N95- 20748

SCHEDULING: A GUIDE FOR PROGRAM MANAGERS

Defense Systems Management College

For 4,500 years after the building of the great pyramid at Giza, nothing surfaced as a better way to develop a schedule than that used for the pyramid. Then, early in this century, Henry L. Gantt, a pioneer in the field of scientific management, unveiled his bar chart technique. From that time forward, program planning and scheduling have consisted of a list of activities with start and completion dates.

Gantt's "daily balance chart" was a significant breakthrough. Suddenly, you could see at a glance the overall program schedule and the start and stop times of the program's individual components (Figure 1). A Gantt chart can be superimposed with ease on a calendar. Then, by shading in each bar as progress is made, a manager can easily measure actual progress against the schedule.

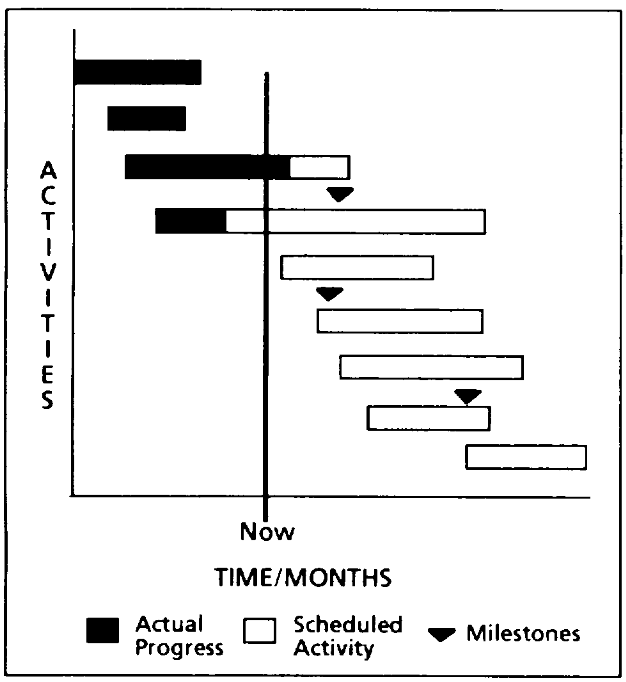

Figure 1. Gantt (Bar) Chart

For 50 years, bar charting was the best way to schedule activities. There were good reasons for it. Bar charts communicate, are easy to prepare and use, show key activities with specific start and completion times (scheduled and actual), relate schedules to calendar dates, and display days or weeks from program start to completion.

Figure 1 shows that milestones may be added to bar charts to display significant events. In fact, it may be appropriate to show a number of milestones associated with a single bar. Because they communicate so well and so quickly, bar charts are still used to plan and monitor progress against the plan. Upper management, in particular, appreciates this capability.

A shortcoming of bar charts is the limited information they portray. Dependency and other interrelationships among activities are difficult to display because bar charts handle a limited degree of complexity. Figure 2 shows how a bar chart can provide a clear, but limited, picture of dependencies and progress. The bar chart can present a history of changes and rescheduling occurring on a program; however, this is done more frequently on milestone charts, which will be discussed later.

As a scheduling tool, the bar chart is simple, communicates well, and displays calendar and significant program dates. Because of its simplicity and ease of interpretation, the bar chart is a particularly good tool for communicating important information to upper management; however, it is limited in the degree of detail and the interrelationships it can portray.

C-2

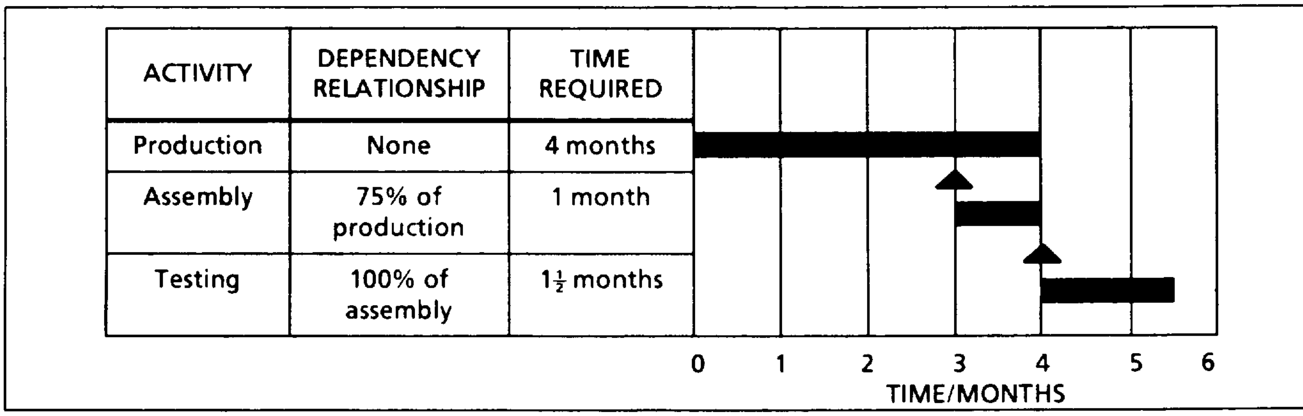

Figure 2. Gantt Chart Showing Dependency

Table 1 summarizes the strengths and weaknesses of Gantt charts. The weaknesses apply to planning, estimating and reporting as well as to bar charts.

Milestone Scheduling

Milestone scheduling is a popular technique being used in Department of Defense (DoD) program management offices. The milestone chart is probably the most commonly used chart at the Air Force Systems Command, Electronic Systems Division (ESD), and many other DoD organizations. The technique is relatively simple. Milestone charts-

Table 1. Gantt Technique: Strengths and Weaknesses

CRITERIA	STRENGTHS	WEAKNESSES
1. Accuracy	Good in repetitive work. Time estimates are likely to be good and production is easy to count.	In non-repetitive work, accuracy of estimate of task completion percentage is subject to error.
2. Reliability	Simplicity of technique helps program manager to set up a consistent progress reporting system.	In repetitive work, production can be "doctored." Large non-repetitive programs involve many different progress estimators, which tends to affect consistency.
3. Simplicity	Easy to understand, accept and implement.	Requires good time estimates, or standards, which are not simple to develop.
4. Universality of program coverage	Effective at work-center levels. Cover a specific phase of program life cycle well.	Not effective for a large,complex program unless program is computer-based.
5. Decision analysis	N/A	No capability to stimulate alternatives.
6. Forecasting	Clearly shows ability to meet schedules in repetitive work.	Does not show ability to meet schedule if many interrelated tasks are involved.
7. Updating	Easy to update if program is static.	N/A
8. Flexibility	N/A	Much chart reconstruction needed to show program changes.
9. Cost	Data gathering and display are relatively inexpensive.	Frequent program changes cause costly redrafting of charts.

are event oriented, bar charts are activity oriented. For a particular program, a set of key events, or milestones, is selected. A milestone is a scheduled event that will occur when a particular activity is started or completed. Milestones are selected from the program management plan. By reviewing the status of key milestones, one can assess quickly the overall program status.

Although milestone charts can present more information than bar charts, they share one important drawback: they invite surprises. A surprise can occur when the number of displayed milestones is too limited or when interdependencies are not portrayed. The result may be that the program manager does not know the status of a key event until it occurs, or until it fails to occur when scheduled. A well conceived milestone status report can provide early warning of a potential problem, and early problem recognition is a key to successful program management.

The milestone scheduling technique uses a symbology consisting of arrows and diamonds, or similar designators, to show originally planned event dates and the changed dates. Figure 3 shows the symbols and their meanings used by the Air Force at ESD. Any symbol can be used; mechanics are not as important as principles.

Arrows are used to show rescheduled events; diamonds indicate the originally planned schedule. As a result, the milestone schedule allows us to improve on the Gantt chart by retaining the baseline dates, while incorporating changes in planned future events. Figure 4 is an example of a milestone chart.

The milestone chart records the manager's assessment. For example, a manager might-reasonably predict that a one-month slip in the start of software development will probably result in a several-month slip in completing the engineering development phase. The milestone chart does not provide the assessment – the manager's experience does. This is the key to understanding the use of milestone charts. Unless the activity and interrelationships of milestones are understood, the chart tells only what has happened and what is yet to happen. However, by coupling historical information with the manager's experience and knowledge, more accurate predictions can be made.

The milestone scheduling technique shows what is scheduled, what has happened and changes in plans. The technique is not as useful for forecasting schedule changes as are the network and line-of-balance techniques discussed later.

Like the bar chart, the milestone chart is an effective method of communication. The symbology is relatively standard and simple to use. The chart presents actual progress against a baseline plan and displays changes in plans. The mechanics of constructing a milestone chart are relatively easy. Many defense contractors use milestone charts extensively in DoD program management.

As with simple bar charts, a major weakness of milestone charts is their inability to show interdependencies and interaction among activities. A potential problem can result if a program manager focuses on a relatively simple milestone format. He may lose sight of the complexity of the relationships among various program tasks.

Although milestone charts are used on complex programs, they are usually the product of network analysis. Milestone chart preparation is relatively simple, but developing and analyzing the information going into the charts can be time consuming. A controlled flow of accurate, timely and appropriate information is important.

