determines the minimum completion time for a project. The analysis involves forward-pass and backward-pass computations. The forward pass determines the earliest start time and the earliest completion time for each activity in the network. The backward pass determines the latest start time and the latest completion time for each activity. Conventional network logic is always drawn from left to right. If this convention is followed, there is no need to use arrows to indicate the directional flow in the activity network. The notations used for activity A in the network are explained below:

A: Activity ID
ES: Earliest starting time
EC: Earliest completion time
LS: Latest starting time
LC: Latest completion time
t: Activity duration

During the forward pass analysis of the network, it is assumed that each activity will begin at its earliest starting time. An activity can begin as soon as the last of its predecessors is finished. The completion of the forward pass determines the earliest completion time of the project. The backward pass analysis is a reverse of the forward pass. The project begins at its latest completion time and ends at the latest starting time of the first activity in the project network. The rules for implementing the forward pass and backward pass analyses in CPM are presented below. These rules are implemented iteratively until the ES, EC, LS, and LC have been calculated for all nodes in the activity network.

Rule 1: Unless otherwise stated, the starting time of a project is set equal to time zero. That is, the first node, *node 1* in the network diagram has an earliest start time of zero. Thus,

$ES(1) = 0$

If a desired starting time t_0 is specified, then

$ES(1) = t_0$

Rule 2: The earliest start time (ES) for any node (activity j) is equal to the maximum of the earliest completion times (EC) of the immediate predecessors of the node. That is,

$\text{ES}(i) = \; j \in P(i)\text{Max}\{\text{EC}(\,j)\}$

where $P(i)$ = set of immediate predecessors of activity i).

Rule 3: The earliest completion time (EC) of activity i is the activity's earliest start time plus its estimated time t_i. That is,

$\text{EC}(i) = \text{ES}(i) + t_i.$

Rule 4: The earliest completion time of a project is equal to the earliest completion time of the very last node, *node n*, in the network. That is,

$\text{EC}(\textit{Project}) = \text{EC}(n).$

Rule 5: Unless the latest completion time (LC) of a project is explicitly specified, it is set equal to the earliest completion time of the project. This is called the *zero-project-slack convention*. That is,

$\text{LC}(\textit{Project}) = \text{EC}(\textit{Project}).$

Rule 6: If a desired deadline T_p is specified for the project, then

$\text{LC}(\textit{Project}) = T_p$. It should be noted that a latest completion time or deadline may sometimes be specified for a project on the basis of contractual agreements.

Rule 7: The latest completion time (LC) for activity j is the smallest of the latest start times of the activity's immediate successors. That is,

$\text{LS}(\,j) = i \in S(\,j)\text{Min}$ where $S(j)$ = immediate successors of activity j

Rule 8: The latest start time for activity j is the latest completion time minus the activity time. That is,

$\text{LS}(\,j) = \text{LC}(\,j) - t_i.$

CPM Example

Table 1 presents the data for a simple project network. The AON network for the example is given in Figure 2. Dummy activities are included in the network to designate single starting and ending points for the project.

Forward Pass

The forward pass calculations are shown in Figure 3. Zero is entered as the ES for the initial node. Because the initial node for the example is a dummy node, its duration is zero. Thus, EC for the starting node is equal to its ES. The ES values for the immediate successors of the starting node are set equal to the EC of the START node and the resulting EC values are computed. Each node is treated as the "start" node for its successor or successors. However, if an activity has more than one predecessor, the maximum of the ECs of the preceding activities is used as the activity's starting time. This happens in the case of activity G, whose ES is determined as Max {6,5,9} = 9. The earliest project completion time for the example is 11 days. Note that this is the maximum of the immediately preceding earliest completion times: Max {6,11} = 11. Since the dummy ending node has no duration, its earliest completion time is set equal to its earliest start time of 11 days.

Backward Pass

The backward pass computations establish the latest start time (LS) and latest completion time (LC) for each node in the network. The results of the backward pass computations are shown in Figure 4. Because no deadline is specified, the latest completion time of the project is set equal to the earliest completion time. By back tracking and using the network analysis rules presented earlier, the latest completion and start times are determined for each node. Note that in the case of activity A, which has two immediate successors, the latest completion time is determined as the minimum of the immediately succeeding latest start times. That is, Min {6,7} = 6. A similar situation occurs for the dummy starting node. In that case, the latest completion time of the dummy start node is Min {0,3,4} = 0. As this dummy node has no duration, the latest starting time of the project is set equal to the node's latest completion time. Thus, the project starts at time 0 and is expected to be completed by time 11.

Within a project network, there are usually several possible paths and a number of activities that must be performed sequentially and some activities that may be performed concurrently. If an activity has ES and EC times that are not equal, then the actual start and completion times of that activity may be flexible. The amount of flexibility an activity possesses is called a slack time. The slack time is used to determine the critical activities in the network as discussed in the next section.

Determination of Critical Activities

The critical path is defined as the path with the least slack in the network diagram. All activities on the critical path are said to be critical activities. These activities can create bottlenecks in the network if they are delayed. The critical path is also the longest path in the network diagram. In some networks, particularly large ones, it is possible to have multiple critical paths. If many paths exist in the network, it may be very difficult to visually identify all the critical paths. The slack time of an activity is also referred to as its *float*. There are four basic types of activity slack. They are described here.

- Total Slack (TS) Total Slack is defined as the amount of time an activity may be delayed from its earliest starting time without delaying the latest completion time of the project. The total slack time of an activity

Table 1. Data for Sample Project for CPM Analysis

Activity	Predecessor	Duration (days)
A	–	2
B	–	6
C	–	4
D	A	3
E	C	5
F	A	4
G	B, D, E	2

Table 2. Result of CPM Analysis for Sample Project

Activity	Duration	ES	EC	LS	LC	TS	FS	Criticality
A	2	0	2	4	6	4	0	–
B	6	0	6	3	9	3	3	–
C	4	0	4	0	4	0	0	Critical
D	3	2	5	6	9	4	4	–
E	5	4	9	4	9	0	0	Critical
F	4	2	6	7	11	5	5	–
G	2	9	11	9	11	0	0	Critical

Table 3. Table of d_2 Values

n	5	10	15	20	25	30	40	50	75	100
d_2	2.326	3.078	3.472	3.735	3.931	4.086	4.322	4.498	4.806	5.015

Table 4. PERT Project Data

Activity	Predecessors	a	m	b	t_e	s^2
A	–	1	2	4	2.17	0.2500
B	–	5	6	7	6.00	0.1111
C	–	2	4	5	3.83	0.2500
D	A	1	3	4	2.83	0.2500
E	C	4	5	7	5.17	0.2500
F	A	3	4	5	4.00	0.1111
G	B,D,E	1	2	3	2.00	0.1111

Table 5. Data for Project Complexity Example

Activity Number	PERT Estimates (a, m, b)	Preceding Activities	Required Resources (x_{i1}, x_{i2})
1	1, 3, 5	–	1, 0
2	0.5, 1, 3	–	1, 1
3	1, 1, 2	–	1, 1
4	2, 3, 6	Activity 1	2, 0
5	1, 3, 4	Activity 2	1, 0
6	1.5, 2, 2	Activity 3	4, 2
			$Z_1 = 5, Z_2 = 2$

Table 6. Initial Table of Work Rate Analysis

Resource	Work Rate	Time	Work Done
Machine A	$1/x$	t	1.0

Table 7. Multiple Resource Work Rate Layout

Resource Type I	Work Rate, r_i	Time, t_i	Work Done, w_i
RES 1	r_1	t_1	$(r_1)(t_1)$
RES 2	r_2	t_2	$(r_2)(t_2)$
...	...	...	...
RES n	r_n	t_n	$(r_n)(t_n)$
		Total	1.0

Table 8. Example for Two Resource Types

Resource Type i	Work rate, r_i	Time, t_i	Work Done, w_i
RES 1	1/50	15	15/50
RES 2	r_2	15	$15(r_2)$
		Total	4/5

Table 9. Incorporation of Pay Rate into Work Rate Analysis

Resource, i	Work Rate, r_i	Time, t_i	Work Done, w	Pay Rate, p_i	Total Cost, C_i
Machine A	r_1	t_1	$(r_1)(t_1)$	p_1	C_1
Machine B	r_2	t_2	$(r_2)(t_2)$	p_2	C_2
...	...	...	...	...	...
Machine n	r_n	t_n	$(r_n)(t_n)$	p_n	C_n
		Total	1.0		Budget

Table 10. Calculations of Unknown Duration

Resource type, i	Work Rate, r_i	Time, t_i	Work Done, w_i
RES 1	1/30	T	T/30
RES 2	1/22	T	T/22
		Total	3/5

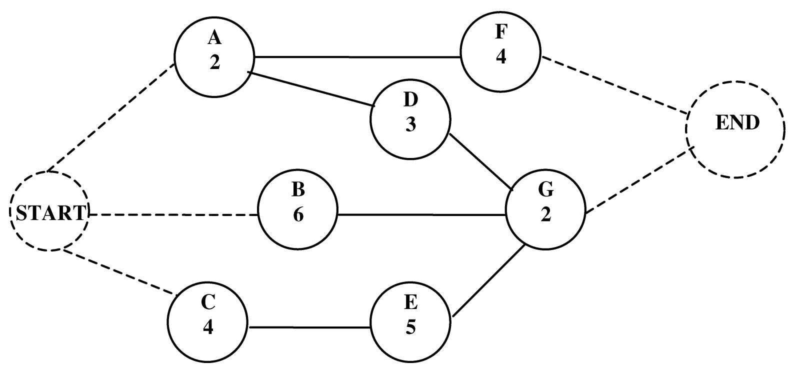

Figure 2. Example of Activity Network.

