Harvard
Business
Review

Process Management

The ABCs of the Critical Path Method

by F. K. Levy, G. L. Thompson, and J. D. Wiest

From the Magazine (September 1963)

Recently added to the growing assortment of quantitative tools for business decision making is the Critical Path Method—a powerful but basically simple technique for analyzing, planning, and scheduling large, complex projects. In essence, the tool provides a means of determining (2) which jobs or activities, of the many that comprise a project, are "critical" in their effect on total project time, and (2) how best to schedule all jobs in the project in order to meet a target date at minimum cost. Widely diverse kinds of projects lend themselves to analysis by CPM, as is suggested in the following list of applications:

- The construction of a building (or a highway).
- Planning and launching a new product.
- Installing and debugging a computer system.
- Research and engineering design projects.
- Scheduling ship construction and repairs.

- The manufacture and assembly of a large generator (or other job-lot operations).

- Missile countdown procedures.

Each of these projects has several characteristics that are essential for analysis by CPM:

(1) The project consists of a well-defined collection of jobs (or activities) which, when completed, mark the end of the project.

(2) The jobs may be started and stopped independently of each other, within a given sequence. (This requirement eliminates continuous-flow process activities, such as oil refining, where "jobs" or operations necessarily follow one after another with essentially no slack.)

(3) The jobs are ordered—that is, they must be performed in technological sequence. (For example, the foundation of a house must be constructed before the walls are erected.)

What is the Method?

The concept of CPM is quite simple and may best be illustrated in terms of a project graph. The graph is not an essential part of CPM; computer programs have been written which permit necessary calculations to be made without reference to a graph. Nevertheless, the project graph is valuable as a means of depicting, visually and clearly, the complex of jobs in a project and their interrelations.

First of all, each job necessary for the completion of a project is listed with a unique identifying symbol (such as a letter or number), the time required to complete the job, and its immediate prerequisite jobs. For convenience in graphing, and as a check on certain kinds of data errors, the jobs may be arranged in "technological order," which means that no job appears on the

list until all of its predecessors have been listed. Technological ordering is impossible if a cycle error exists in the job data (e.g., job *a* precedes *b*,*b* precedes *c,* and *c* precedes *a*).

Then each job is drawn on the graph as a circle, with its identifying symbol and time appearing within the circle. Sequence relationships are indicated by arrows connecting each circle (job) with its immediate successors, with the arrows pointing to the latter. For convenience, all circles with no predecessors are connected to a circle marked "Start"; likewise, all circles with no successors are connected to a circle marked "Finish." (The "Start" and "Finish" circles may be considered pseudo jobs of zero time length.)

Typically, the graph then depicts a number of different "arrow paths" from Start to Finish. The time required to traverse each path is the sum of the times associated with all jobs on the path. The critical path (or paths) is the longest path (in time) from Start to Finish; it indicates the minimum time necessary to complete the entire project.

This method of depicting a project graph differs in some respects from that used by James E. Kelley, Jr., and Morgan R. Walker, who, perhaps more than anyone else, were responsible for the initial development of CPM. (For an interesting account of its early history see their paper, "Critical-Path Planning and Scheduling."[1]) In the widely used Kelley-Walker form, a project graph is just the opposite of that described above: jobs are shown as arrows, and the arrows are connected by means of circles (or dots) that indicate sequence relationships. Thus all immediate predecessors of a given job connect to a circle at the tail of the job arrow, and all immediate successor jobs emanate from the circle at the head of the job arrow. In essence, then, a circle marks an event—the completion of all jobs leading into the circle. Since these jobs are the immediate prerequisites for all jobs leading out of the circle, they must all be completed before *any* of the succeeding jobs can begin.

In order to accurately portray all predecessor relationships, "dummy jobs" must often be added to the project graph in the Kelley-Walker form. The method described in this article avoids the necessity and complexity of dummy jobs, is easier to program for a computer, and also seems more straightforward in explanation and application.