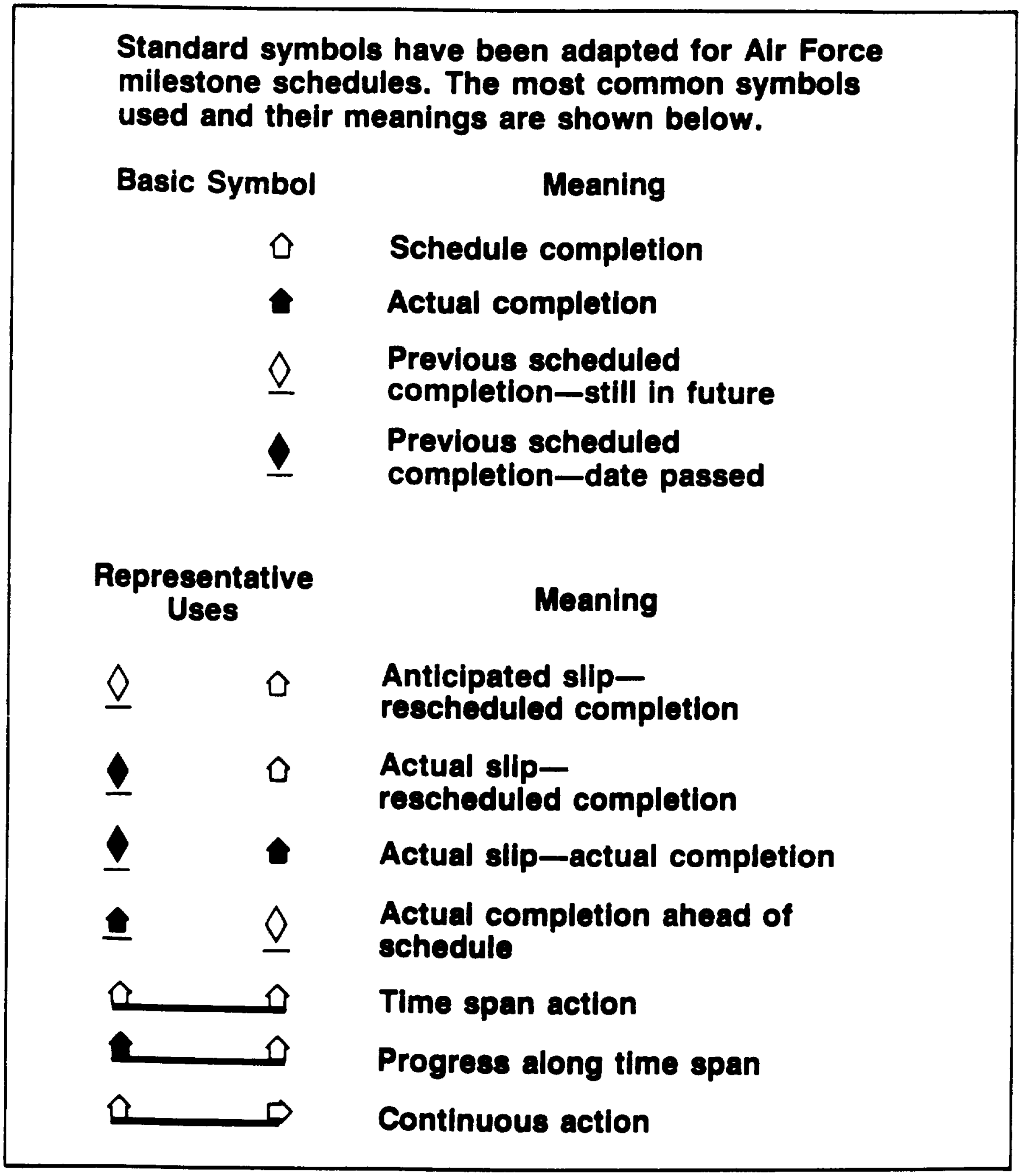

Figure 3. Milestone Symbols

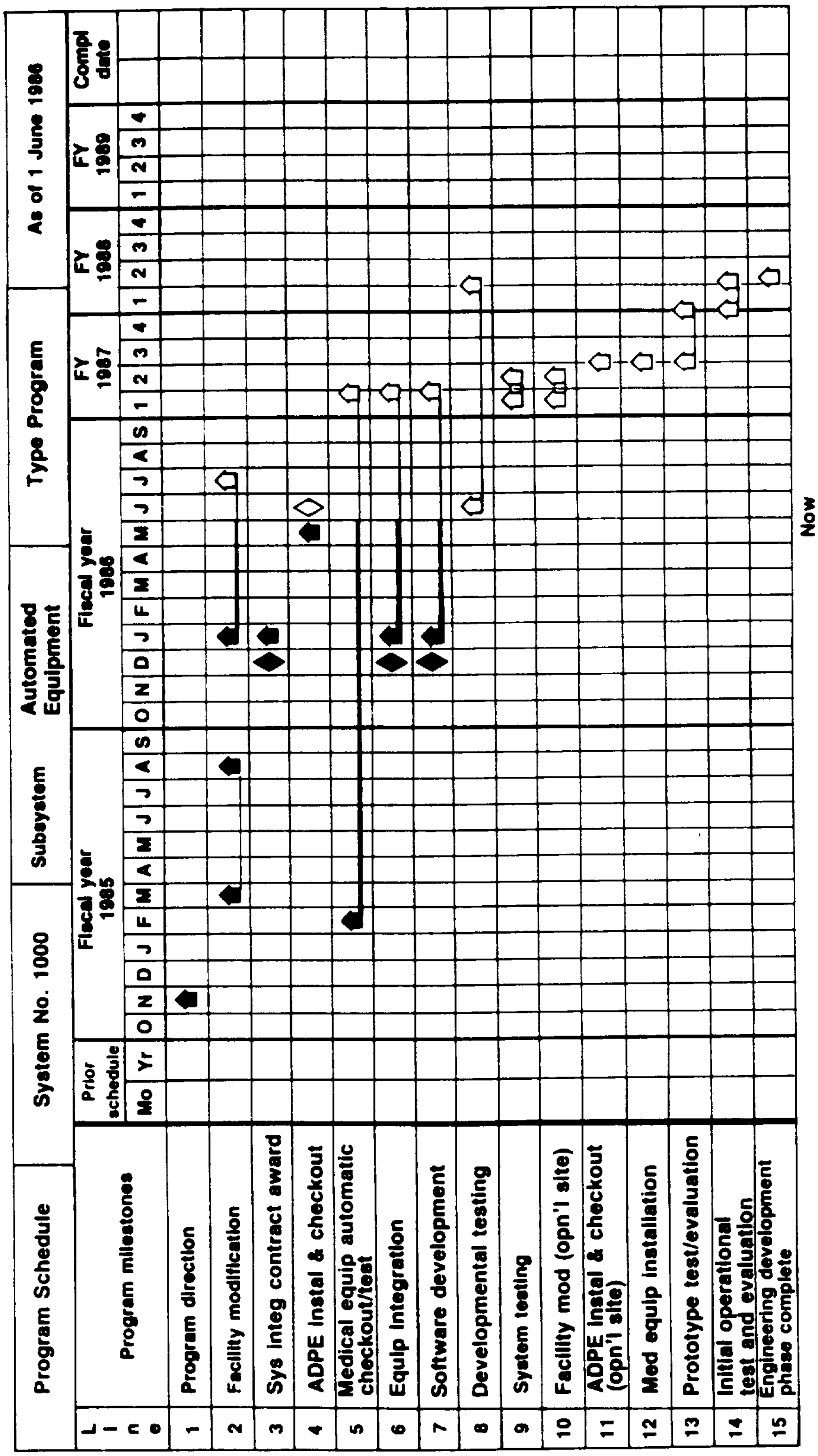

Figure 4. Milestone Chart

Milestone charts represent a simple and effective means to display actual versus planned progress of a program, and to show schedule changes that have occurred. These charts emphasize start and completion dates, rather than the activities that take place between these dates. Milestone charts display very limited dependency information and they may present program status in a deceptively simple manner.

Network Scheduling

Shortcomings of bar and milestone charts gave rise in the 1950s to network scheduling. The network techniques provided a may to graphically display information for program managers that could not be presented with bars or milestones.

First, a program, is separated into activities. Each activity is based on a particular undertaking and each is defined by a distinct start and completion point. Network scheduling provides a method for finding the longest time-consuming path. This gives the manager two important tools. First, the project manager is able to more accurately estimate the total time from program start to completion. Second, by being able to identify items on the critical (or longest) path as opposed to tasks less critical, the project manager is able to analyze problems as they arise.

Program Evaluation and Review Technique

The Program Evaluation and Review Technique (PERT) was developed in 1957 under the sponsorship of the U.S. Navy Special Projects Office. The Navy wanted PERT as a management tool for scheduling and controlling the Polaris missile program, a program which involved 250 prime contractors and more than 9,000 subcontractors. The project manager had to keep track of hundreds of thousands of tasks.

PERT helps answer such questions as:

- When is each segment of the program scheduled to begin and end?
- Considering all of the program segments, which segments must be finished on time- to avoid missing the scheduled completion date?
- Can resources be shifted to critical parts of the program (those that must be completed on time) from non-critical parts (those that can be delayed) without affecting the overall scheduled completion date for the program?
- Among the myriad program tasks, where should management efforts be concentrated at any particular time?

Since most activities in a PERT network take a long time to accomplish, time is usually expressed in days or weeks. The expected time for an activity is often described by a probability distribution rather than a single estimate because of the uncertainty associated with programs that have not been done the same way before. The characteristics of the distribution used to express the variation in time are:

- A small probability of reaching the most optimistic time (shortest time), time a.
- A small probability of reaching the most pessimistic time (longest time), time b.
- The one most likely time which would fall between the two extremes above, time m.
- The ability to measure uncertainty in estimating.

Because it has all four attributes, the beta distribution was chosen for determining the expected time. Figure 5 shows a beta distribution with the time designations under the curve.

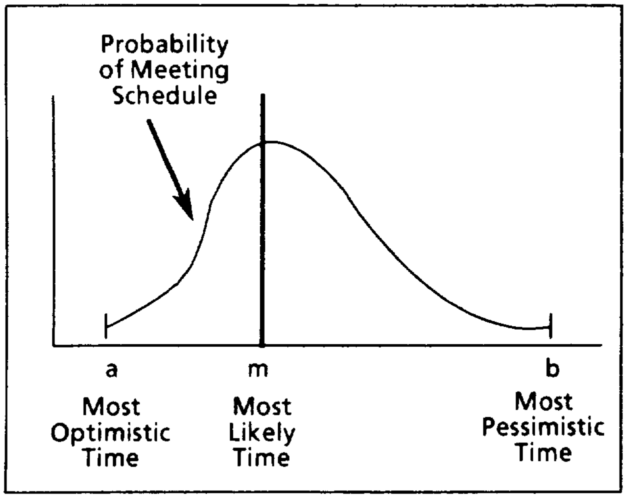

Figure 5. Beta Distribution with Symbols Indicating Time Estimates

The three time estimates shown may be combined into a single workable time value. By using the following weighted average formula, the expected time for an activity can be found:

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

where *t* is the expected time. According to MacCrimmon and Ryavec, the error in the answer using this formula is small enough to make it satisfactory to use in most cases.

According to Hugh McCullough, former Polaris project business manager, PERT had a disciplinary effect. The Polaris project had a 20,000-event network and the application of PERT is credited with saving two years in bringing the Polaris missile submarine to combat readiness.

As PERT's popularity grew, consulting firms specializing in network scheduling sprang up overnight. The DoD established a PERT Operation and Training Center (POTC, nicknamed "Potsie") in Washington, DC. During the next few years, PERT became widely used throughout DoD systems acquisition programs.

A few years later, the use of PERT declined sharply, and by the 1970s, it was rarely employed in defense systems programs. Why did PERT go through such a rapid rise and abrupt decline? In essence, the predictable happened. When PERT was combined with cost data or other non-scheduling aspects of program management, it became cumbersome. Eventually, use of such an embellished technique resulted in the tail wagging the management dog. The DoD program managers and defense contractors spent immense amounts of time collecting and entering detailed data. Soon, the cost of maintaining PERT systems far outweighed the benefits they offered the program manager.

An article published in the DSMC magazine *Program Manager* reveals how little network scheduling is being used by DoD acquisition program managers. Seventy percent of the major defense programs surveyed do not use a network scheduling system. However, the remaining 30 percent employ a network technique. (Ingalls)

DoD and the defense industry returned to simpler techniques like milestone and bar charts, probably an overreaction. The private sector continues to make good use of network scheduling in varied efforts like new product development, construction, and major maintenance activities. This resurgence is due in part to the development of PERT and other networking software programs that run on microcomputers.

In spite of misuses that have occurred in PERT applications, the technique enables the manager to visualize the entire program, see interrelationships and dependencies, and recognize when delays are acceptable. Thus, the manager is better able to assess problems as the program evolves. In order to apply PERT and similar networking techniques, it is important that certain conditions exist:

1. The program must consist of clearly defined activities, each with identifiable start and completion points.

2. The sequence and interrelationships of activities must be determined.

3. When all individual activities are completed, the program is completed.

Many program-oriented industries, like aerospace, construction and shipbuilding, meet these criteria and use a network scheduling technique. Many defense system programs also meet these criteria.

Critical Path Method

Like PERT, the Critical Path Method (CPM) is composed of three phases: planning, scheduling and controlling. This technique, developed in 1957 by J.E. Kelly of Remington-Rand and M.R. Walker of DuPont to aid in scheduling maintenance shutdowns in chemical processing plants, is essentially activity oriented; PERT is event oriented. CPM has enjoyed more use among network techniques than any other technique.

CPM brings the concept of cost more prominently into the scheduling and control process than PERT does. When time can be estimated closely and when labor and material costs can be calculated early in a program, the CPM technique is superior to PERT. When there is much uncertainty and when control over time outweighs control over costs, PERT is a better technique to use. The basic networking principles in PERT and CPM are similar.

In a common version of CPM, two time and cost estimates are given for each activity in the network. These are the normal estimate and the crash estimate (see Figure 6). The normal time estimate approximates the most likely time estimate in PERT. The normal cost is the cost associated with finishing the program in the normal time. The crash time estimate is the time that will be required to finish an activity if a special effort is made to reduce program time to a minimum. The crash cost is the cost associated with performing the effort rapidly, in order to minimize the time to completion.

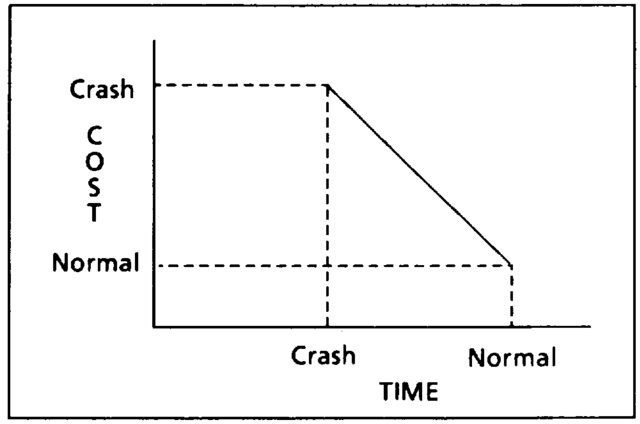

Figure 6. CPM Time-Cost Curve

Developing a Network

Although CPM and PERT are conceptually similar, their symbols and charting techniques vary. The PERT historically has used probability techniques while, CPM generally does not. The following procedure applies to both CPM and PERT.