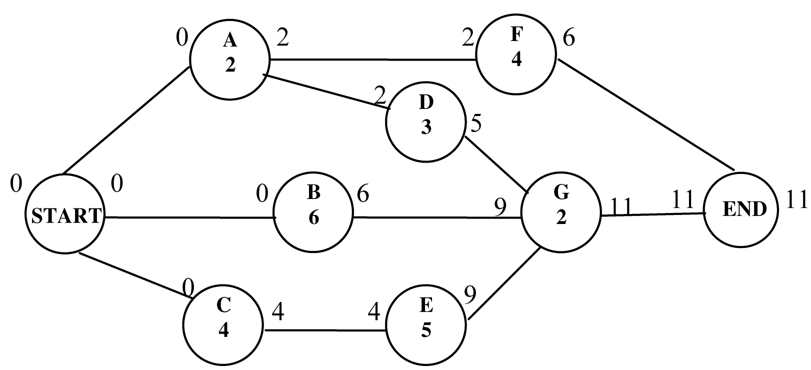

Figure 3. Forward Pass Analysis for CPM Example

is the difference between the latest completion time and the earliest completion time of the activity, or the difference between the latest starting time and the earliest starting time of the activity.

$$TS(j) = LC(j) - EC(j)$$
or
$$TS(j) = LS(j) - ES(j)$$

Total Slack is the measure that is used to determine the critical activities in a project network. The critical activities are identified as those having the minimum total slack in the network diagram. If there is only one critical path in the network, then all the critical activities will be on that one path.

- Free Slack (FS). Free Slack is the amount of time an activity may be delayed from its earliest starting time

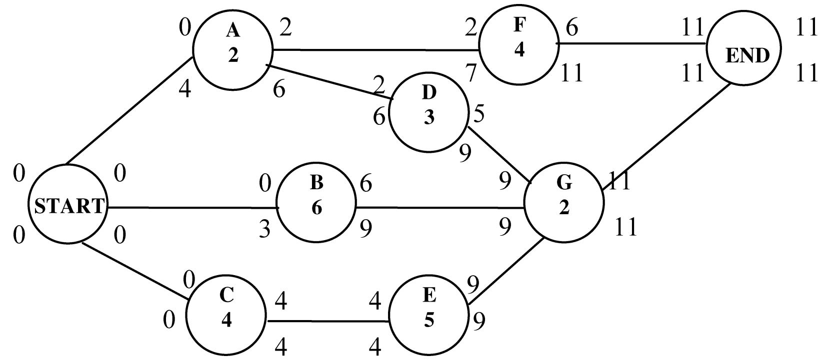

Figure 4. Backward Pass Analysis for CPM Example

without delaying the starting time of any of its immediate successors. Activity free slack is calculated as the difference between the minimum earliest starting time of the activity's successors and the earliest completion time of the activity.

$$\mathrm{FS}(j) = j \in S(j)\mathrm{Min}\{\mathrm{ES}(i)'s\} - \mathrm{EC}(j).$$

- Interfering Slack (IS). Interfering Slack or interfering float is the amount of time by which an activity interferes with (or obstructs) its successors when its total slack is fully used. This is rarely used in practice. The interfering float is computed as the difference between the total slack and the free slack.

$$\mathrm{IS}(j) = \mathrm{TS}(j) - \mathrm{FS}(j).$$

- Independent Float (IF). Independent float or independent slack is the amount of float that an activity will always have regardless of the completion times of its predecessors or the starting times of its successors. Independent float is computed as:

$$\mathrm{IF} = j \in S(k), i \in P(k)\mathrm{Max}\{0, (j \in S(k)\mathrm{Min}\,\mathrm{ES}_j$$
$$-i \in P(k)\mathrm{Max}\,\mathrm{LC}_i - t_k)\}$$

Where ES_j is the earliest starting time of the succeeding activity, LC_i is the latest completion time of the preceding activity, and t is the duration of the activity whose independent float is being calculated. Independent float takes a pessimistic view of the situation of an activity. It evaluates the situation whereby the activity is pressured from either side. That is, when its predecessors are delayed as late as possible while its successors are to be started as early as possible. Independent float is useful for conservative planning purposes, but it is not used much in practice. Despite its low level of use, independent float does have practical implications for better project management. Activities can be buffered with independent floats as a way to handle contingencies.

For Figure 4 the total slack and the free slack for activity A are calculated, respectively, as:

$$\mathrm{TS} = 6 - 2 = 4\,\text{days}$$
$$\mathrm{FS} = \mathrm{Min}\{2, 2\} - 2 = 2 - 2 = 0.$$

Similarly, the total slack and the free slack for activity F are:

$$\mathrm{TS} = 11 - 6 = 5\,\text{days}$$
$$\mathrm{FS} = \mathrm{Min}\{11\} - 6 = 11 - 6 = 5\,\text{days}.$$

Table 2 presents a tabulation of the results of the CPM example. The Table contains the earliest and latest times for each activity as well as the total and free slacks. The results indicate that the minimum total slack in the network is zero. Thus, activities C, E, and G are identified as the critical activities. The critical path is highlighted in Figure 4 and consists of the following sequence of activities:

$$\text{Start} \rightarrow \text{C} \rightarrow \text{E} \rightarrow \text{G} \rightarrow \text{End}$$

The total slack for the overall project itself is equal to the total slack observed on the critical path. The minimum slack in most networks will be zero since the ending LC is set equal to the ending EC. If a deadline is specified for a project, then we would set the project's latest completion time to the specified deadline. In that case, the minimum total slack in the network would be given by the expression below:

$$\mathrm{TS}_{\mathrm{Min}} = (\text{Project Deadline}) - \text{EC of the last node.}$$

This minimum total slack will appear as the total slack for each activity on the critical path. If a specified deadline is lower than the EC at the finish node, then the project will start out with a negative slack. That means that it will be behind schedule before it even starts. It may then become necessary to expedite some activities (i.e., crashing) in order to overcome the negative slack. Figure 5 shows an example with a specified project deadline. In this case, the deadline of 18 days comes after the earliest completion time of the last node in the network.

Using Forward Pass to Determine the Critical Path

The critical path in CPM analysis can be determined from the forward pass only. This can be helpful in cases where it is desired to quickly identify the critical activities without performing all the other calculations needed to obtain the latest starting times, the latest completion times, and total slacks. The steps for determining the critical path from the forward pass only are:

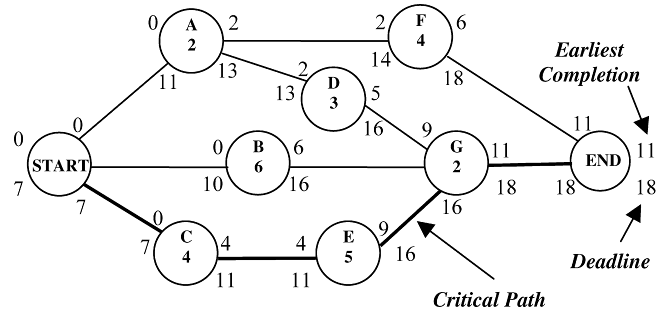

Figure 5. CPM Network with Deadline

1. Complete the forward pass in the usual manner.
2. Identify the last node in the network as a critical activity.
3. If activity i is an immediate predecessor of activity j, which is determined as a critical activity, then check EC_i and ES_j. If $EC_i = ES_j$, then label activity i as a critical activity. When all immediate predecessors of activity j are considered, mark activity j.
4. Continue the backtracking from each unmarked critical activity until the project starting node is reached. Note that if there is a single starting node or a single ending node in the network, then that node will always be on the critical path.