In essence, the critical path is the bottleneck route. Only by finding ways to shorten jobs along the critical path can the overall project time be reduced; the time required to perform noncritical jobs is irrelevant from the viewpoint of total project time. The frequent (and costly) practice of "crashing" *all* jobs in a project in order to reduce total project time is thus unnecessary. Typically, only about 10% of the jobs in large projects are critical. (This figure will naturally vary from project to project.) Of course, if some way is found to shorten one or more of the critical jobs, then not only will the whole project time be shortened but the critical path itself may shift and some previously noncritical jobs may become critical.

Example: Building a House

A simple and familiar example should help to clarify the notion of critical path scheduling and the process of constructing a graph. The project of building a house is readily analyzed by the CPM technique and is typical of a large class of similar applications. While a contractor might want a more detailed analysis, we will be satisfied here with the list of major jobs (together with the estimated time and the immediate predecessors for each job) shown in Exhibit I.

EXHIBIT I
Sequence and Time Requirements of Jobs

Job No.	Description	Immediate predecessors	Normal time (days)
a	Start		0
b	Excavate and pour footers	*a*	4
c	Pour concrete foundation	*b*	2
d	Erect wooden frame including rough roof	*c*	4
e	Lay brickwork	*d*	6
f	Install basement drains and plumbing	*c*	1
g	Pour basement floor	*f*	2
h	Install rough plumbing	*f*	3
i	Install rough wiring	*d*	2
j	Install heating and ventilating	*d,g*	4
k	Fasten plaster board and plaster (including drying)	*i,j,h*	10
l	Lay finish flooring	*k*	3
m	Install kitchen fixtures	*l*	1
n	Install finish plumbing	*l*	2
o	Finish carpentry	*l*	3
p	Finish roofing and flashing	*e*	2
q	Fasten gutters and downspouts	*p*	1
r	Lay storm drains for rain water	*c*	1
s	Sand and varnish flooring	*o,t*	2
t	Paint	*m,n*	3
u	Finish electrical work	*t*	1
v	Finish grading	*q,r*	2
w	Pour walks and complete landscaping	*v*	5
x	Finish	*s,u,w*	0

Exhibit I Sequence and Time Requirements of Jobs

In that exhibit, the column “immediate predecessors” determines the sequence relationships of the jobs and enables us to draw the project graph, Exhibit II. Here, in each circle the letter before the comma identifies the job and the number after the comma indicates the job time.