1. Identify all individual tasks comprising the program.

2. Determine the expected time to complete each activity.

3. Determine precedence and interrelationships of the activities.

4. Develop a network diagram presenting these activities in proper sequence and reflecting any dependency relationships. Activities are indicated by lines; events or milestones are indicated by circles. Dependency or sequencing relationships among activities on separate paths can be shown by dotted lines (dummy activities).

5. Compute and annotate the cumulative time required to reach each milestone along the paths, which will indicate earliest time work can start on the next activity. The final number will indicate the total time required to complete a path.

6. Identify the critical path. This is the sequence of events, or route, taking the longest time to complete.

7. Starting at the program completion milestone on the right side of the diagram, begin working backward and compute the latest time an activity can start without delaying the overall program. For example, if the total program takes 40 weeks and the last activity requires five weeks, the final activity cannot begin later than week 35. The difference between the earliest start time and the latest time before each activity is the slack time, or float. The critical path contains no slack time.

Figure 7 shows a simple network diagram for a computer installation program. This network diagram shows the total program will require 20 days to complete. The critical path is F–G and any delay on this path will delay final completion of program. However, delay of one day can occur along path C–D–E, a delay of five days can occur along path A–B–E, and the final completion date would not be extended.

Critical path programs may be either activity oriented or event oriented. This means that the input and output data are associated with either activities or events. The distinction between the two is not a substantive one with respect to computational practices.

Although it has not been done in the above example involving the installation of a computer, many CPM programs and a few PERT programs require that events (circles on the diagram) be numbered in ascending order. This inhibits the flexibility of the network and causes event-numbering bookkeeping problems, so it is not always done.

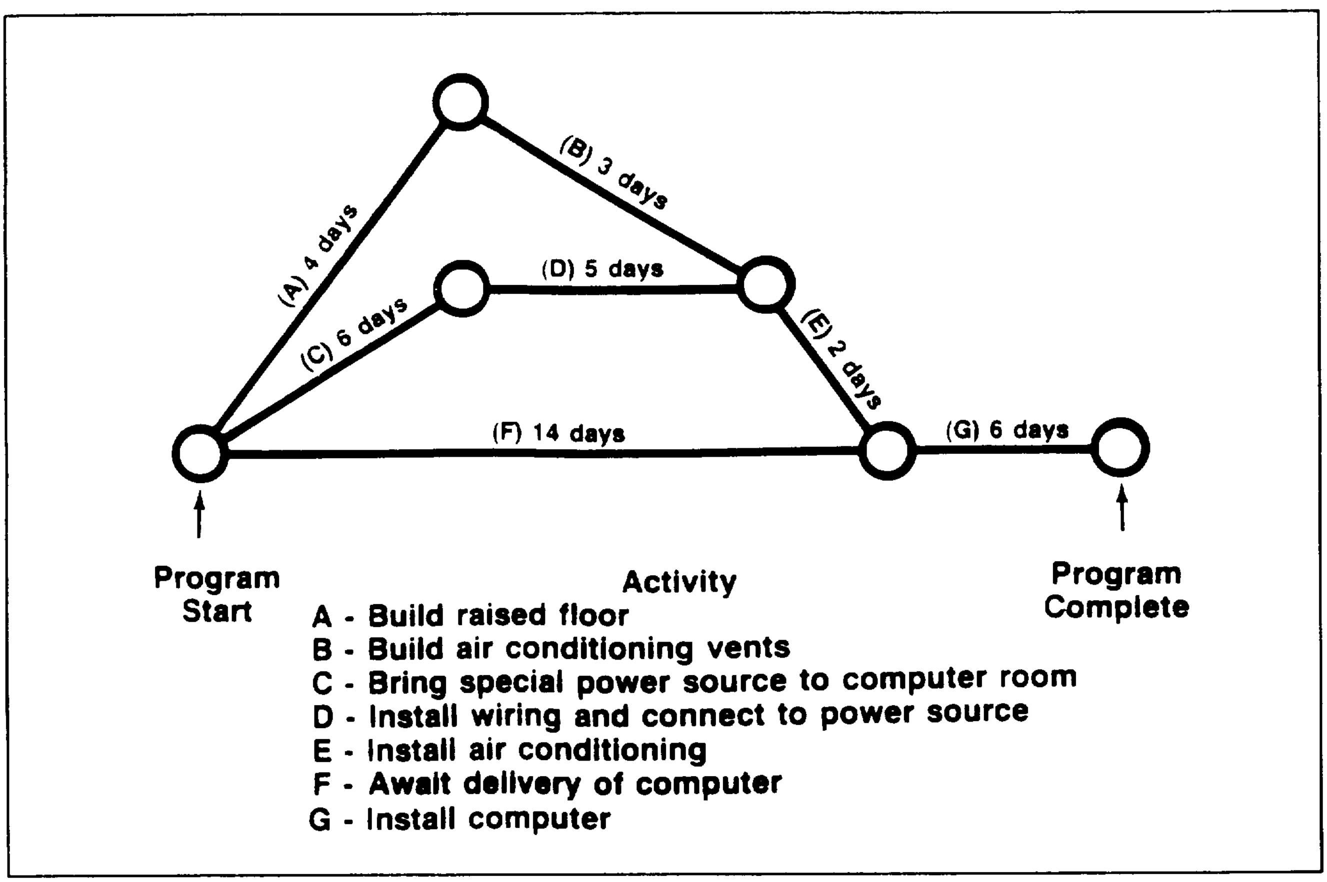

Figure 7. Network Diagram for Computer Installation Program

Converting an Ugly Duckling to a Swan
Although the traditional CPM technique provides useful visibility and clarity about a program, it has shortcomings in that it is difficult to draw the chart to match a time or calendar scale. Although the critical path and slack times can be computed easily, they are not readily apparent. Also, this technique does not display progress to date. Consequently, a simpler technique, sometimes called the Swan Network, is useful.

Let's take an Ugly Duckling Network (Figure 8) and turn it into a Swan Network (Figure 9). Letters in Figure 8 represent activities between the start (S) and completion (C) points. Numbers indicate weeks required for each activity. In Figure 9, activity A is represented by a horizontal bar four weeks long. Constraints are represented by vertical lines or "fences;" for example, the fence after B means B must be completed before E and F can begin (the same as in Figure 8). The result is shown in Figure 9.

What does the Swan network show?

1. The critical path. Time constraints do not have to be calculated. There are, in fact, two critical paths, B–F–I and C–J–K, which are critical because each has a continuous series of activities. There is no slack in either path. Also, the figure has a time scale, which adds greatly to the meaning of the chart.

2. Weeks from start. Scales for "calendar weeks," and "weeks to completion" can be added. In Figure 9, the program is scheduled for completion after 14 weeks.

3. Where there is slack in the schedule and the extent of that slack. For example, there are only two weeks of slack in the A–D–H path. If B–F–I and C–J–K were shortened by more than two weeks, A–D–H would become a critical path. This changing of two critical paths is important when conducting "what if" exercises.

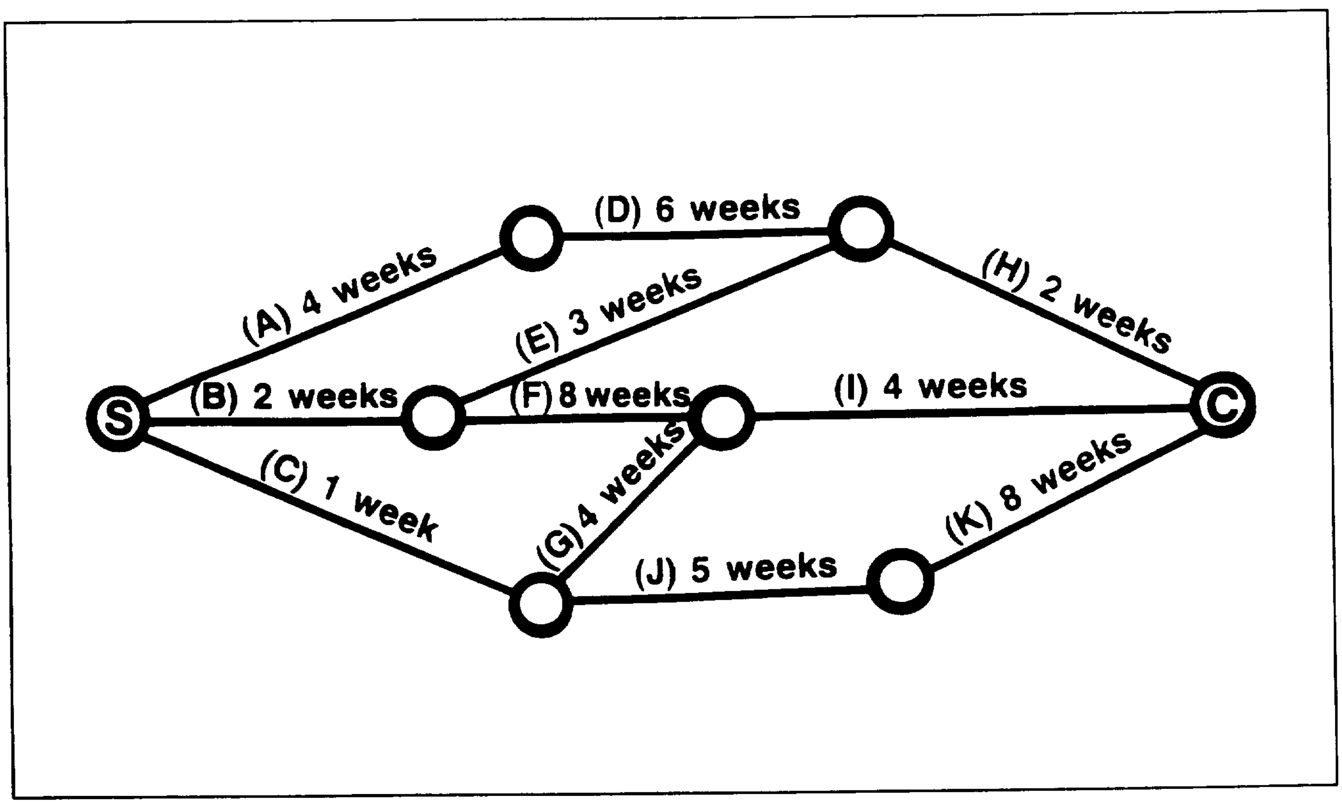

Figure 8. The Ugly Duckling

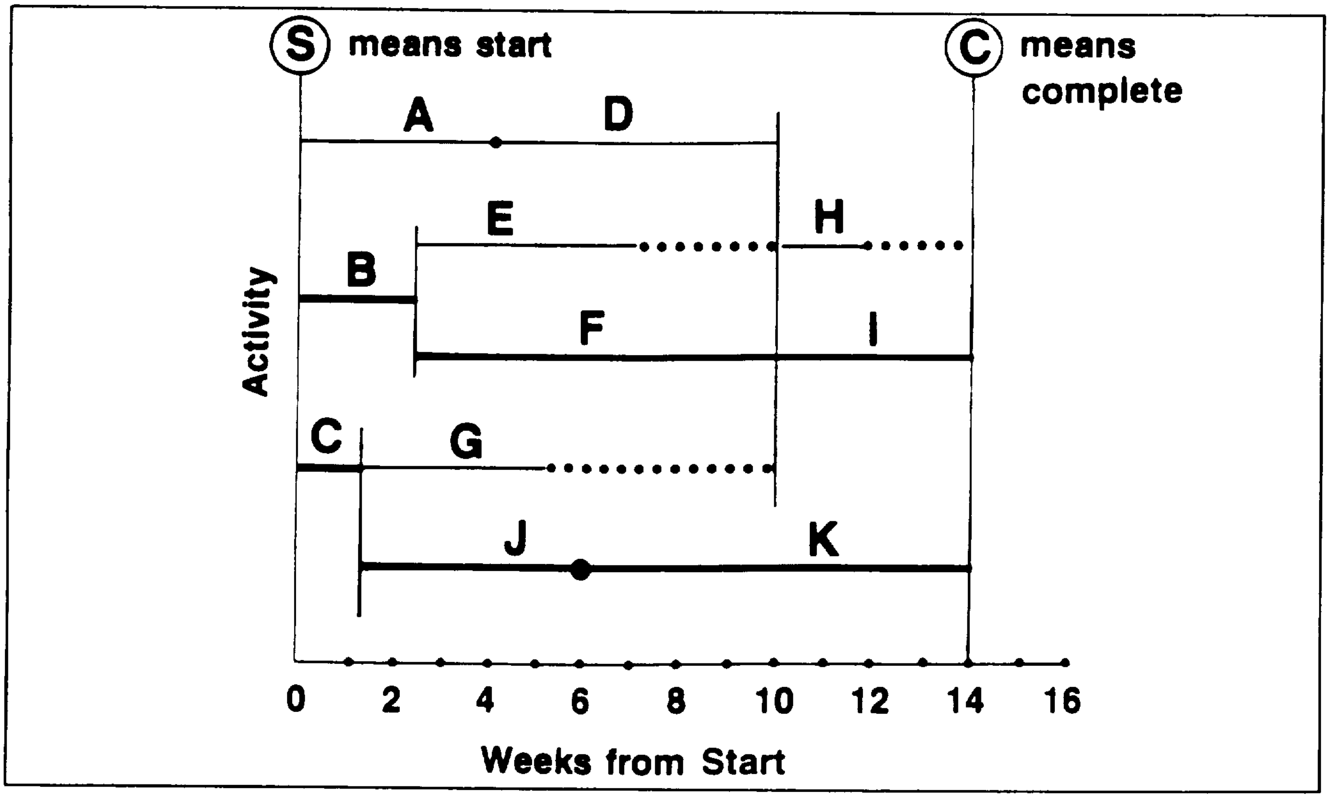

Figure 9. The Swan

The high visibility offered by the Swan network does the following:

- Communicates.
- Motivates. If the level of detail is sufficient, everyone associated with a program activity can see how they affect the schedule, and vice versa.
- Gets top-level attention.
- Makes omissions and errors easier to detect; for example, one company discovered that by using the Swan network, two test activities on the critical path had been omitted. (This was not apparent in the Ugly Duckling network.)
- Shows early start, early finish, late start and late finish.
- Avoids reams of printouts, provided (but not used) for the Ugly Duckling.

The Swan network can be developed in several ways; it can be translated from another network, as shown in the preceding example; it can be developed from a listing of the preceding and following events or activities, as in the network scheduling problem that follows; it can be developed from scratch, with the sequencing and time estimating required in originating any network; and it can be developed from milestones.

A "fence" in the Swan network is usually a milestone like a review or a major event, regardless of how the network is developed.

Actual progress can be shown in the same way as on a Gantt chart. Shading on each bar indicates progress made. A vertical "now" line shows whether activities are on, ahead of, or behind schedule, and by how much.

Now, let's go through an exercise involving network scheduling. Take time to work the problem shown on the following pages.

Network Scheduling Problem

Assume you are program manager. Your objective is to schedule the activities on your program so that one lot of missiles will be assembled and shipped to the test site within 56 days and at the least cost. Use any technique with which you feel comfortable. If you're not comfortable with a particular technique, use the Swan network. Proceed in the following manner, using Tables 2 and 3 provided.

Using lined tablet paper, lay out the normal schedule. This will show the critical path and total number of days required. Identify the initial critical path (number of days). Using Table 3, select the final critical path and related costs that will ensure the completion of the program in 56 days and at the least cost.

It will probably take about 20 minutes for you to determine the solution. The cost for a 56-day program will be in excess of $778,000.

Table 2. Activities, Dependencies, Times and Costs

ACTIVITY[a]	ACTIVITY DEPENDENCY	TIME (WORK DAYS)	NORMAL COST ($000)
1-2 Fab. Initial Guidance Assemblies	None	12	60
1-3 Controls Fabrication[b]	None	24	96
1-4 Rocket Motor Fabrication	None	28	105
1-5 Process Warheads (GFE)	None	16	37.5
2-6 Additional Guidance Assemblies	1-2	20	90
2-3 Guidance Checkout and Sub-Assemblies	1-2	16	120
3-5 G&C Sub-Assemblies	1-3, 2-3	8	70
4-5 Machine Rocket Motors	1-4	12	30
5-6 Missile Assembly	3-5, 4-5, 1-5	6	37.5
6-7 Test	2-6, 5-6	10	62.5
7-8 Ship to Test Site	6-7	8	30

Note: a. Table 3 contains "crash" data.
b. Work on controls fabrication cannot start until after day 2 due to limited resources

Table 3. Activity Time/Cost Relationships

Activity 1-2

Time	Cost
10	62.5

Activity 1-3

Time	Cost
22	105
20	117
18	127

Activity 1-4

Time	Cost
26	110
24	115
20	142

Activity 1-5

Time	Cost
14	60
12	67.5

Activity 2-6

Time	Cost
18	97.5
16	110

Activity 2-3

Time	Cost
14	132.5
12	150

Activity 3-5

Time	Cost
*	*

Activity 4-5

Time	Cost
10	35
8	45

Activity 5-6

Time	Cost
4	60

Activity 6-7

Time	Cost
*	*

Activity 7-8

Time	Cost
6	60

Note: Crash time is in work days and cost is in thousands of dollars.

Crash costs include normal schedule costs. For example, the Activity 1-2 crash cost ($62.5K) includes the normal schedule cost of $60K.

The activity marked * cannot be "crashed."

Network Scheduling When Resources Are Limited

In the previous discussion, the assumption was that a new activity could start as soon as preceding activities were completed, because sufficient resources were available to perform the work. In practice, however, resources to proceed are not always available.

Let's look at an example to illustrate how this network differs in format from previous networks. First, it uses curved lines for activities. This eliminates zero-time activities. Second, it identifies each activity in three ways: by a letter (A), (B), (C), etc.; by estimated duration of activity (in weeks); and by number of people available to work on the activity based on the manager's estimate at the time the network is prepared (see Figure 10).

The network in Figure 10 can be shown in another manner (see Figure 11). In this network, each activity is plotted on the schedule graph with a horizontal time scale. The duration of each activity is represented by the length of that activity's line. The description of each activity represents its letter designation and number of people assigned to that activity at the time indicated (size of work group). The row across the bottom shows total people scheduled to work each week. In this example, 5 to 15 people will be needed, depending on the week being scheduled.

Now, let's suppose only nine people are available to work this nine-week period. What alternatives do we have? We can produce a personnel loading chart by plotting the number of people scheduled to work in any week against time (Figure 12). Then, if we know that only nine people are available, we can see we will not have sufficient workers during the first, fourth and fifth weeks. We will have sufficient workers to perform the work scheduled during the second and sixth week. During the third, seventh, eighth and ninth weeks, we will have a surplus of workers for the work scheduled. The task becomes one of rearranging the schedule so that the peaks and valleys are evened out without scheduling more work than nine people can do. It may not be possible to rearrange the network and still finish the program on time. Under present circumstances, there will not be enough workers to complete the first week's scheduled work on time.

The scheduling problem we are considering can be solved quickly by hand; however, when there are many activities, it becomes very difficult to find the optimum answer, even with a computer. A heuristic program should be used to solve this kind of problem. The heuristic rule is a rule of thumb that works; therefore, collection of rules of thumb is usually known as a heuristic program.

In our example, the heuristic approach is one of finding activities having the most slack and attempting to delay them as long as possible without delaying completion of the entire program. We can delay the start of activity (C) for two weeks and activities (A) and (B) can begin simultaneously without exceeding the limit of nine workers. Continuing to apply this approach, the revised schedule could look like that shown in Figure 13.

When an activity is delayed to improve the schedule, the time which it is delayed is usually shown by a dotted line. At the end of the third week in our example, we had an opportunity to delay activities (D), (G) and (I). We chose to delay (D) and (H) one week and (I) four weeks. Although our example is simple, it is not possible to achieve a perfectly balanced schedule. Given the complexities of the program on which these techniques are often used, most managers would be happy to achieve the success we did in this example.