Gantt Charts

When the results of a CPM analysis are fitted to a calendar time, the project plan becomes a schedule. The Gantt chart is one of the most widely used tools for presenting a project schedule. A Gantt chart can show planned and actual progress of activities. The time scale is indicated along the horizontal axis, while horizontal bars or lines representing activities are ordered along the vertical axis. As a project progresses, markers are made on the activity bars to indicate actual work accomplished. Gantt charts must be updated periodically to indicate project status. Figure 6 presents the Gantt chart for our illustrative example using the earliest starting (ES) times from Table 2. Figure 7 presents the Gantt chart for the example based on the latest starting (LS) times. Critical activities are indicated by the shaded bars.

Figure 6 shows that the starting time of activity F can be delayed from day two until day seven (i.e., TS = 5) without delaying the overall project. Likewise, A, D, or both may be delayed by a combined total of four days (TS = 4) without delaying the overall project. If all the four days of slack are used up by A, then D cannot be delayed. If A is delayed by one day, then D can be delayed only by up to three days, without causing a delay of G, which determines project completion. The Gantt chart also indicates that activity B may be delayed by up to three days without affecting the project completion time.

In Figure 7, the activities are scheduled by their latest completion times. This represents a pessimistic case where activity slack times are fully used. No activity in this schedule can be delayed without delaying the project. In Figure 7, only one activity is scheduled over the first three days. This may be compared to the schedule in Figure 6, which has three starting activities. The schedule in Figure 7 may be useful if there is a situation that permits only a few activities to be scheduled in the early stages of the project. Such situations may involve shortage of project personnel, lack of initial budget, time for project initiation, time for personnel training, allowance for learning period, or general resource constraints. Scheduling of activities based on ES times indicates an optimistic view. Scheduling on the basis of LS times represents a pessimistic approach.

PERT NETWORK ANALYSIS

Program Evaluation Review Technique (PERT) is an extension of CPM which incorporates variability in activity durations into project network analysis. PERT has been used extensively and successfully in practice.

In real life, activities are often prone to uncertainties that determine the actual durations of the activities. In CPM, activity durations are assumed to be deterministic. In PERT, the potential uncertainties in activity durations are accounted for by using three time estimates for each activity. The three time estimates represent the spread of the estimated activity duration. The greater the uncertainty of an activity, the wider the range of the estimates.

PERT uses three time estimates and PERT formulas to compute the expected duration and variance for each activity. The PERT formulas are based on a simplification of the expressions for the mean and variance of a beta distribution. The approximation formula for the mean is a simple weighted average of the three time estimates, with the end points assumed to be equally likely while the mode is assumed to be four times as likely. The approximation formula for PERT is based on the recognition that most of the observations from a distribution will lie within plus or minus three standard deviations, or a spread of six standard deviations. This leads to the simple method of setting the PERT formula for standard deviation equal to one sixth of the estimated duration range. While there is no theoretical validation for these approximation approaches, the PERT formulas do facilitate ease of use. The formulas are presented below:

$$t_e = \frac{a + 4m + b}{6}$$

$$s^2 = \frac{(b-a)^2}{36},$$

Where:

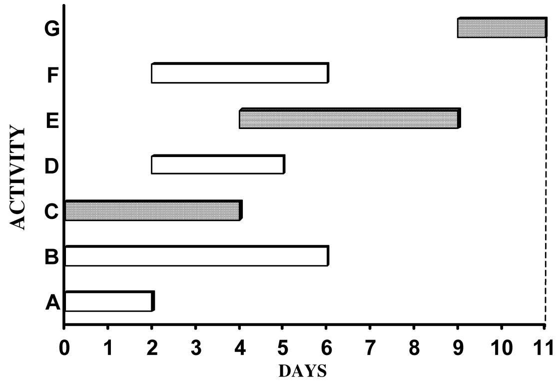

Figure 6. Gantt Chart Based on Earliest Starting Times

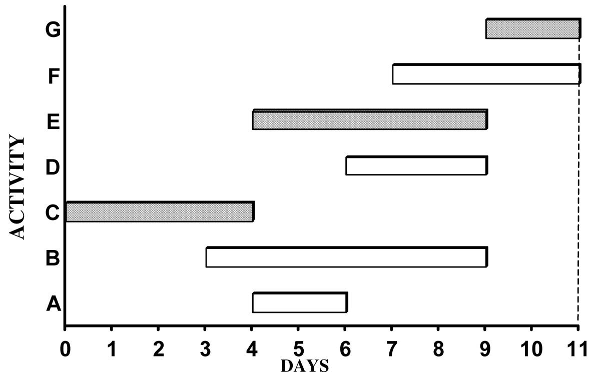

Figure 7. Gantt Chart Based on Latest Starting Times

a = optimistic time estimate
m = most likely time estimate
b = pessimistic time estimate
$a < m < b$
t_e = expected time for the activity; and
s^2 = variance of the duration of the activity.

After obtaining the estimate of the duration for each activity, the network analysis is carried out using the same forward and backward calculations presented previously for CPM. The major steps in PERT analysis are summarized below:

1. Obtain three time estimates a, m, and b for each activity.
2. Compute the expected duration for each activity by using the formula for t_e.
3. Compute the variance of the duration of each activity from the formula for s^2. It should be noted that CPM analysis cannot calculate variance of activity duration because it uses a single time estimate for each activity.
4. Compute the expected project duration, T_e. As in the case of CPM, the duration of a project in PERT analysis is the sum of the durations of the activities on the critical path.
5. Compute the variance of the project duration as the sum of the variances of the activities on the critical path. The variance of the project duration is denoted by S^2. It should be recalled that CPM cannot compute the variance of the project duration because individual variances of activity durations are not computed.
6. If there are two or more critical paths in the network, choose the one with the largest variance to determine the project duration and its variance. Thus, PERT is pessimistic with respect to the variance of project duration when there are multiple critical paths in the network. For some networks, it may be necessary to perform a ***mean-variance analysis*** to determine the relative importance of the multiple paths by plotting the expected project duration versus the path duration variance.
7. If desired, compute the probability of completing the project within a specified time period. This is not possible under CPM.

In practice, a question often arises as to how to obtain good estimates of a, m, and b. Several approaches can be used in obtaining the required time estimates for PERT. Some of the commonly used approaches are described below:

- Estimates furnished by an experienced person
- Estimates extracted from standard time data
- Estimates obtained from historical data
- Estimates obtained from simple regression and/or forecasting
- Estimates generated by simulation
- Estimates derived from heuristic assumptions
- Estimates dictated by customer requirements

The pitfall of using estimates furnished by an individual is that they may be inconsistent, since they are limited by the experience and personal bias of the person providing them. Individuals responsible for furnishing time estimates are usually not experts in estimation, and they generally have difficulty in providing accurate PERT time estimates. There is often a tendency to select values of a, m, and b that are optimistically skewed. This is because a conservatively large value is typically assigned to b by inexperienced individuals.

The use of time standards, on the other hand, may not reflect the changes occurring in the current operating environment due to new technology, work simplification, new personnel, and so on. The use of historical data and forecasting is very popular because estimates can be verified and validated by actual records. In the case of regression and forecasting, there is the danger of extrapolation beyond the data range used for fitting the regression and forecasting models. If the sample size in a historical data set is sufficient and the data can be assumed to reasonably represent prevailing operating conditions, the three PERT estimates can be computed as follows:

$$\begin{aligned} \hat{a} &= \bar{t} - kR \\ \hat{m} &= \bar{t} \\ \hat{b} &= \bar{t} + kR \end{aligned}$$

where R = range of the sample data; $\bar{t}$ = arithmetic average of the sample data; $k = 3/d_2$; and d_2 = an adjustment factor for estimating the standard deviation of a population. If $kR > \bar{t}$, then set $a = 0$ and $b = 2\bar{t}$. The factor d_2 is widely tabulated in the quality control literature as a function of the number of sample points, n. Selected values of d_2 are presented in Table 3.

In practice, probability distributions of activity times can be determined from historical data. The procedure involves three steps:

1. Appropriate organization of the historical data into histograms.
2. Determination of a distribution that reasonably fits the shape of the histogram.
3. Testing of the goodness-of-fit of the hypothesized distribution by using an appropriate statistical model. The Chi-square Test and the Kolmogrov-Smirnov (K-S) test are two popular methods for testing goodness-of-fit. Most statistical texts present the details of how to carry out goodness-of-fit tests.