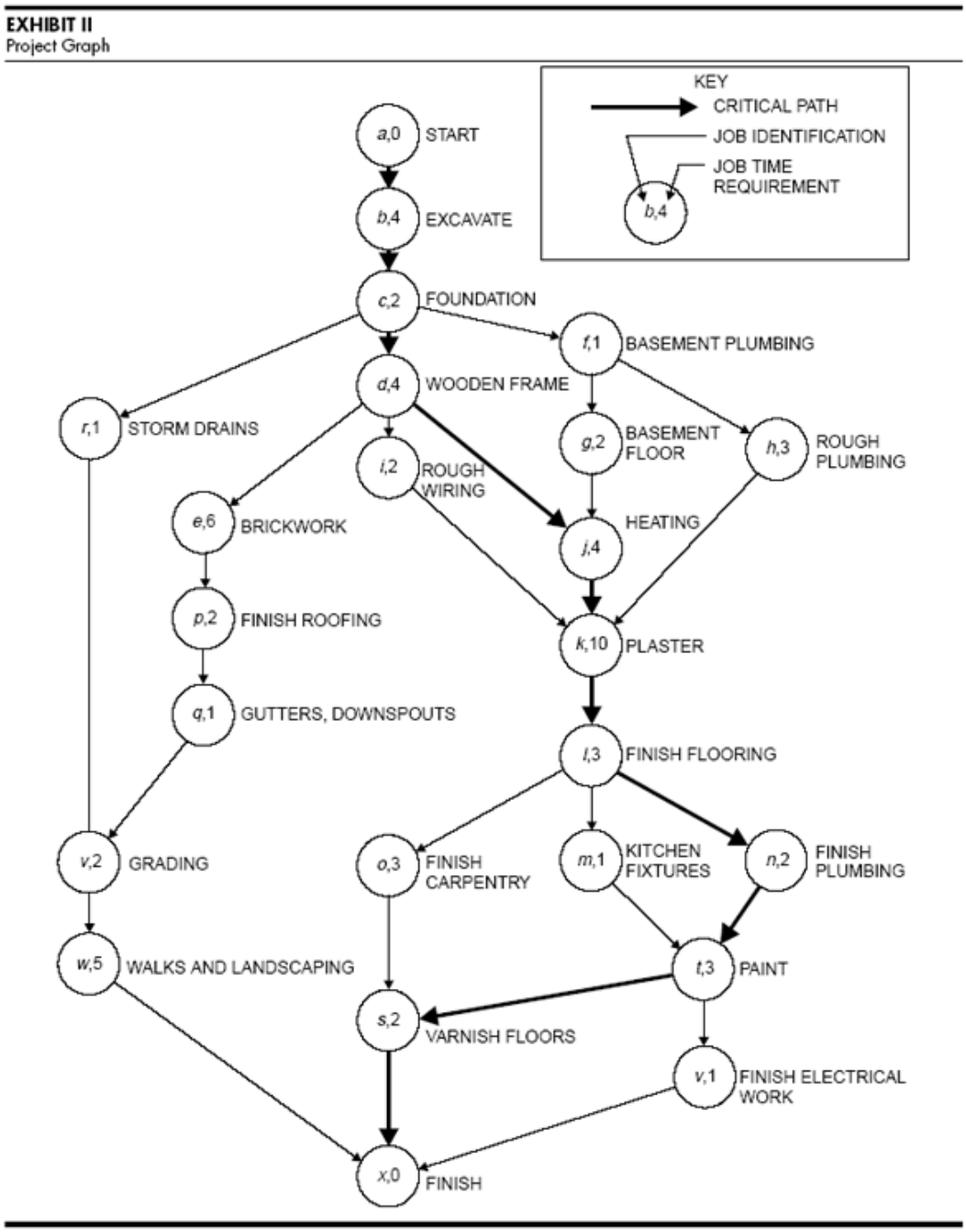


Exhibit II Project Graph

Following the rule that a "legal" path must always move in the direction of the arrows, we could enumerate 22 unique paths from Start to Finish, with associate times ranging from a minimum of 14 days (path *a-b-c-r-v-w-x*) to a maximum of 34 days (path *a-b-c-d-j-k-l-n-t-s-x*). The latter is the critical path; it determines the over-all project time and tells us which jobs are critical in their effect on this time. If the contractor wishes to complete the house in less than 34 days, it would be useless to shorten jobs not on the critical path. It may seem to him, for example, that the brickwork

(e) delays progress, since work on a whole series of jobs *(p-q-v-w)* must wait until it is completed. But it would be fruitless to rush the completion of the brickwork, since it is not on the critical path and so is irrelevant in determining total project time.

Shortening the CP

If the contractor were to use CPM techniques, he would examine the critical path for possible improvements. Perhaps he could assign more carpenters to job *d,* reducing it from four to two days. Then the critical path would change slightly, passing through jobs *f* and *g* instead of *d*. Notice that total project time would be reduced only one day, even though two days had been shaved off job *d*. Thus the contractor must watch for possible shifting of the critical path as he affects changes in critical jobs.

Shortening the critical path requires a consideration of both engineering problems and economic questions. Is it physically possible to shorten the time required by critical jobs (by assigning more men to the job, working overtime, using different equipment, and so on)? If so, would the costs of speedup be less than the savings resulting from the reduction in overall project time? CPM is a useful tool because it quickly focuses attention on those jobs that are critical to the project time, it provides an easy way to determine the effects of shortening various jobs in the project, and it enables the user to evaluate the costs of a "crash" program.