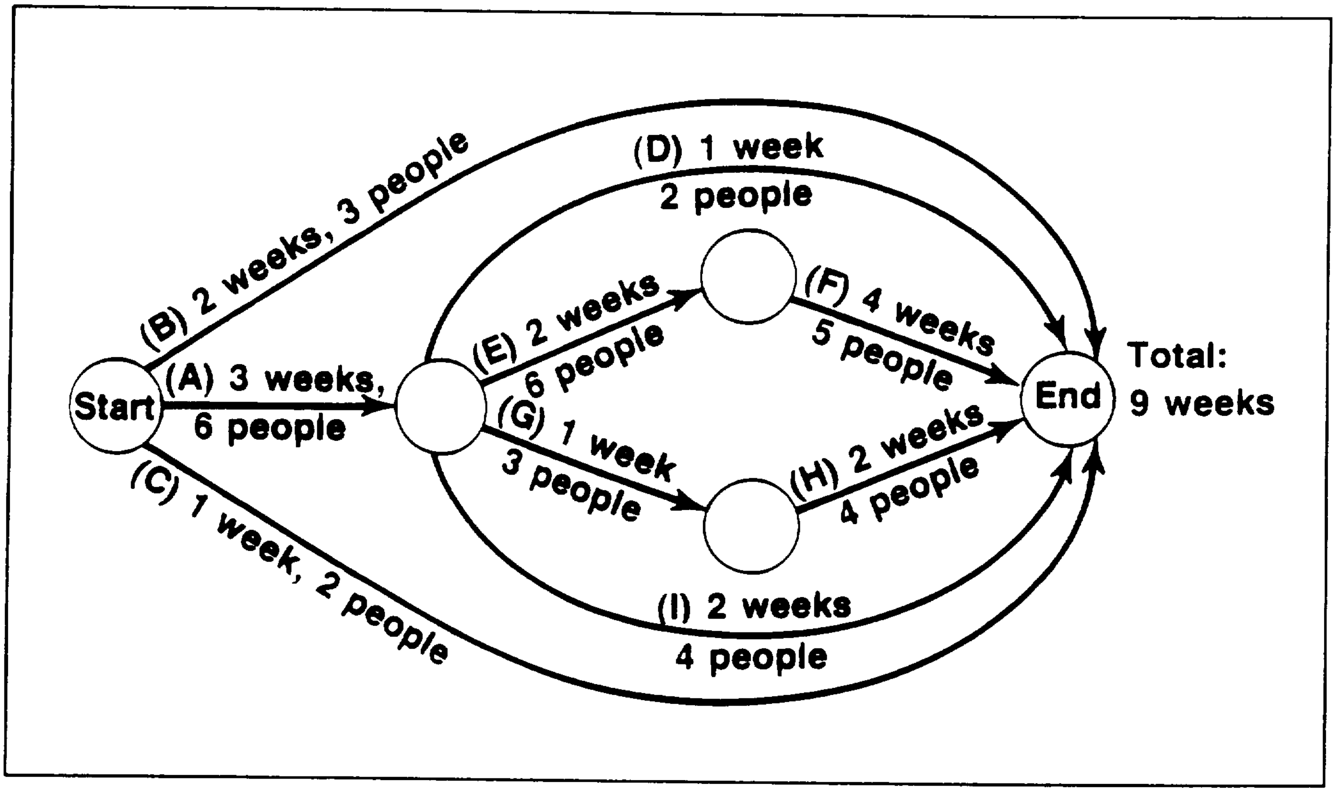

Figure 10. Network Illustrating Problem When Resources Are Limited

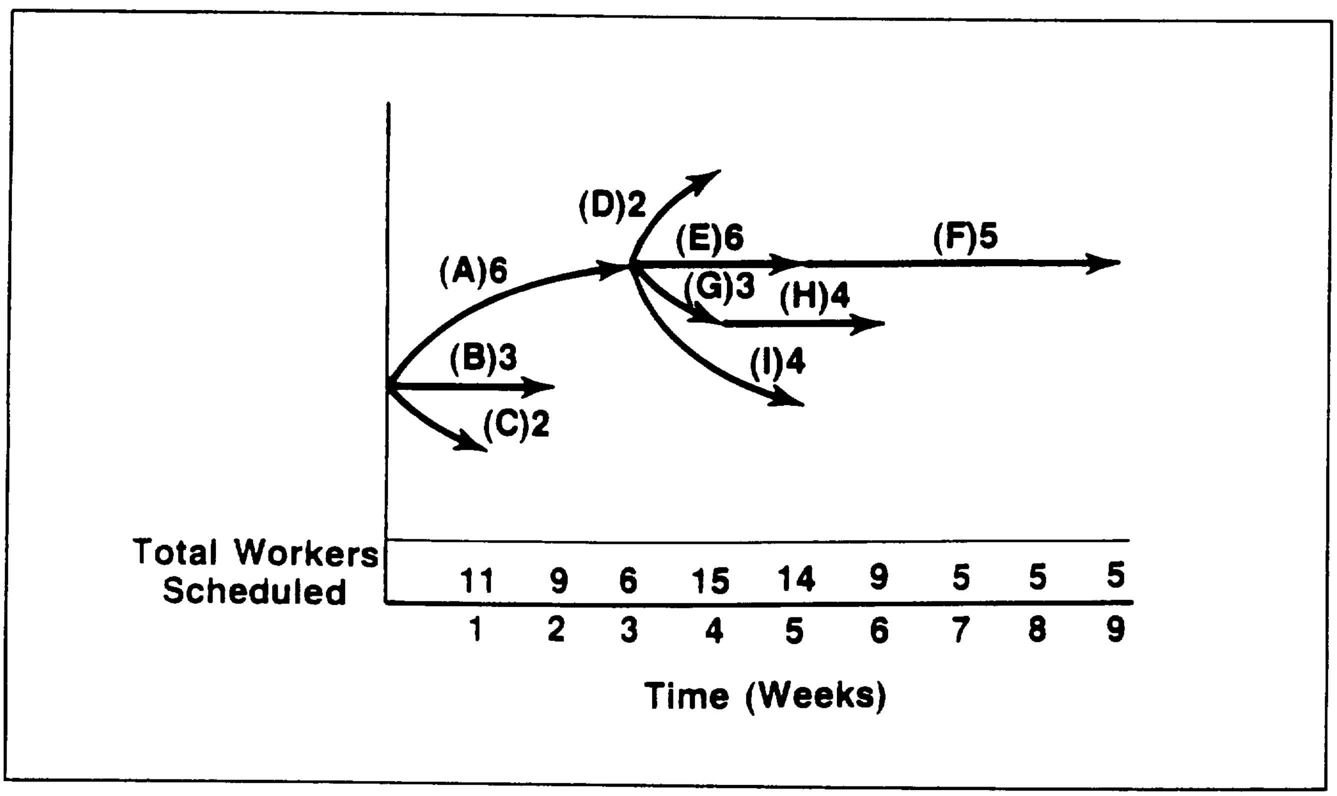

Figure 11. Limited Resource Network Plotted on Schedule Graph

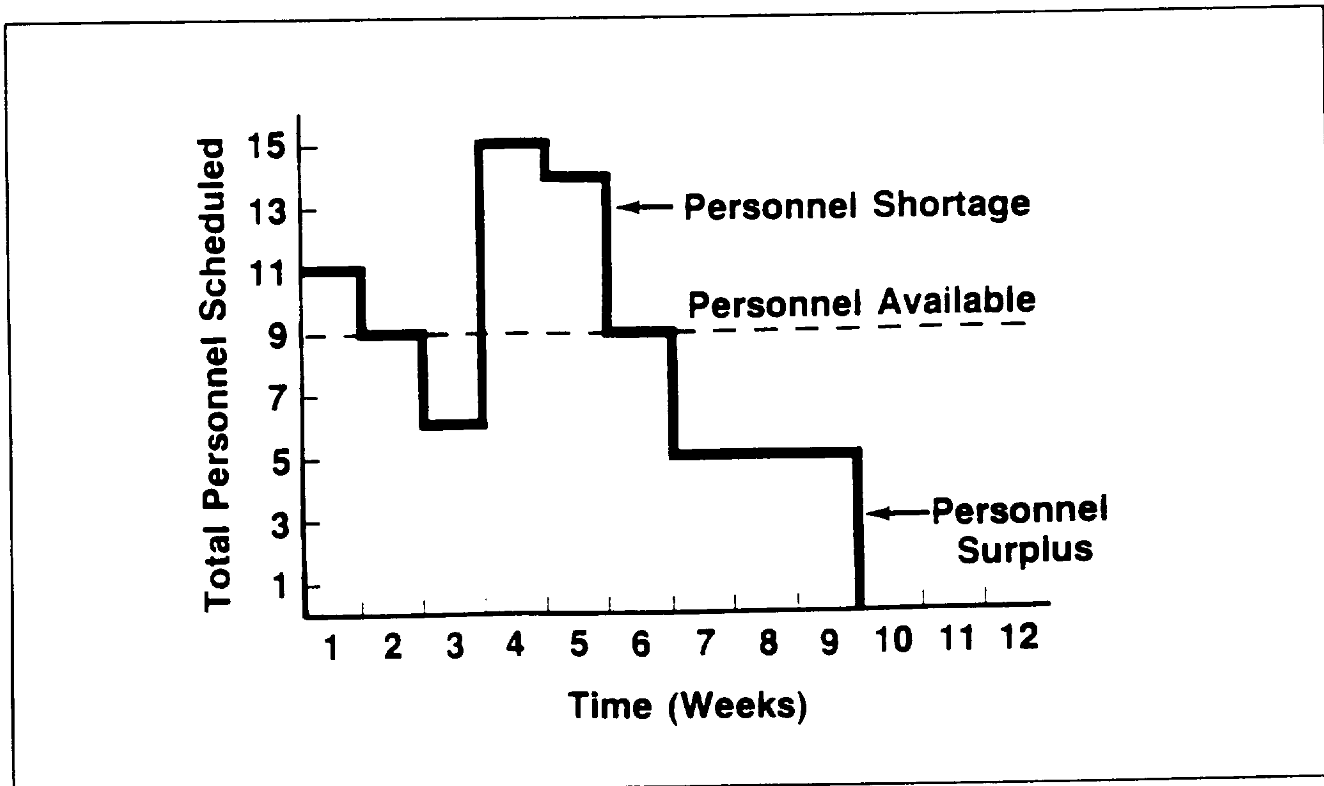

Figure 12. Personnel Loading Chart for Schedule Graph

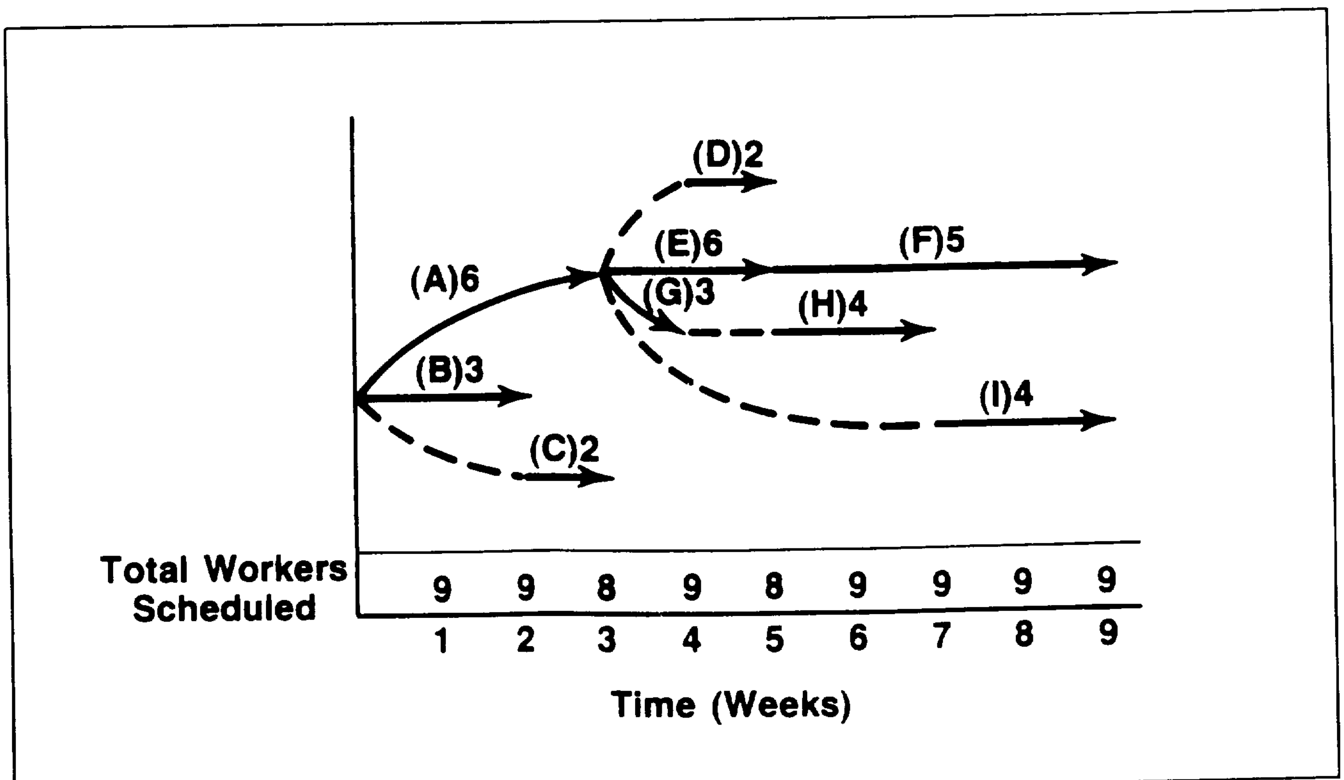

Figure 13. Revised Schedule Graph

Multi-program Considerations

In his dissertation at Purdue University in 1964, J.H. Mize offered a method for multi-method control. He developed a non-iterative heuristic model that schedules activities for several operating facilities of a multi-program organization when the objective is to minimize due date slippage. The outputs from program critical path analyses become the inputs to the schedule. Mize took into account the dynamic relationships of activities to activities and program to program when conflicts arise. The method offered by Mize is applicable generally to any program involving more than one program competing for the same limited resources.

In 1968, L.G. Fendley developed a system based on the concept of assigning the due dates to incoming programs and then sequencing activities of the programs toward meeting the due-date. He used the heuristic approach to solve the scheduling problem. Fendley concluded that giving priority to the activity with minimum slack-from-due date (his MSF rule) resulted in the best performance. He used the MSF rule to set realistic due dates by determining the amount of slippage that must occur to perform all programs with fixed resources.

In 1970, Mize and L.F. Jordan applied a simulation technique to the scheduling of multi-engineering programs. They discovered that a rule based upon a combination of processing time and due date yielded good results.

All networking concepts can be applied to the scheduling of several programs jointly administered by a single organization. For example, consider the three programs shown in Figure 14. In this example, Program A must be completed before Program B can start. Program C and Program D may begin and be completed any time between week 1 and week 6, respectively. Thus, the dotted lines in Figure 14 indicate dummy dependencies only. They serve to indicate the time span

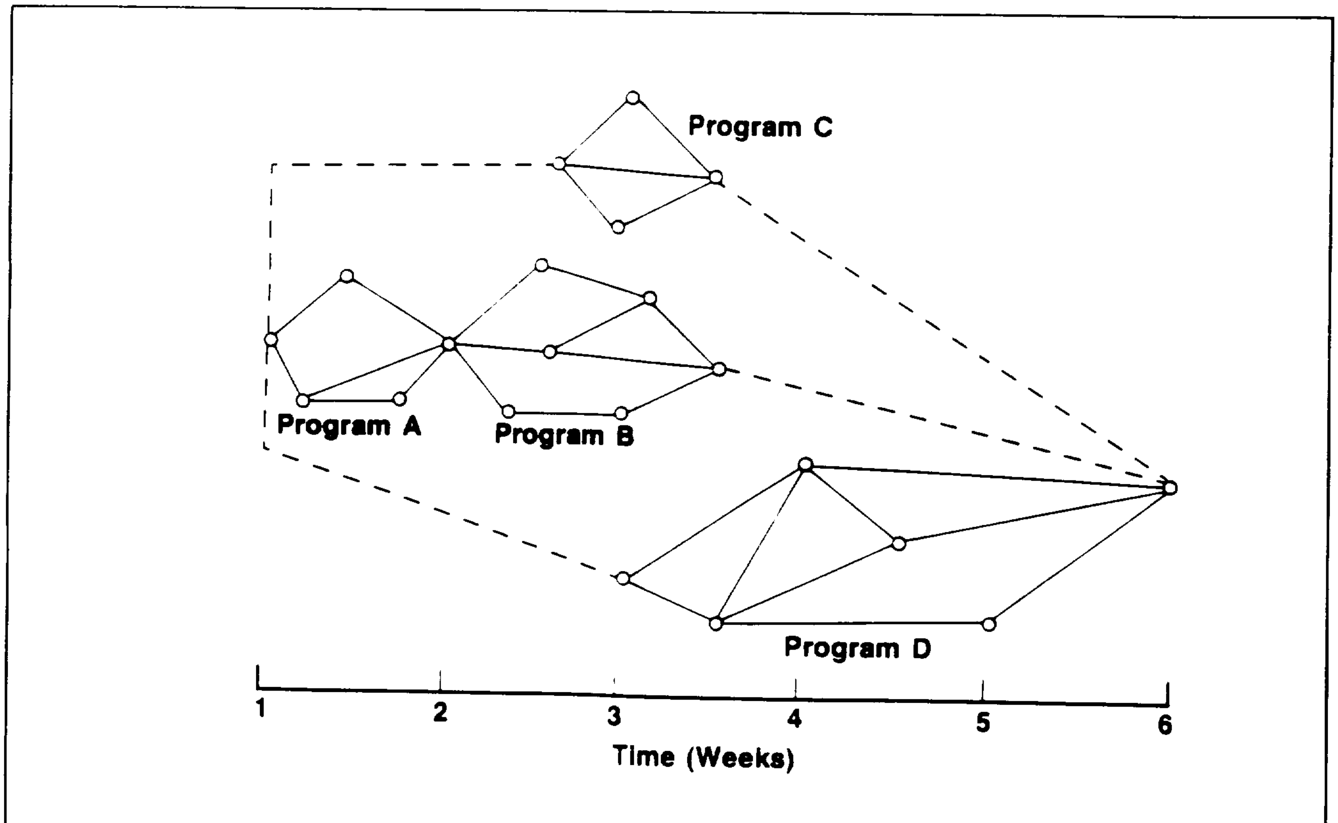

Figure 14. A Multi-program Network

available for all four programs. Duration times could be placed on these dummies to achieve early start and late finish program dates, if they exist. The program floats implied by these dummy jobs can be used in the same way that dummy jobs are used in single-program networks.