Activity Time Distributions in PERT

PERT analysis assumes that the probabilistic properties of activity duration can be modeled by the beta probability density function. The beta distribution is defined by two end points and two shape parameters. The beta distribution was chosen by the original developers of PERT as a reasonable distribution to model activity times because it has finite end points and can assume a variety of shapes based on different shape parameters. While the true distribution of activity time will rarely ever be known, the beta distribution serves as an acceptable model. Figure 8 shows examples of alternate shapes of the standard beta distribution between zero and one. The uniform distribution between 0 and 1 is a special case of the beta distribution with both shape parameters equal to one.

The standard beta distribution is defined over the interval 0 to 1, while the general beta distribution is defined over any interval a to b. The general beta probability density function is given by:

$$f(t) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot \frac{1}{(b-a)^{\alpha+\beta-1}} \cdot (t-a)^{\alpha-1}(b-t)^{\beta-1}$$

for $a \le t \le b$ and $a > 0,\ \beta > 0$.

where: a = lower end point of the distribution; b = upper end point of the distribution; and α, β are the shape parameters of the distribution. The mean, variance, and mode of the general beta distribution are defined as shown below:

$$\begin{aligned} \mu &= a + (b-a)\frac{\alpha}{\alpha+\beta} \\ \sigma^2 &= (b-a)^2 \frac{\alpha\beta}{(\alpha+\beta+1)(\alpha+\beta)^2} \\ m &= \frac{a(\beta-1)+b(\alpha-1)}{\alpha+\beta-2}. \end{aligned}$$

The general beta distribution can be transformed into a standardized distribution by changing its domain from [a,b] to the unit interval, [0,1]. This is accomplished by using the relationship $t_s = a + (b-a)t_s$, where t_s is the standard beta random variable between 0 and 1. This yields the standardized beta distribution, given by:

$$\begin{aligned} f(t) &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} t^{\alpha-1}(1-t)^{\beta-1}; 0 < t < 1; \alpha, \beta > 0 \\ &= 0; \quad \text{elsewhere}, \end{aligned}$$

which has mean, variance, and mode defined as:

$$\begin{aligned} \mu &= \frac{\alpha}{\alpha+\beta} \\ \sigma^2 &= \frac{\alpha\beta}{(\alpha+\beta+1)(\alpha+\beta)^2} \\ m &= \frac{a(\beta-1)+b(\alpha-1)}{\alpha+\beta-2}. \end{aligned}$$

The triangular probability density function has been used as an alternative to the beta distribution for modeling activity times. The triangular density has three essential parameters: a minimum value (a), a mode (m) and a maximum (b). The triangular density function is defined mathematically as:

$$\begin{aligned} f(t) &= \frac{2(t-a)}{(m-a)(b-a)}; \quad a \le t \le m \\ &= \frac{2(b-t)}{(b-m)(b-a)}; \quad m \le t \le b, \end{aligned}$$

Which has mean and variance defined, respectively, as:

$$\begin{aligned} \mu &= \frac{a+m+b}{3} \\ \sigma^2 &= \frac{a(a-m)+b(b-a)+m(m-b)}{18} \end{aligned}$$

Figure 9 presents a graphical representation of the triangular density function. The three time estimates of PERT can be inserted into the expression for the mean of the triangular distribution to obtain an estimate of the expected activity duration. Recall that in the conventional PERT formula, the mode (m) is assumed to carry four times as much

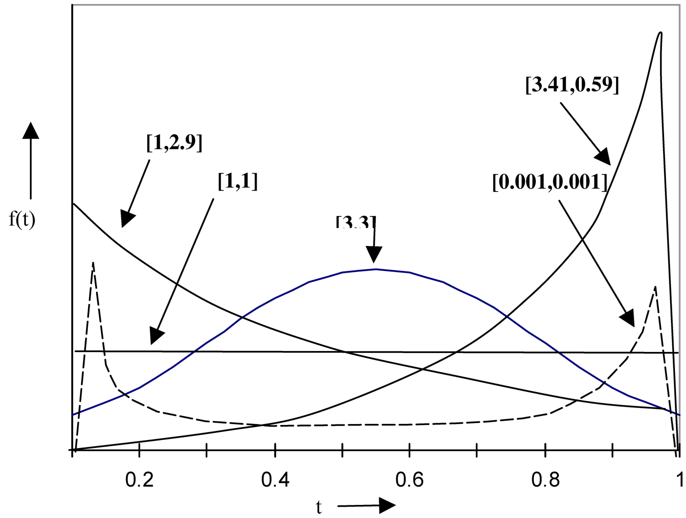

Figure 8. Beta distribution profiles for PERT

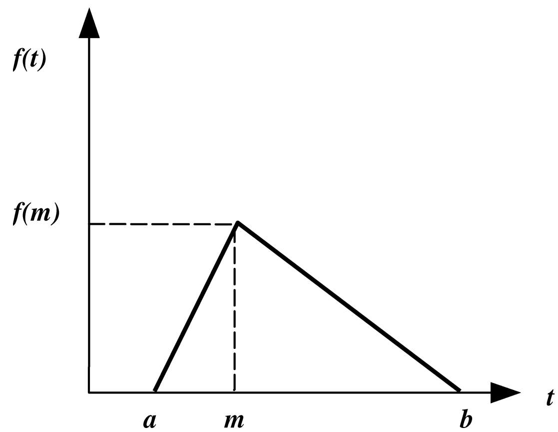

Figure 9. Triangular probability density function.

weight as either a or b when calculating the expected activity duration. By contrast, under the triangular distribution, the three time estimates are assumed to carry equal weights.

For cases where only two time estimates instead of three are to be used for network analysis, the uniform density function may be assumed for activity times. This is acceptable for situations where extreme limits of an activity duration can be estimated and it can be assumed that the intermediate values are equally likely to occur. The uniform distribution is defined mathematically as follows:

$$\begin{aligned} f(t) &= \frac{1}{b-a}; \quad a \le t \le b \\ &= 0; \quad \text{otherwise,} \end{aligned}$$

with mean and variance defined, respectively, as:

$$\begin{aligned} \mu &= \frac{a+b}{2} \\ \sigma^2 &= \frac{(b-a)^2}{12}. \end{aligned}$$

Figure 10 presents a graphical representation of the uniform distribution. In the case of the uniform distribution, the expected activity duration is computed as the average of the upper and lower limits of the distribution. The appeal of using only two time estimates, a and b, is that the estimation error due to subjectivity can be reduced and the estimation task simplified. Even when a uniform distribution is not assumed, other statistical distributions can be modeled over the range of a to b.

Other distributions that have been explored for activity time modeling include the normal distribution, lognormal distribution, truncated exponential distribution, and Weibull distribution. Once the expected activity durations have been computed, the analysis of the activity network is carried out just as in the case of single-estimate CPM network analysis.

Project Duration Distribution

Regardless of the distribution assumed for activity durations, the *central limit theorem* suggests that the distribution of the project duration will be approximately normally distributed. The theorem states that the distribution of averages obtained from any probability density function will be approximately normally distributed if the sample size is large and the averages are independent. In mathematical terms, the theorem is stated as described below:

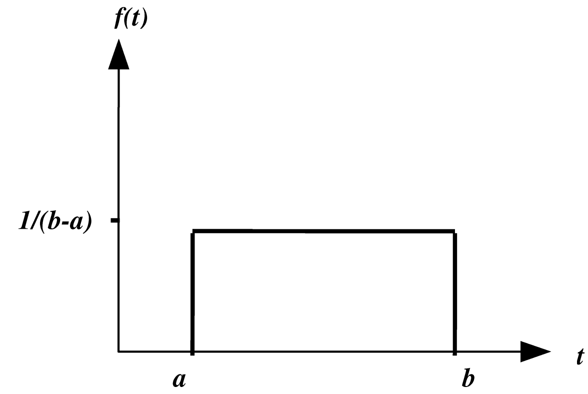

Figure 10. Uniform Probability Density Function

Central limit Theorem. Let $X_1, X_2, \ldots, X_N$ be independent and identically distributed random variables. Then, the sum of the random variables is normally distributed for large values of N. The sum is defined as:

$$T = X_1 + X_2 + \cdots + X_N$$

In activity network analysis, T represents the total project length as determined by the sum of the durations of the activities of the critical path. The mean and variance of T are expressed as:

$$\begin{aligned} \mu &= i = 1 \sum E[X_i] \\ \sigma^2 &= i = 1 \sum V[X_i], \end{aligned}$$

where E[X_i] = expected value of random variable X_i; and V[X_i] = variance of random variable X_i.

When applying the central limit theorem to activity networks, it should be noted that the assumption of independent activity times may not always be satisfied. Because of precedence relationships and other interdependencies of activities, some activity durations may not be independent.