Two important applications of these features come to mind:

Du Pont, a pioneer in the application of CPM to construction and maintenance projects, was concerned with the amount of downtime for maintenance at its Louisville works, which produces an intermediate product in the neoprene process. Analyzing the maintenance schedule by CPM, Du Pont engineers were able to cut downtime for maintenance from 125 to 93 hours. CPM pointed to further refinements that were expected to reduce

total time to 78 hours. As a result, performance of the plant improved by about one million pounds in 1959, and the intermediate was no longer a bottleneck in the neoprene process.

PERT (i.e., Program Evaluation Review Technique), a technique closely related to the critical path method, is widely credited with helping to shorten by two years the time originally estimated for completion of the engineering and development program for the Navy's Polaris missile. By pinpointing the longest paths through the vast maze of jobs necessary for completion of the missile design, PERT enabled the program managers to concentrate their efforts on those activities that vitally affected total project time.[2]

Even with our small house-building project, however, the process of enumerating and measuring the length of every path through the maze of jobs is tedious. A simple method of finding the critical path and, at the same time, developing useful information about each job is described next.

Critical Path Algorithm

If the start time or date for the project is given (we denote it by S), then there exists for each job an earliest starting time (ES), which is the earliest possible time that a job can begin, if all its predecessors are also started at their ES. And if the time to complete the job is *t,* we can define, analogously, its earliest finish time (EF) to be ES + *t*.

There is a simple way of computing ES and EF times using the project graph. It proceeds as follows:

(1) Mark the value of S to the left and to the right of Start.

(2) Consider any new unmarked job *all of whose predecessors have been marked,* and mark to the left of the new job the *largest* number marked to the right of any of its *immediate* predecessors. This number is its early start time.

(3) Add to this number the job time and mark the result (EF time) to the right of the job.

(4) Continue until Finish has been reached, then stop.

Thus, at the conclusion of this calculation the ES time for each job will appear to the left of the circle which identifies it, and the EF time will appear to the right of the circle. The number which appears to the right of the last job, Finish, is the early finish time (F) for the entire project.

To illustrate these calculations let us consider the following simple production process:

An assembly is to be made from two parts, A and B. Both parts must be turned on the lathe, and B must be polished while A need not be. The list of jobs to be performed, together with the predecessors of each job and the time in minutes to perform each job, is given in Exhibit III.

EXHIBIT III
Data for Production Process

Job No.	Description	Immediate predecessors	Normal time (minutes)
a	Start		0
b	Get materials for A	*a*	10
c	Get materials for B	*a*	20
d	Turn A on lathe	*b,c*	30
e	Turn B on lathe	*b,c*	20
f	Polish B	*e*	40
g	Assemble A and B	*d,f*	20
h	Finish	*g*	0

Exhibit III Data for Production Process

The project graph is shown in Exhibit IV. AS previously, the letter identifying each job appears before the comma and its job time after the comma. Also shown on the graph are the ES and EF

times for each job, assuming that the start time, S, is *zero*. The ES time appears to the left of the circle representing a job, and the EF time appears to the right of the circle. Note that F = 100. The reader may wish to duplicate the diagram without these times and carry out the calculations for himself as a check on his understanding of the computation process described above.