For example, suppose the same resources are used on Programs B and C. Furthermore, suppose these resource requirements exceed the availabilities because of the simultaneous demands by Programs B and C. Figure 14 shows that the start of Program C can be delayed until week 4, while the resources are fully employed on Program B. After Program B is completed, the resources can be released for use on Program C. Alternatively, both programs can use the resources at a reduced rate, and both programs will then float out (as long as they do not float beyond week 6.) Whole programs may be cost-expedited. Thus, multi-program networking techniques are completely analogous to single-program networking techniques.

There is, however, one new aspect in multi-program scheduling: program priorities. Suppose that Program C (Figure 14) is deemed to be the most important program and management wishes to have it start before any other program. In the Resource Allocation and Multi-Project Scheduling (RAMPS) computer algorithm developed in 1963 at C–E–I–R, Inc. by Moshman, Johnson and Larsen, the program priority is used as a weighting factor in scheduling and allocating resources among competing alternative uses in the multi-program network.

In general, the iterative use of multi-program level and program-level network methods provides a medium through which program and department level managers may devise integrated total plans. Optimized networks may be submitted by each program manager. In 1974, Woodworth and Dane found these networks could be merged into a multi-program network. Several multi-program network schedules may be developed, given various assumptions about priorities and resources. These alternative multi-program schedules may then be examined in staff meetings attended by each program manager and multi-program manager. The best multi-program schedule may then be selected, based on discussions and criticisms by everyone involved. Of course, several iterations of the schedule may be required between the program and multi-program level before an acceptable plan is developed.

Influence on Program Performance

Program completion may be strongly influenced by the company's risk-failing propensity, the customer's decision process and the ability of the company to expand its organization rapidly without losing its effectiveness. On some programs, these aspects may have more influence on performance.

Network scheduling techniques, like PERT and CPM, are much alike in providing interdependencies, depth of detail, a critical path and slack. The swan technique provides simplicity and visibility through time scales that have been used for many years in bar charts.

The choice between PERT and CPM depends primarily on the type of program and managerial objectives. The PERT is particularly appropriate if there is considerable uncertainty in program activity times, and it is important to control the program schedule effectively. On the other hand, CPM is particularly appropriate when activity times can be adjusted readily, and it is important to plan an appropriate tradeoff between program time and cost.

Actually, differences between current versions of PERT and CPM are not necessarily as pronounced as this section may convey. Most versions of PERT now allow only a single (most likely) estimate of each activity time.

When several small programs are to be scheduled, a multi-program network might be considered. In this situation, each program can be treated as a separate entity and the entire set of programs diagrammed and handled as one large network. The RAMPS computer program is convenient to apply in such a situation. The programs in the multi-program network should be importance-weighted or priority-constrained. This will determine which programs to schedule earlier than others. Table 4 cites the strengths and weaknesses of network scheduling techniques.

Line-of-Balance Technique

Network scheduling techniques are used primarily in development and other one-time programs. The line-of-balance (LOB) technique is used in repetitive activities like production. In production programs, LOB charts are particularly useful to balance inventory

Table 4. Networking Technique: Strengths and Weaknesses

CRITERIA	STRENGTHS	WEAKNESSES
1. Accuracy	N/A	The technique is as accurate as the activity-time estimates. The margin of error is generally less in construction than in development.
2. Reliability	N/A	Compounded, unreliable estimates in a large program may lead to unreliable status information.
3. Simplicity	Brings simple order out of mass confusion.	Concepts of slack and network families can be difficult to grasp. Computerized networking complicates the process; however, on a complex program without computerization for criteria 5 through 8, the strengths shown cannot be obtained readily.
4. Universality of Program Coverage	Very good for one-time programs like construction and development.	Weak in production phase of life cycle. Not well-suited for quantity production.
5. Decision Analysis	Excellent for stimulating alternatives, especially when coupled with time-cost data.	If computer based.
6. Forecasting	Excellent for forecasting ability to meet schedules.	If computer based.
7. Updating	Easy to update estimates as progress information is received.	If computer based.
8. Flexibility	Portions of network can be changed easily to reflect program changes.	If computer based.
9. Cost		Because considerable data is required, it is usually costly – especially if computerized.

acquisition with the production process and delivery requirements.

A line-of-balance chart shows which control points need attention, not to maintain future delivery schedules. Using the LOB technique, reporting to customers or top management is quick, inexpensive and graphic. Charts used for analysis and trouble shooting are suitable for at-a-glance status reporting. Without a computer controlled production process, the line-of-balance technique doesn't lend itself readily to day-to-day updating. However, a monthly or weekly check usually keeps the process on schedule.

A line-of-balance technique consists of four elements: (1) objectives of the program; (2) production plan; (3) current program status; and (4) a comparison between where the program is and where it's supposed to be, striking the line-of-balance (Figure 15).

The first step in preparing the LOB is drawing the contract delivery schedule on the objective chart (Figure 15-A), which shows cumulative units on the vertical scale and dates of delivery along the horizontal scale. The contract schedule line shows the cumulative units to be delivered over a period of time on the program. Actual deliveries to date (cumulative) are shown. The second step is charting the production plan (Figure 15-B). The assembly plan is a lead time chart. Select only the most meaningful events as control points in developing this chart.

These main events can be given symbols that show whether they involve purchased items, subcontracted parts, or parts and assemblies produced in-house. Assemblies break down into subassemblies, which break down into parts or operations. Thus, one can develop a production plan for any part or level of assembly.

The more steps that are monitored, the more sensitive and more complicated the chart becomes. Generally, control points on a single chart should be limited to 50. If there are more than 50, subsidiary production plans can be used to feed the top plan. Thus, each chart can be kept simple and easy to understand. The shipping date of subsidiary charts is when a sub-program must be ready to join the overall schedule.

On the production plan chart, each monitored step is numbered left to right. Step 1 has the longest lead time. The shipping date is the highest numbered step. When two steps are done at the same time, they are numbered from top to bottom.

The production plan chart shows interrelationships and sequence of major steps, and lead times required for each step. An understanding of the manufacturing processes involved and sound judgment are required to know which step and how many steps must be monitored.

The 12 control points in the production plan chart used as an example represent key tasks in manufacturing one lot of missiles. The plan indicates that fabricate ballistics shell (control point 1) must begin 24 work days before government acceptance. Thus, this activity must begin 24 work days before January 1 to meet the first scheduled delivery of five units by the end of December (see the objective chart). The lead time for other control points can be related to the scheduled delivery in a similar manner.

In a five-day-week operation, a month is generally recognized as having 22 work-days. Time for in-house transfer and storage must be allowed in addition to the processing time.

To control production, the manager needs monthly status information for each control point. On the program status chart (Figure 15-C), the bar for control point (12) shows that 14 units of the product have been accepted by the government. The bar for control

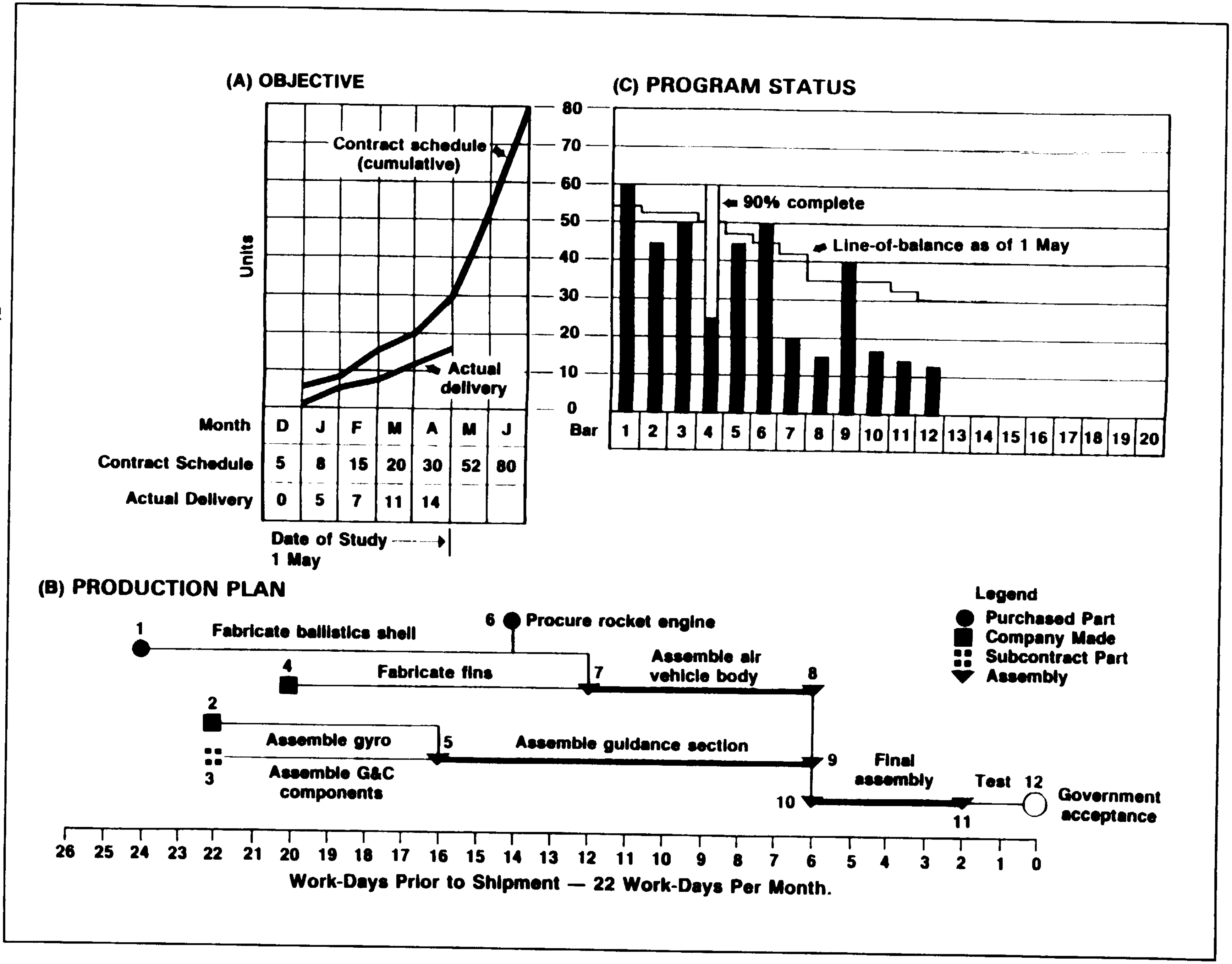

Figure 15. Line-of-Balance Charting

point (9) shows that 40 units of the guidance section have been assembled. The bar for control point (4) shows that in-house fabrication has begun on 60 fins, but only 25 have been completed. The status at other control points are determined in a similar manner. Final deliveries (government acceptances) are shown month-by-month on the objective chart.