PERT Analysis of Due Dates

If the project duration T_e can be assumed to be approximately normally distributed based on the central limit theorem, then the probability of meeting a specified deadline T_d can be computed by finding the area under the standard normal curve to the left of T_d. Figure 11 shows an example of a normal distribution describing the project duration.

Using the familiar transformation formula given here, a relationship between the standard normal random variable z and the project duration variable can be obtained:

$$z = \frac{T_d - T_e}{S},$$

where T_d = specified deadline; T_e = expected project duration based on network analysis; and S = standard deviation of the project duration. The probability of completing a project by the deadline T_d is then computed as:

$$P(T \le T_d) = P(z \le \frac{T_d - T_e}{S}).$$

The probability is obtained from the standard normal Table that is available in most statistics textbooks. Examples presented here illustrate the procedure for probability calculations in PERT. Suppose we have the project data presented in Table 4. The expected activity durations and variances as calculated by the PERT formulas are shown in the last two columns of the table. Figure 12 shows the PERT network. Activities C, E, and G are shown to be critical, and the project completion time is 11 time units.

The probability of completing the project on or before a deadline of 10 time units (i.e., T_d = 10) is calculated as shown below:

$$\begin{aligned} T_e &= 11 \\ S^2 &= V[C] + V[E] + V[G] \\ &= 0.25 + 0.25 + 0.1111 \\ &= 0.6111 \\ S &= \sqrt{0.6111} \\ &= .7817 \end{aligned}$$

$$\begin{aligned} P(T \le T_d) &= P(T \le 10) \\ &= P(z \le \frac{10 - T_e}{S}) \\ &= P(z \le \frac{10 - 11}{0.7817}) \\ &= P(z \le -1.2793) \\ &= 1 - P(z \le 1.2793) \\ &= 1 - 0.8997 \\ &= 0.1003 \end{aligned}$$

Thus, there is just over 10% probability of finishing the project within 10 days. By contrast, the probability of finishing the project in 13 days is calculated as:

$$\begin{aligned} P(T \le 13) &= P(z \le \frac{13 - 11}{0.7817}) \\ &= P(z \le 2.5585) \\ &= 0.9948 \end{aligned}$$

This implies that there is more than a 99% probability of finishing the project within 13 days. Note that the probability of finishing the project in exactly 13 days will be zero. If we desire the probability that the project can be completed within a certain lower limit (T_L) and a certain upper limit (T_U), the computation will proceed as follows:

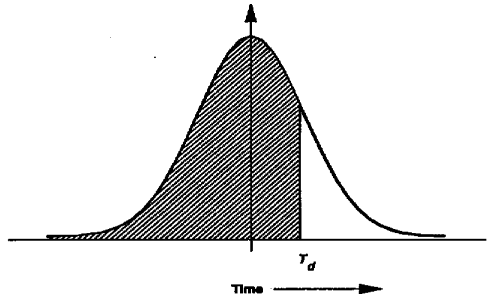

Figure 11. Area under the Normal Curve

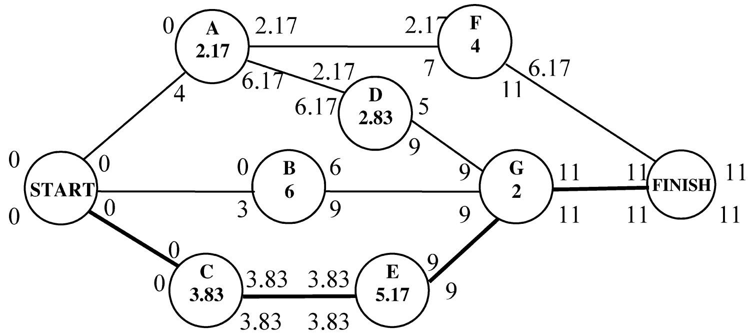

Figure 12. PERT Network Example

Let T_L =9 and T_U = 11.5. Then,

$$
\begin{aligned}
P(T_L \leq T \leq T_u) &= P(9 \leq T \leq 11.5) \\
&= P(T \leq 11.5) - P(T \leq 9) \\
&= P(z \leq \frac{11.5 - 11}{0.7817}) - P(z \leq \frac{9 - 11}{0.7817}) \\
&= P(z \leq 0.6396) - P(z \leq -2.5585) \\
&= P(z \leq 0.6396) - [1 - P(z \leq 2.5585)] \\
&= 0.7389 - [1 - 0.9948] \\
&= 0.7389 - 0.0052 \\
&= 0.7337.
\end{aligned}
$$

COMPLEXITY OF PERT NETWORKS

The performance of a scheduling heuristic will be greatly influenced by the complexity of the project network. The more activities there are in the network and the more resource types are involved, the more complex the scheduling effort. Numerous analytical experiments have revealed the lack of consistency in heuristic performances. Some heuristic perform well for both small and large projects. Some perform well only for small projects. Still, some heuristics that perform well for certain types of small projects may not perform well for other projects of comparable size. The implicit network structure based on precedence relationships and path interconnections influences network complexity and, hence, the performance of scheduling heuristics. The complexity of a project network may indicate the degree of effort that has been devoted to planning the project. The better the planning for a project, the lower the complexity of the project network can be expected to be. This is because many of the redundant interrelationships among activities can be identified and eliminated through better planning.

There have been some attempts to quantify the complexity of project networks. Because the structures of projects vary from very simple to very complex, it is desirable to have a measure of how difficult it will be to schedule a project. Some of the common measures of network complexity (C) are presented next.

For PERT networks

$$C = (\text{Number of Activities})^2/(\text{Number of Events}),$$

where an event is defined as an end point (or node) of an activity.

For precedence networks

$$C = (\text{Preceding Work Items})^2/(\text{Work Items})$$

The preceding expressions represent simple measures of the degree of interrelationship of the project network.

$$C = 2(A - N + 1)/(N - 1)(N - 2)$$

where A is the number of activities and N is the number of nodes in the project network. A measure defined as the Total Activity Density D is used to convey the complexity of a project netwrok. The network density is defined as:

$$D = i = 1 \sum \text{Max}\{0, (p_i - s_i)\}$$

where N is the number of activities, p_i is the number of predecessor activities for activity i, and s_i is the number of successor activities for activity i. Other measures of complexity include a measure of total work content for resource type $j(w_j)$, an obstruction factor (O), which is a measure of the ratio of excess resource requirements to total work content, adjusted obstruction per period based on earliest start time schedule (O_{est}), adjusted obstruction per period based on latest start time schedule (O_{lst}), and a resource utilization factor (U). These are computed as follows:

$$C = \frac{\text{number of activities}}{\text{number of nodes}}$$

$$D = \frac{\text{sum of job durations}}{\text{sum of job durations} + \text{total free slack}}$$

$$W_j = i = 1 \sum d_i r_{ij}$$
$$= i = 1 \sum r_{jt}$$

where d_i = duration of job i; r = per-period requirement of resource type j by job i; t = time period; N = number of jobs; CP = original critical path duration; and r_{jt} = total resource requirements of resource type j in time period t.

$$O = j = 1 \sum O_j$$
$$= j = 1 \sum \left(\frac{\mathrm{t} = 1 \sum \mathrm{Max}\{0, \mathrm{r_{jt}} - A_j\}}{w_j}\right)$$

where O_j = the obstruction factor for resource type j; CP = original critical path duration; A = units of resource type j available per period; M = number of different resource types; w_j = total work content for resource type j; and r_{jt} = total resource requirements of resource type j in time period t.

$$O_{\text{est}} = j = 1 \sum \left(\frac{t = 1 \sum \mathrm{Max}\{0, r_{jt(\text{est})} - A_j\}}{(M)(CP)}\right)$$

where $r_{jt(\text{est})}$ is the total resource requirements of resource type j in time period t based on earliest start times.

$$O_{\text{lst}} = j = 1 \sum \left(\frac{t = 1 \sum \mathrm{Max}\{0, r_{jt(\text{lst})} - A_j\}}{(M)(CP)}\right)$$

where $r_{jt(\text{lst})}$ is the total resource requirements of resource type j in time period t based on latest start times. The measures O_{est} and O_{lst} incorporate the calculation of excess resource requirements adjusted by the number of periods and the number of different resource types.

$$U = \mathrm{Max}_j\{f_j\}$$
$$= \mathrm{Max}_j\left\{\frac{w_j}{(CP)(A_j)}\right\}$$

where f_j is the resource utilization factor for resource type j. This measures the ratio of the total work content to the total work initially available. Badiru's measure of network complexity (1) is defined by the expression below:

$$\lambda = \frac{p}{d}\left[(1 - \frac{1}{L}) i = 1 \sum t_i + j = 1 \sum \left(\frac{i = 1 \sum t_i x i j}{Z_j}\right)\right]$$

where λ = project network complexity; L = number of activities in the network; t_i = expected duration for activity i; R = number of resource types; x_{ij} = units of resource type j required by activity i; Z_j = maximum units of resource type j available; p = maximum number of immediate predecessors in the network; and d = PERT duration of the project with no resource constraint.

The terms in the expression for the complexity are explained as follows: the maximum number of immediate predecessors, p,xs is a multiplicative factor that increases the complexity and potential for bottlenecks in a project network. The (1 − 1/L) term is a fractional measure (between 0.0 and 1.0) that indicates the time intensity or work content of the project. As L increases, the quantity (1 − 1/L) increases, and a larger fraction of the total time requirement (sum of t_i) is charged to the network complexity. Conversely, as L decreases, the network complexity decreases proportionately with the total time requirement. The sum of $(t_i x_{ij})$ indicates the time-based consumption of a given resource type j relative to the maximum availability. The term is summed over all the different resource types. Having the project duration in the denominator helps to express the complexity as a dimensionless quantity by cancelling out the time units in the numerator. In addition, it gives the network complexity per unit of total project duration.

There is always a debate as to whether or not the complexity of a project can be accurately quantified. There are several quantitative and qualitative factors with unknown interactions that are present in any project. As a result, any measure of project complexity should be used as a relative measure of comparison rather than as an absolute indication of the difficulty involved in scheduling a given project.

Because the performance of a scheduling approach can deteriorate sometimes with the increase in project size, a further comparison of the rules may be done on the basis of a collection of large projects. A major deficiency in the existing measures of project network complexity is that there is a shortage of well-designed experiments to compare and verify the effectiveness of the measures. Also, there is usually no guideline as to whether a complexity measure should be used as an ordinal or a cardinal measure, as is illustrated in the following example. Table 5 presents a sample project for illustrating the network complexity computation. Using the formulation for network complexity presented by Badiru (1996), we obtain

$$p = 1 \quad L = 6 \quad d = 6.33$$
$$i = 1 \sum t_i = i = 1 \sum t_x x_{i1} = 22.5, \quad i = 1 \sum t_i x_{i2} = 6.3$$
$$\lambda = \frac{1}{6.33}\left[\left(\frac{6-1}{6}\right)(13.5) + \left(\frac{22.58}{5} + \frac{6.25}{2}\right)\right]$$
$$= 2.99$$

If the preceding complexity measure is to be used as an ordinal measure, then it must be used to compare and rank alternate project networks. For example, when planning a project, one may use the complexity measure to indicate the degree of simplification achieved in each iteration of the project life cycle. Similarly, when evaluating project options, one may use the ordinal complexity measure to determine which network option will be easiest to manage. If the complexity measure is to be used as a cardinal (absolute) measure, then a benchmark value must be developed. In other words, control limits will be needed to indicate when a project network can be classified as simple, medium, or complex.

RESOURCE ALLOCATION IN PERT NETWORKS

Basic CPM and PERT approaches assume unlimited resource availability in project network analysis. In realistic projects, both the time and resource requirements of activities should be considered in developing network schedules. Projects are subject to three major constraints of time limitations, resource constraints, and performance requirements. As these constraints are difficult to satisfy simultaneously, trade-offs must be made. In some cases, the smaller the resource base, the longer the project schedule and the lower the quality of work. Resource allocation facilitates the transition of a project from one state to another state. Given that the progress of a project is in an initial state defined as S_i and a future state is defined as S_f, then three possible changes can occur.

1. Further progress may be achieved in moving from the initial state to the future state (i.e., $S_f > S_i$).
2. Progress may be stagnant between the initial state and the future state (i.e., $S_f = S_i$).
3. Progress may regress from the initial state to the future state (i.e., $S_f < S_i$).

Resource allocation strategies must be developed to determine which is the next desired state of the project, when the next state is expected to be reached, and how to move towards that next state. Resource availability and criticality will determine how activities should be assigned to resources to facilitate progress of a project from one state to another. Graphical tools can provide guidance for resource allocation strategies. Critical path method (CPM), program evaluation and review technique (PERT), and precedence diagramming method (PDM) are examples of simple graphical tools based on activity scheduling. There is a need for similarly simple tools for resource allocation planning, scheduling, tracking, and control. The Critical Resource Diagramming (CRD) method developed by Badiru (1995) represents such a simple tool for resource scheduling.

CRITICAL RESOURCE DIAGRAMMING

Badiru (2) presents a simple extension of the PERT/CPM diagram for resource scheduling purposes. The extension, called critical resource diagram (CRD), is a graphical tool that brings the advantages of CPM diagram to resource scheduling. With its focus on resource scheduling, a CRD takes a reverse view to activity scheduling in CPM. The advantages of CRD include simplified resource tracking and control, better job distribution, better information to avoid resource conflicts, and better resource leveling. In this section, we illustrate how critical resource diagramming can be used to develop strategies for assigning activities to resources or assigning resources to activities in engineering projects.

RESOURCE SCHEDULING CONSTRAINTS

Resource management is a complex task that is subject to several limiting factors including the following examples:

- Resource interdependencies
- Conflicting resource priorities
- Mutual exclusivity of resources
- Limitations on resource substitutions
- Variable levels of resource availability
- Limitations on partial resource allocation
- Limitations on duration of resource availability

Resources are needed by activities, activities produce products, products constitute projects, and projects make up organizations. Thus, resource management can be viewed as a basic component of the management of any organization. It is logical to expect different resource types to exhibit different levels of criticality in a resource allocation problem. For example, some resources may be very expensive. Some resources may possess special skills. Some may have very limited supply. The relative importance of different resource types should be considered when carrying out resource allocation in activity scheduling. The critical resource diagram helps in representing resource criticality.

RESOURCE PROFILING

Resource profiling involves the development of graphical representations to convey information about resource availability, utilization, and assignment. Resource loading and resource leveling graphs are two popular tools for profiling resources. Resource idleness graph and critical resource diagram are two additional tools that can effectively convey resource information.

Resource Loading

Resource loading refers to the allocation of resources to work packages in a project network. A resource loading graph presents a graphical representation of resource allocation over time. Figure 13 shows an example of a resource loading graph. A resource loading graph may be drawn for the different resource types involved in a project.

The graph provides information useful for resource planning and budgeting purposes. In addition to resource units committed to activities, the graph may also be drawn for other tangible and intangible resources of an organization. For example, a variation of the graph may be used to present information about the depletion rate of the budget available for a project. If drawn for multiple resources, it can help identify potential areas of resource conflicts. For situations where a single resource unit is assigned to multiple tasks, a variation of the resource loading graph can be developed to show the level of load (responsibilities) assigned to the resource over time.

RESOURCE LEVELING

Resource leveling refers to the process of reducing the period-to-period fluctuations in a resource loading graph. If resource fluctuations are beyond acceptable limits, actions are taken to move activities or resources around in order to level out the resource loading graph. Proper resource planning will facilitate a reasonably stable level of the work force. Advantages of resource leveling include simplified resource tracking and control, lower cost or resource management, and improved opportunity for learning. Acceptable resource leveling is typically achieved at the expense of longer project duration or higher project cost. Figure 3 shows a somewhat leveled resource loading.

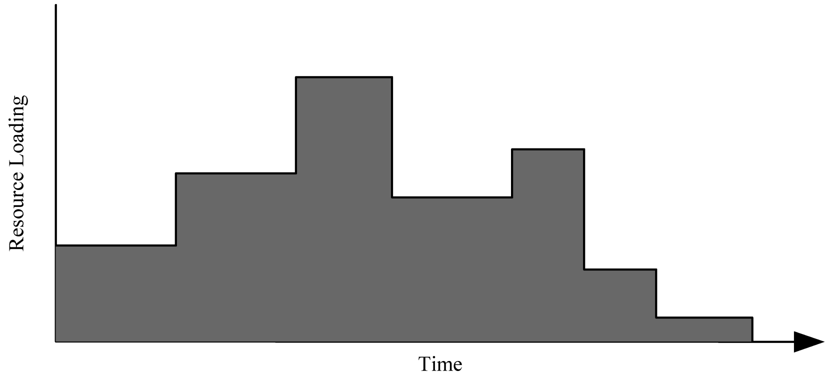

Figure 13. Resource Loading Graph

It should be noted that not all of the resource fluctuations in a loading graph can be eliminated. Resource leveling attempts to minimize fluctuations in resource loading by shifting activities within their available slacks. One heuristic procedure for leveling resources, known as the Burgess's Method (1), is based on the technique of minimizing the sum of squares of the resource requirements in each period.