To analyze how present status of each control point will affect future schedules, the line-of-balance (LOB) has been constructed. The line-of-balance represents the number of units of the product that should have passed through each control point to satisfy future delivery schedules. This line-of-balance is drawn above the bars on the program status chart to show the status of control points. Normally, the line steps down to the right.

The difference between the line and the top of the bar for each control point is the number of units behind or ahead of schedule. Thus, control point (12) is 16 units behind schedule, control point (9) is 5 units ahead of schedule, and control point (7) is 21 units behind schedule. Control point (12) is behind schedule now, May 1, because there is no lead time available for it. The main impact of control point (7) being behind schedule will be felt in 12 workdays, which is the lead time for control point (7). An insufficient number of assembled air vehicle bodies started into production on May 1. This will adversely affect final deliveries 12 workdays hence. All other control points can be analyzed in the same way.

To recap, the line-of-balance is constructed in the following manner:

- Select a control point, for example (7).
- From the program (Figure 15-B) determine the lead time, the time from the control point to shipment point (12 workdays).
- Using this number, determine the date the units now at the control point should be completed. (May 1 plus 12 workdays is May 16).
- Find the point corresponding to this date (May 16) on the contract schedule line and determine how many units scheduled for completion this represents (41 units).
- Draw a line on the program status chart (Figure 15-C) at that level (41 units) over the control point (7).
- Repeat for each control point and connect the horizontal lines over the control points. The resulting line is the LOB, indicating the quantities of units that should have passed through each control point on the date of the study (May 1) if the delivery schedule had been met.

Analysis

Using the LOB charts in Figure 15, management can tell at glance how actual progress compares with planned progress. Analysis of charts can pinpoint problem areas. Delays at control point (7) in the example may have been causing final delivery problems throughout the contract. However, the purpose of line-of-balance analysis is not to show what caused the slippage in the shipping date, but to detect potential future problems.

In the example, the government acceptance point is control point (12). The bar does not reach the line-of-balance; therefore, deliveries are behind schedule. Control points (10) and (11) are short. However, point (9) is on schedule. Since point (10) depends on points (8) and (9), we know control point (8) is the offender. Both points (7) and (8) are short, but there are more than enough purchased items at control point (6). What's the problem with control point (7)? Trace it back to control point (4), which is seriously short. It is obvious that not having enough completed-fins is holding up the whole progress. Control points (2), (3) and (5) are short, but are not directly responsible for the failure to meet the delivery schedule. The problem with the fins should be addressed before management attention is devoted to other short operations. The averages at control points (1) and (6) may be examined from the point of view of inventory control.

Updating the charts requires a good status reporting system, which can be mechanized if the program is large and complex. A computer program, developed by the U.S. Army Management Engineering Training Activity (AMETA), Rock Island, Illinois, provides printouts of all information required on LOB charts. Actually, because the program provides all information, printouts can be used by themselves. Charts are not required, but a graphic display of the information is usually desirable.

The line-of-balance is a means for measuring actual progress against a scheduled objective. It employs the exception principle. The four phases to line-of-balance are: the objective, the program, program progress and comparison of program progress to the objective. The statement of objective is presented in terms of the number of units/time period, number of units to be delivered, scheduled completion date or any other appropriate quantity/time combination (Figure 15-A).

The graphical representation of the principal steps to be taken enroute to the objective—a modified Gantt Chart—is shown in the production plan chart (Figure 15-B).

The graphical representation of the inventory of the stock status for the principal steps is shown in the program status chart (Figure 15-C) with a vertical axis of the same units as those shown in the objective chart.

Striking the line-of-balance involves transferring points from the objective chart to the program status chart for the date being studied. A program that is exactly in phase results in a line-of-balance that intersects every bar of the program status chart at (or near) its top.

Because the LOB technique is production oriented, it provides quick detection of bottlenecks in the production process. Management can then take appropriate action, such as increasing resources at each bottleneck. Table 5 summarizes strengths and weaknesses of the LOB technique.

Table 5. Line-of-Balance Technique: Strengths and Weaknesses (In Production)

CRITERIA	STRENGTHS	WEAKNESSES
1. Accuracy	Completion time estimates are good because work is repetitive; however, this may not be true early in the production phase of a program.*	N/A
2. Reliability	Compares favorably with Gantt technique.	N/A
3. Simplicity	N/A	Construction of the line of balance is not always understood.
4. Universality of Program Coverage	N/A	Well suited only for production phase of life cycle. Does not emphasize resource allocation directly.
5. Decision Analysis	N/A	No capability to simulate alternatives.
6. Forecasting	Very good for indicating whether or not schedules can be met.	N/A
7. Updating	N/A	Considerable clerical effort is needed to update graphs. Computer processing can reduce this effort.
8. Flexibility	N/A	Inflexible. When major program changes occur, all LOB phases must be redesigned.
9. Cost	Data gathering and computations can be handled routinely and at moderate expense.	

* Most of the production span time (~70 to ~80 percent) consists of wait and move time. These times are usually less accurate than the standard times used for set up and run. Reporting accuracy is a key to the reliability of this technique.

"Dost thou love life, then do not squander time for that's the stuff life is made of."

– Benjamin Franklin

Time Management

Program managers are busy people, particularly those in the DoD and defense-related industry. Therefore, it is important that program managers manage time well. Some managers could be more productive, perhaps as much as 20 to 40 percent, by applying effective approaches.

This section concerns the aspects of time management related to programs: the program manager's time reserve, a "now" schedule and value of time.

In contractor performance measurement, much emphasis is placed on "management reserve," the reserve budget controlled by the industry program manager. The program manager doesn't know when or where this reserve will be needed. But the program manager in industry and his counterpart in DoD know it will be needed.

A time reserve is needed as well as a budget reserve; the program manager needs a time reserve to accommodate unknowns that he will encounter. The use of time reserve should be approached with caution, especially where it is visible, so as not to negate the value of the schedule plan and status for management use.

Literature describing a program manager's time reserve is scarce. Based on discussions with a sampling of managers of large and small programs, the main aspects of time reserve are clear.

- Most program managers establish a time reserve of about 10 percent. On a 40-month program, for example, a four-month time reserve would be established.
- The time reserve must be held closely by the program manager, otherwise every manager on his program may think, "I know there's a time reserve; I don't really have to meet my schedule." The program manager may place this reserve under "additional system tests" or another downstream activity. The point is, it shouldn't be visible. (A built-in safety factor between the manufacturing schedule and the delivery schedule is often used.)
- A tough and disciplined approach to meeting the published schedule is required from the start of a program in order to maintain the reserve and, consequently, to meet the program schedule in spite of slippages caused by the unknown unknowns (unk unks) that inevitably arise.

A direct relationship exists between time and cost for any activity. This relationship takes into account the people, resources and method used, and considers the efficiency achieved. Generally, the least costly schedule is the current one. To speed up the schedule costs more; to stretch out the schedule also costs more. The sum of the direct and indirect costs gives a U-shaped total program cost curve. The optimum schedule for implementing the program is the schedule corresponding to the minimum point on this curve. The relationship among direct, indirect and total program costs is shown in Figure 16.

Because schedule stability affects program costs, which may, in turn, affect technical performance, it is clear that schedule stability has a great deal to do with whether the program meets its cost and technical objectives. Unfortunately, budget constraints and other factors, like changes in quantities (items over which the program manager has no control) have often been imposed on a program with the comment, "Do the best you can."

When a schedule must be revised, the superseded schedule is often discarded. If the new schedule is superseded, the process is repeated. Often, the organization causing a slip in

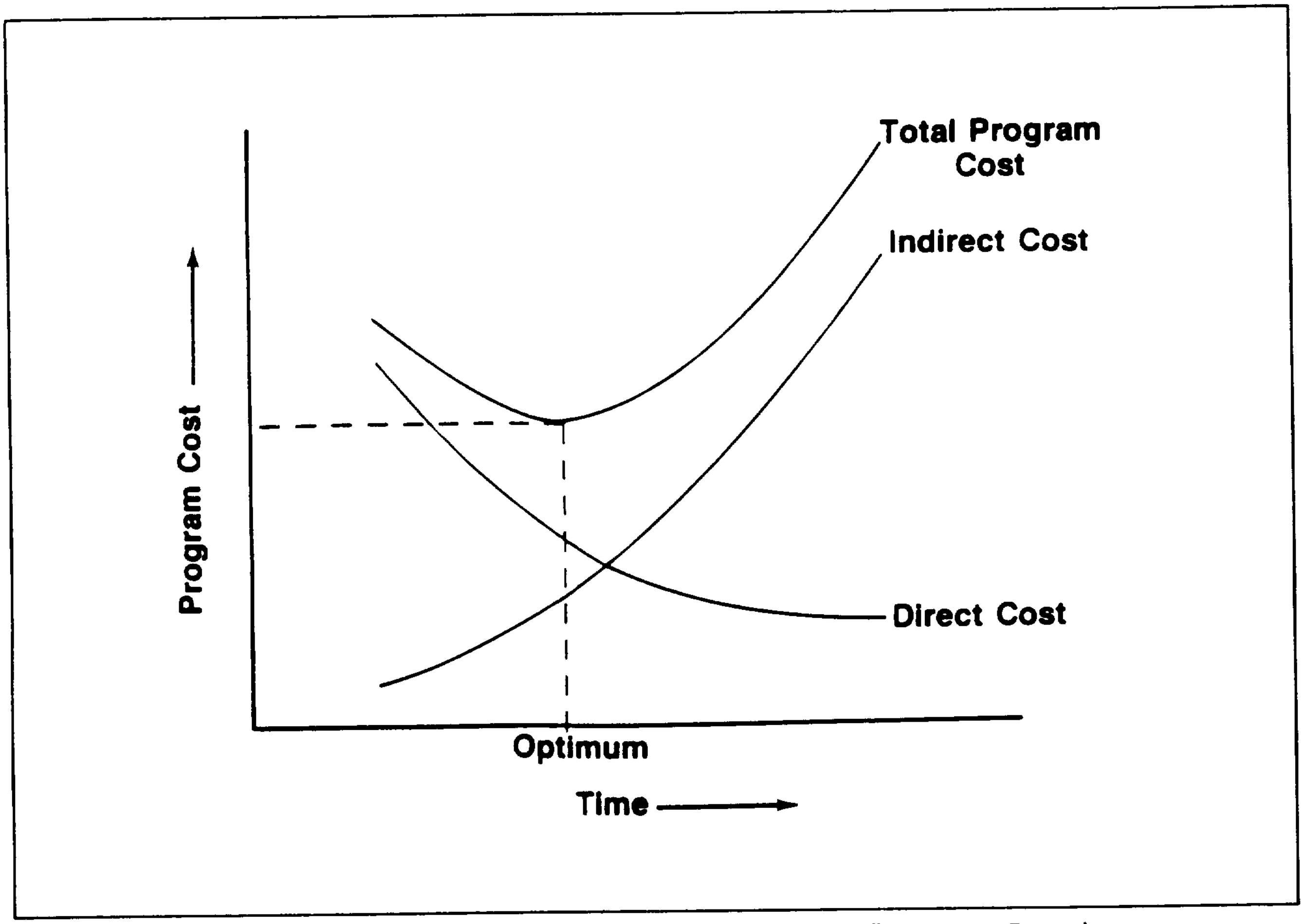

Figure 16. Total Cost Analysis for Selecting "Optimum" Program Duration

schedule becomes a repeat offender. The principal value of retaining a former schedule is in being able to identify the offender, thus making schedule slips less acceptable.

The significance of maintaining a stable schedule is becoming more widely recognized. Appendix A describes the development of a master schedule and the importance of maintaining schedule discipline.

While serving as under secretary of defense (research and engineering), Dr. William J. Perry said, "Our acquisition process is cautious, slow, and expensive. It now takes us 12 years or more for development, production and deployment of a typical (defense) system, so that our lead in technology is lost by the time the system is deployed."

According to the late John H. Richardson, president of Hughes Aircraft Company, "A basic reason for adopting project (or program) management, when tackling the difficult and unique tasks associated with developing and producing a system, is to eliminate unnecessary delays in accomplishing the job at hand. Time is a resource in systems management, to be treated with indifference or used well like any other resource. For projects not yet in full swing, it is important to recognize that time has economic value, and that we may be taking time too much for granted." Richardson cites historic reasons for stretching out program schedules.