RESOURCE IDLENESS

A resource idleness graph is similar to a resource loading graph except that it is drawn for the number of unallocated resource units over time. The area covered by the resource idleness graph may be used as a measure of the effectiveness of the scheduling strategy employed for a project. Suppose two scheduling strategies yield the same project duration and a measure of the resource utilization under each strategy is desired as a means to compare the strategies. Figure 4 shows two hypothetical resource idleness graphs for the alternate strategies. The areas are computed as follows:

$$\begin{aligned} \text{Area A} &= 6(5) + 10(5) + 7(8) + 15(6) + 5(16) \\ &= 306 \text{ resource-units-time.} \\ \text{Area B} &= 5(6) + 10(9) + 3(5) + 6(5) + 3(3) = 12(12) \\ &= 318 \text{ resource-units-time.} \end{aligned}$$

Because Area A is less than Area B, it is concluded that Strategy A is more effective for resource utilization than Strategy B. Similar measures can be developed for multiple resources. However, for multiple resources, the different resource units must all be scaled to dimensionless quantities before computing the areas bounded by the resource idleness graphs.

CRD NETWORK CONSTRUCTION

Figure 16 shows an example of a critical resource diagram for a small project requiring six different resource types. Each node identification, RES *j*, refers to a task responsibility for resource type *j*.

In a CRD, a node is used to represent each resource unit. The interrelationships between resource units are indicated by arrows. The arrows are referred to as *resource-relationship (R-R) arrows*. For example, if the job of *Resource 1* must precede the job of *Resource 2*, then an arrow is drawn from the node for resource 1 to the node for resource 2. Task durations are included in a CRD to provide further details about resource relationships. Unlike activity diagrams, a resource unit may appear at more than one location in a CRD provided that there are no time or task conflicts. Such multiple locations indicate the number of different jobs for which the resource is responsible. This information may be useful for task distribution and resource leveling purposes. In Figure 16, Resource type 1 (RES 1) and Resource type 4 (RES 4) appear at two different nodes, indicating that each is responsible for two different jobs within the same work scenario. However, appropriate precedence constraints may be attached to the nodes associated with the same resource unit if the resource cannot perform more than one task at the same time. This is illustrated in Figure 17.

CRD NETWORK ANALYSIS

The same forward and backward computations used in CPM are applicable to a CRD diagram. However, the interpretation of the critical path may be different since a single resource may appear at multiple nodes. Figure 18 presents a computational analysis of the CRD network in Fig. 16. Task durations (days) are given below the resource identifications. Earliest and latest times are computed and appended to each resource node in the same manner as in CPM analysis. RES 1, RES 2, RES 5, and RES 6 form the critical resource path. These resources have no slack times with respect to the completion of the given project. Note that only one of the two tasks of RES 1 is on the critical resource path.

Thus, RES 1 has a slack time for performing one job, while it has no slack time for performing the other. None of the two tasks of RES 4 is on the critical resource path. For RES 3, the task duration is specified as zero. Despite this favorable task duration, RES 3 may turn out to be a bottleneck resource. RES 3 may be a senior manager whose task is that of signing a work order. But if he or she is not available to sign at the appropriate time, then the tasks of several other resources may be adversely affected. A major benefit of a CRD is that both the senior-level and lower-level resources can be modeled in the resource planning network.

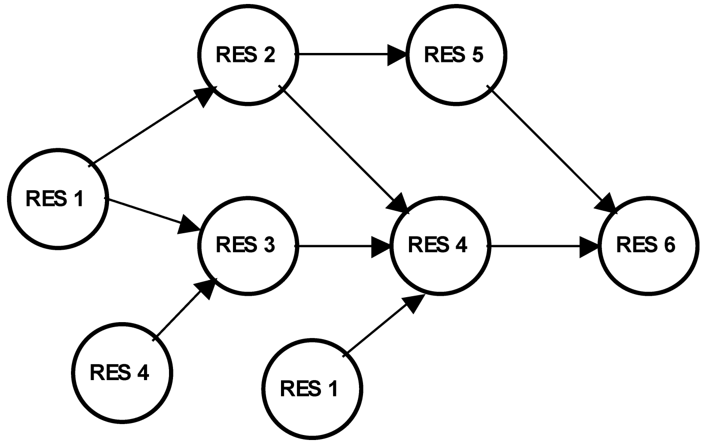

Figure 16. Basic critical resource diagram.

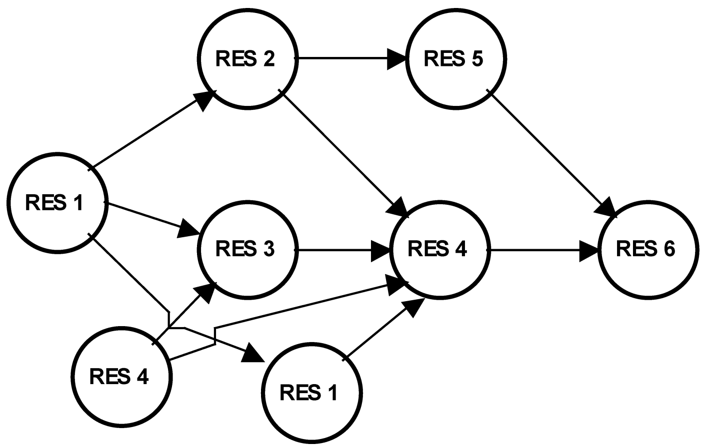

Figure 17. CRD with singular resource precedence constraint

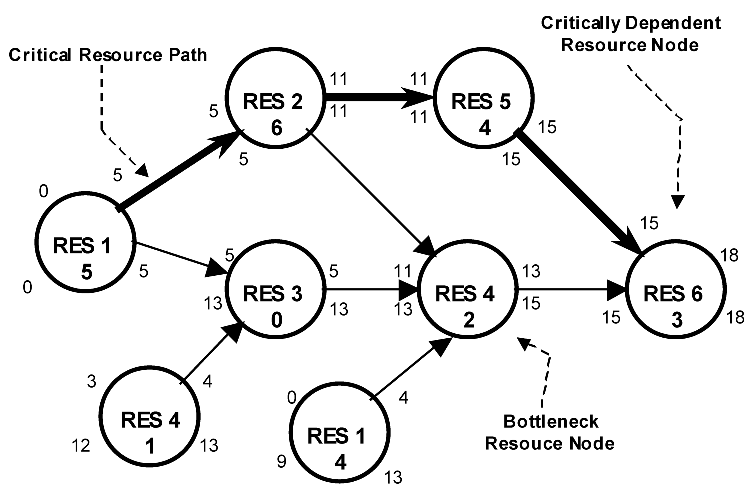

Figure 18. CRD network analysis

CRD Node Classifications

A *bottleneck* resource node is defined as a node at which two or more arrows merge. In Figure 18, RES 3, RES 4, and RES 6 have bottleneck resource nodes. The tasks to which bottleneck resources are assigned should be expedited in order to avoid delaying dependent resources. A *dependent* resource node is a node whose job depends on the job of immediate preceding nodes. A *critically dependent* resource node is defined as a node on the critical resource path at which several arrows merge. In Figure 18, RES 6 is both a critically dependent resource node and a bottleneck resource node. As a scheduling heuristic, it is recommended that activities that require bottleneck resources be scheduled as early as possible. A *burst* resource node is defined as a resource node from which two or more arrows emanate. Like bottleneck resource nodes, burst resource nodes should be expedited since their delay will affect several following resource nodes.

RESOURCE SCHEDULE CHART

The critical resource diagram has the advantage that it can be used to model partial assignment of resource units across multiple tasks in single or multiple engineering projects. A companion chart for this purpose is the resource schedule (RS) chart. Figure 19 shows an example of an RS chart based on the earliest times computed in Figure 18. A horizontal bar is drawn for each resource unit or resource type. The starting point and the length of each resource bar indicate the interval of work for the resource. Note that the two jobs of RES 1 overlap over a four-day time period. By comparison, the two jobs of RES 4 are separated by a period of six days. If RES 4 is not to be idle over those six days, tasks that "fill-in" must be assigned to it. For resource jobs that overlap, care must be taken to ensure that the resources do not need the same tools (e.g., equipment, computers, lathe, etc.) at the same time. If a resource unit is found to have several jobs overlapping over an extensive period of time, then a task reassignment may be necessary to offer some relief for the resource.

The RS chart is useful for a graphical representation of the utilization of resources. Although similar information can be obtained from a conventional resource loading graph, the RS chart gives a clearer picture of where and when resource commitments overlap. It also shows areas where multiple resources are working concurrently. Note that activity slacks do not appear in the resource schedule chart. This is an important difference from conventional Gantt charts, in which activity slack times can be identified. Resources do not have slack times in the traditional sense of "slack" because resources are assumed to be fully engaged throughout the project. Resource units move on to other activities as soon as one activity is completed. If it is desired to show the idle time of an activity on the resource schedule chart, a "delay activity" can be created. The resource can then be assigned to that delay activity for the period of idleness.