- Funding can create a problem. In hungry years the schedule is often stretched because of reduced funding.

- A better product can be expected if it is more thoroughly debugged and tested. However, a system does not really get wrung-out until it is in the user's hands, regardless of beforehand debugging.

- Cost of concurrency (overlap of development and production), may lead to a decision not to overlap program phases. Such a decision may be popular in many cases but it cannot be tolerated when the pendulum swings toward the importance of time; that is, when top management says "Get the system out the door; never mind what it costs."

Stretched-out schedules incur cost penalties because of inflation, additional engineering changes, and changes in program managers or other key program management officials. Another near-term cost results from the increased chance that a program will be canceled because of obsolescence or competing technology. History shows that stretch-outs invite cancellation. Also, competing technology, which may get a toe-hold during a program stretch out, may lead to a program cancellation. Long schedules with no opportunities to incorporate improvements are a negative factor when considering a new start.

Delayed decisions increase costs. For example, waiting to acquire 90 percent of the facts bearing on a decision, rather than going ahead when 80 percent of the facts are in hand, is not usually cost effective. The schedule is prolonged when the decision is withheld.

According to R. W. Peterson, former Du Pont executive, "All business men are concerned, and properly so, about the long time it takes to move a new development from its inception to a profit status. But frequently forgotten is the fact that a month's delay in the early stages of development is exactly as long as a month's delay in the later stages. While it may seem innocuous to put off a decision for a month or two in the early years of the project (or program) with an uncertain future, that delay may turn out to be just as costly as is procrastination when the final decisions are made. In short, a sense of urgency is essential to decision making in all stages of a new venture, not just the latter stages."

A consideration having more impact on the value of a defense system, a point often overlooked, is the useful life of the system. Leading producers of commercial and industrial products are aware of the importance of bringing a new product to market without delay to gain the greatest return on the costs of product development and production.

Making use of time to increase the life of a system applies to defense systems as well as to commercial/industrial products/systems. Concentration on system or product cost, without considering the life of the resulting system or product, overlooks a key point: whether the buyer obtains value for each dollar. The most costly product, in terms of value, is one appearing when it no longer fulfills a useful purpose, even though it has been produced at minimum cost. Each month added to the development and production of a new high-technology system or product tends to reduce by one month the operational life of the system or product.

In spite of the 10 to 20 percent cost premium that may be paid for tight scheduling, as compared to orderly but stretched-out scheduling, the longer resulting operational life usually provides greater economic value. This is looking at time only from the viewpoint of economics; i.e., acquisition cost per year of operational availability.

Another way of looking at time is that defense system availability is survival insurance. An executive of a major shipbuilding company noted that, "the time we're spending waiting for our ships to come in . . . is time we just may not have."

Consideration of alternative plans and schedules will also help; e.g., if event so-and-so occurs, proceed with plan "A"; if event such-and-such occurs, proceed with plan "B", and so on. Anticipation and preparation for most likely events, along with the tools described, and coupled with effective communication of the plans, can change the management style from crisis management to skillful management.

Planning and scheduling can do much to prevent running out of time and having to make the least desirable decision because of lack of time. Establishing a time reserve, a "now" schedule and recognizing the value of time in decision making all contribute to the manager's repertoire of good tools.

Sir Jeffrey Quill, manager of the British Spitfire Development Program, commented during a visit to the Defense Systems Management College that, "After 1935, costs were not particularly important. What mattered was time. We worked three shifts a day. Everything was time. Quantity and time. It turned out that we probably produced at the lowest cost too; but the emphasis was on time."

"I wasted time: now doth time waste me."
– Richard II

Recapitulization

The program manager is responsible to top management for getting the job done on schedule and within the allowable cost. Today, network based systems assist the manager in planning, scheduling and controlling the work to be accomplished—often by people in separate organizations not under the manager's direct control. The manager needs a plan that will provide a constant and up-to-date picture of the operation that is understood by all.

Scheduling, cost and performance are major elements of concern to the program manager, who should be able to blend them to meet program objectives. When selecting a scheduling method, the program manager can make a conscious tradeoff between the sophisticated methods available and cost.

Gantt charts can be used effectively for small programs and when program activities are not highly interconnected. Often, a Gantt chart is selected because, considering the benefits it will provide, network scheduling does not justify the additional cost. Figure 17 illustrates the evolution of network-based systems. The differences between the CPM and PERT techniques result from the environments in which they evolved.

The CPM arrow-diagram network evolved from activity-oriented bar charts. The arrow diagram resulted from linking the activities in a sequence of dependence, often without identification of the connecting points. Factors leading to the CPM technique are a well-defined program, a dominating organization, few small uncertainties and a single geographical location for the program.

The PERT network evolved from a combination of bar charts and milestone charts on which the milestones were identified as events, or specific points in time. The PERT network is heavily event-oriented. Factors that led to the development of the PERT technique are large programs with difficult-to-define objectives, multiple and overlapping responsibilities among organizations, large time and cash uncertainties, and wide geographic dispersal of activities and complex logistics.

When network scheduling is justified, and you wish to choose one of the methods discussed in this guide, be sure to consider that many network scheduling methods are computerized. Software packages are available commercially, or at DSMC, to cover many scheduling methods.

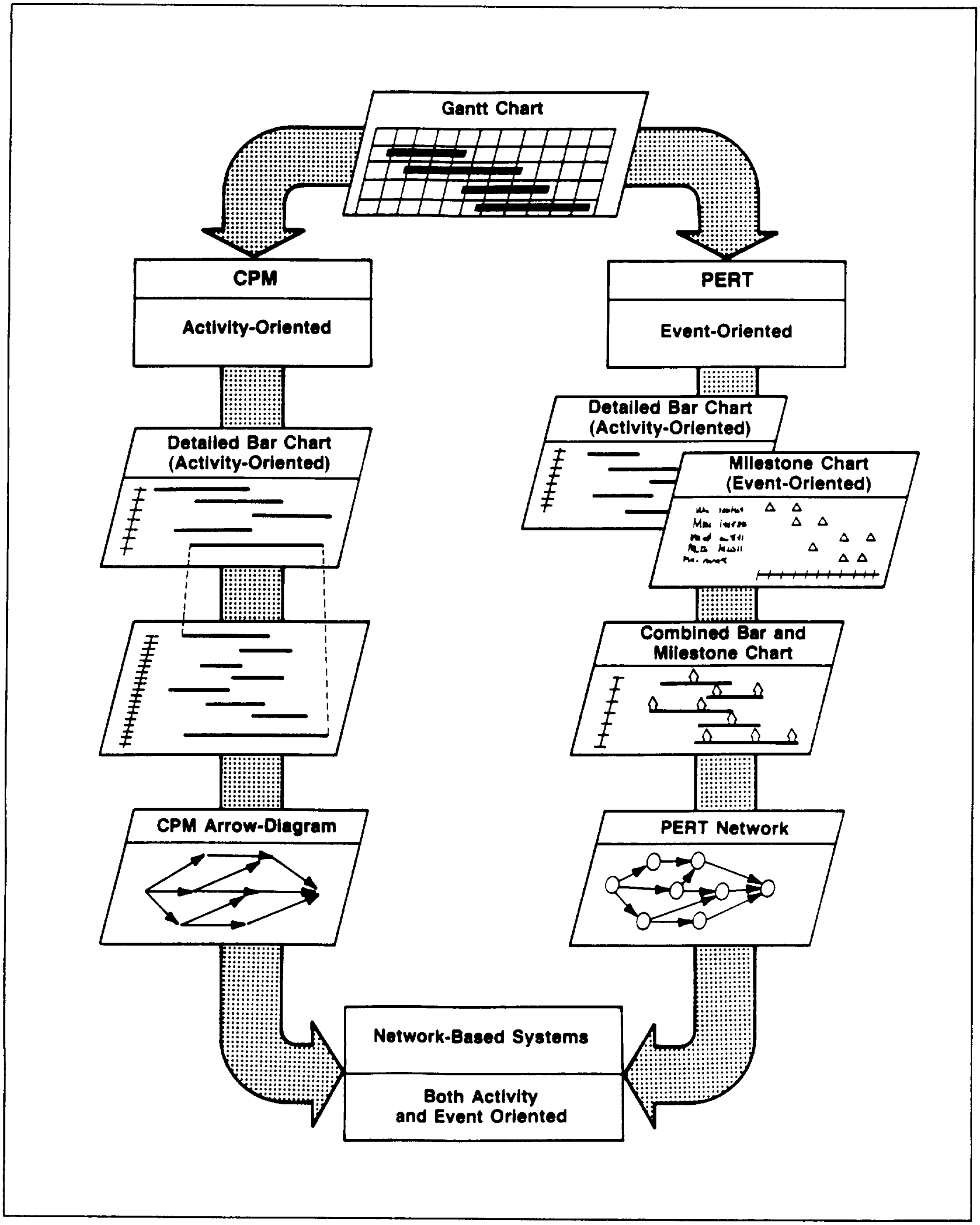

Figure 17. Evolution of Network Based Systems

The principal points to be derived from this section include the following:

- Schedule, time and cost are three major elements to control in any program. These can be in conflict and, tradeoffs may have to be made.

- All programs involve planning, scheduling and controlling. During the planning phase, objectives, organization and resources are determined. During the scheduling phase—the phase with which this section is concerned—personnel requirements have to be determined; time to complete the work and the cost have to be estimated. During the control phase, the program's progress in terms of time, cost, and performance have to be measured. Necessary corrections have to be made to ensure achievement of the program objectives.

- The activity oriented Gantt charts are useful when activities are not closely related and the program is relatively small. The chart shows relationships among variables clearly and quickly, and focuses on situations needing attention.

- The milestone charts, which are event-oriented and display start and completion dates, invite surprises because the program manager may not know the status until an event occurs or fails to occur.

- The network displays how a program can be done and the schedule establishes how it is planned to be done.

- A network identifies the critical path, slack (time an activity or event can be extended and still be completed on time) and activities needing rescheduling. Activities on the critical path have zero slack and must be completed on time to prevent slippage of program completion date.

- The PERT network-based scheduling method may use three time estimates for each event: most optimistic time, most pessimistic time and most likely time.

- The CPM, a network-based scheduling method, uses a linear time-cost tradeoff; i.e., it adds the concept of cost to the PERT format. If necessary, each activity can be completed in less than normal time by crashing the activity for a given cost.

- The line-of-balance (LOB) technique of scheduling is effective in manufacturing where a final assembly line is fed by many component lines, and delivery of end-units is required at predetermined specified intervals. The effectiveness of LOB is based on the design of the assembly plan.

- Computer programs are available for network-based scheduling. Manual calculations are feasible for small problems like those set forth in this section; however, computer assistance is usually necessary for large, complex problems.

- Network theory assumptions that activities are independent, discrete and predictable are not always appropriate in actual applications. The departure from reality, however, does not normally affect planning and coordinating efforts on critical-path scheduling.

Resources

L. G. Fendley, "Toward the Development of a Complete Multi-Project Scheduling System," *Journal of Industrial Engineering,* 1968.

Ted Ingalls, "We're Looking for a Few Good People," *Program Manager,* July-August 1984.

K. R. MacCrimmon and C. A. Ryavec, *An Analytical Study of the PERT Assumptions,* Memo RM-3408-PR, Rand Corporation, Santa Monica, CA, December 1962.

J. H. Mize, *A Heuristic Scheduling Model for Multi-Project Organizations,* unpublished Ph. D. thesis, Purdue University, Lafayette, IN, 1964.

J. Moshman, J. Johnson, and M. Larsen, "RAMPS – A Technique for Resource Allocation and Multi-Project Scheduling," *Proceedings of the 1963 Spring Joint Computer Conference,* Spartan Books, Baltimore, MD, 1963.

Russell W. Peterson, former Du Pont executive, Governor of Delaware, and White House advisor, "New Venture Management in a Large Company," *Harvard Business Review,* May-June 1967, p. 72.

John H. Richardson, *Time Defeats Technology,* Hughes Aircraft Company, Culver City, CA, date unknown.

B. M. Woodworth and C. W. Dane, *Multi-Project Network Analysis with Resource Leveling: State-of-the-Art and a Governmental Application,* Oregon State University, Corvalis, OR, 1974.