CRD AND WORK RATE ANALYSIS

When resources work concurrently at different work rates, the amount of work accomplished by each may be computed by a procedure presented by Badiru (2). The critical resource diagram and the resource schedule chart provide information to identify when, where, and which resources work concurrently. The general relationship between work, work rate, and time can be expressed as

$$w = rt$$

where w = amount of actual work accomplished This is expressed in appropriate units, such as miles of road completed, lines of computer code typed, gallons of oil spill cleaned, units of widgets produced, or surface area painted; r = rate at which the work is accomplished; and t = total time required to accomplish the work. It should be noted that work rate can change due to the effects of learning curves. In the discussions that follow, it is assumed that work rates remain constant for at least the duration of the work being analyzed.

Work is defined as a physical measure of accomplishment with uniform destiny (i.e., homogeneous). For example, a computer programming task may be said to be homogeneous if one line of computer code is as complex and desirable as any other line of code in the program. Similarly, cleaning one gallon of oil spill is as good as cleaning any other gallon of oil spill within the same work environment. The production of one unit of a product is identical to the production of any other unit of the product. If uniform work density cannot be assumed for the particular work being analyzed, then the relationship presented above will need to be modified. If the total work to be accomplished is defined as one whole unit, then the tabulated relationship in Table 6 will be applicable for the case of a single resource performing the work, where $1/x$ is the amount of work accomplished per unit time. For a single resource to perform the whole unit of work, we must have the following:

$$(1/x)(t) = 1.0$$

That means that magnitude of x must equal the magnitude of t. For example, if Machine A is to complete one work unit in 30 minutes, it must work at the rate of 1/30 of work per unit time. If the magnitude of x is greater then the magnitude of t, then only a fraction of the required work will be performed. The information about the proportion of work completed may be useful for resource planning and productivity measurement purposes. In the case of multiple resources performing the work simultaneously, the work relationship is as presented in Table 7. For multiple resources, we have the following expression:

$$i = 1 \sum r_i t_i = 1.0$$

where n = number of different resource types; r_i = work rate of resource type i; t_i = work time of resource type i. The expression indicates that even though the multiple resources may work at different rates, the sum of the total work they accomplished together must equal the required whole unit. For partial completion of work, the expression becomes

$$i = 1 \sum r_i t_i = p,$$

where p is the proportion of the required work actually completed.

COMPUTATIONAL EXAMPLES

Suppose that RES 1, working alone, can complete a job in 50 minutes. After RES1 has been working on the job for 10 minutes, RES 2 was assigned to help RES 1 in completing the job. Both resources working together finished the remaining work in 15 minutes. It is desired to determine the work rate of RES 2.

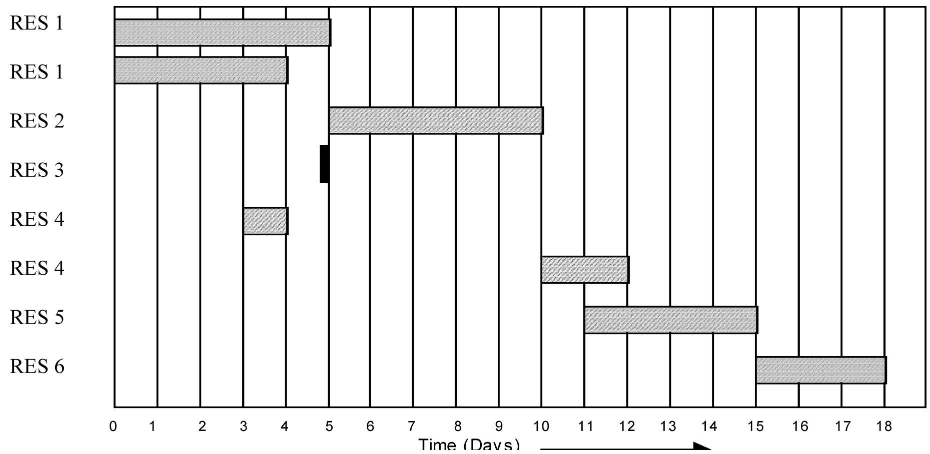

Figure 19. Resource schedule chart based on earliest start

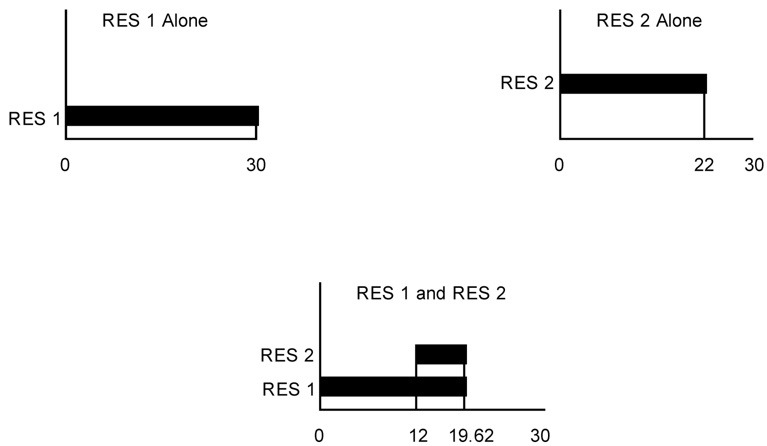

Figure 20. Resource schedule chart for RES 1 and RES 2

The amount of work to be done is 1.0 whole unit. The work rate of RES1 is 1/50 of work per unit time. Therefore, the amount of work completed by RES 1 in the 10 minutes it worked alone is (1/50)(10) = 1/5 of the required work. This may also be expressed in terms of percent completion or earned value using C/SCSC (cost-schedule control systems criteria). The remaining work to be done is 4/5 of the total work. The two resources working together for 15 minutes yield the results shown in Table 8.

Thus, we have $15/50 + 15(R_2) = 4/5$, which yields $r_2 = 1/30$ for the work rate of RES 2. This means that RES 2, working alone, could perform the job in 30 minutes. In this example, it is assumed that both resources produce identical quality of work. If quality levels are not identical for multiple resources, then the work rates may be adjusted to account for the different quality levels or a quality factor may be introduced into the analysis. The relative costs of the different resource types needed to perform the required work may be incorporated into the analysis as shown in Table 9.

As another example, suppose that the work rate of RES 1 is such that it can perform a certain task in 30 days. It is desired to add RES 2 to the task so that the completion time of the task could be reduced. The work rate of RES 2 is such that it can perform the same task alone in 22 days. If RES 1 has already worked 12 days on the task before RES2 comes in, find the completion time of the task. It is assumed that RES 1 starts the task at time 0.

As usual, the amount of work to be done is 1.0 whole unit (i.e., the full task). The work rate of RES 1 is 1/30 of the task per unit time and the work rate of RES 2 is 1/22 of the task per unit time. The amount of work completed by RES 1 in the 12 days it worked alone is (1/30)(12) = 2/5 (or 40%) of the required work. Therefore, the remaining work to be done is 3/5 (or 60%) of the full task. Let T be the time for which both resources work together. The two resources working together to complete the task yield the entries in Table 10.

Thus, we have $T/30 + T/22 = 3/5$, which yields $T = 7.62$ days. Consequently, the completion time of the task is $(12 + T) = 19.62$ days from time zero. The results of this example are summarized in the resource schedule charts in Figure 20. It is assumed that both resources produce identical quality of work and that the respective work rates remain consistent. As mentioned earlier, the respective costs of the different types may be incorporated into the work rate analysis.

CONCLUSION

The CRD, Resource Work Rates, and RS chart are simple extensions of PERT/CPM tools in project management. They are simple to use and they convey resource information quickly. They can be used to complement existing resource management tools in engineering projects. For example, resource-dependent task durations and resource cost can be incorporated into the CRD and RS procedures to enhance their utility for resource management decisions.

BIBLIOGRAPHY

1. A. B. Badiru, *Project Management in Manufacturing and High Technology Operations*, 2nd ed., New York; Wiley, 1996.
2. Badiru, Adedeji B., "Activity-Resource Assignments Using Critical Resource Diagramming," *Project Management Journal*, Vol.**24**, No. 3,Sept. 1993, pp. 15–21.

The views expressed in this article are those of the authors and do not reflect the official policy or position of Air Force Institute of Technology, the United States Air Force, Department of Defense, or United States government.

ADEDEJI BADIRU
Air Force Institute of Technology, Department of Systems and Engineering Management, Wright Patterson Air Force Base, OH