EXHIBIT IV
Calculation of Early Start and Early Finish Times for Each Job

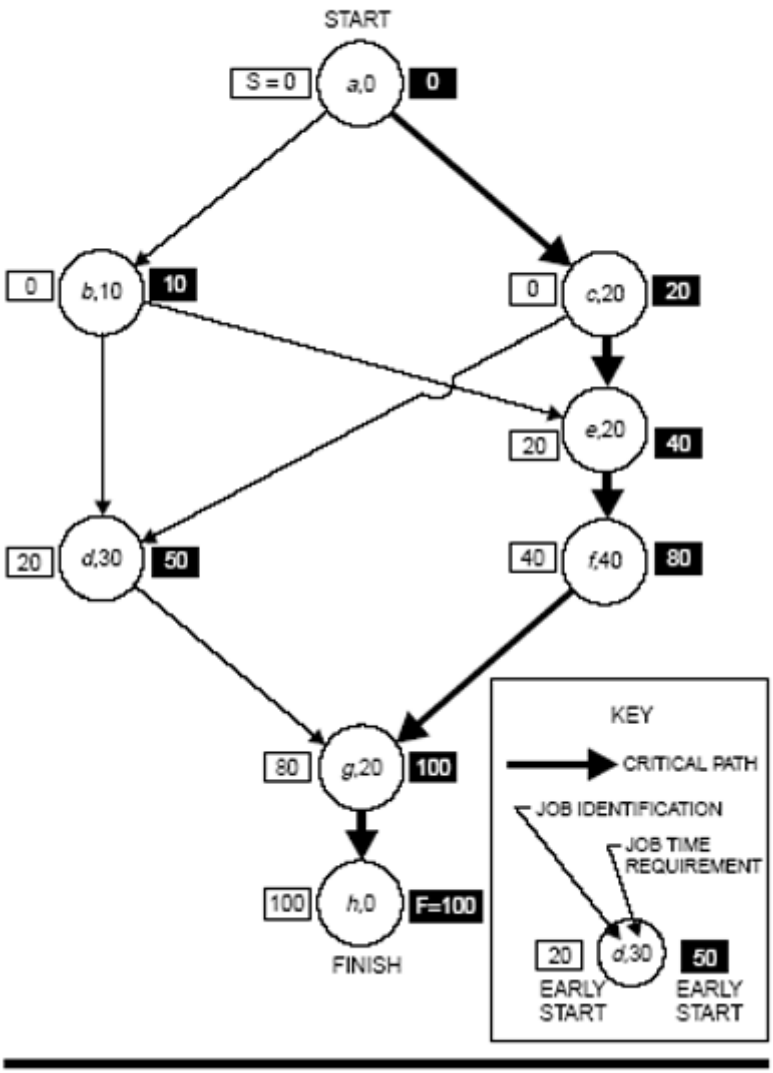


Exhibit IV Calculation of Early Start and Early Finish Times for Each Job

Latest Start & Finish Times

Suppose now that we have a target time (T) for completing the project. T may have been originally expressed as a calendar date, e.g., October 1 or February 15. When is the latest time that the project can be started and finished?

In order to be feasible it is clear that T must be greater (later) than or equal to F, the early finish time for the project. Assuming this is so, we can define the concept of late finish (LF), or the latest time that a job can be finished, without delaying the total project beyond its target time (T). Similarly, late start (LS) is defined to be LF—*t,* where *t* is the job time.

These numbers are determined for each job in a manner similar to the previous calculations except that we work from the end of the project to its beginning. We proceed as follows:

(1) Mark the value of T to the right and left of Finish.

(2) Consider any new unmarked job *all of whose successors have been marked,* and mark to the right of the new job the *smallest* LS time marked to the left of any of its immediate successors.

The logic of this is hard to explain in a few words, although apparent enough by inspection. It helps to remember that the smallest LS time of the successors of a given job, if translated into calendar times, would be the latest finish time of that job.

(3) Subtract from this number the job time and mark the result to the left of the job.

(4) Continue until Start has been reached, then stop.

At the conclusion of this calculation the LF time for a job will appear to the right of the circle which identifies it, and the LS time for the job will appear to the left of the circle. The number appearing to the right of Start is the latest time that the entire project can be started and still finish at the target time T.

In Exhibit V we carry out these calculations for the example of Exhibit III. Here T = F = 100, and we separate early start and finish and late start and finish times by semicolons so that ES; LS appears to the left of the job and EF; LF to the right. Again the reader may wish to check these calculations for himself.

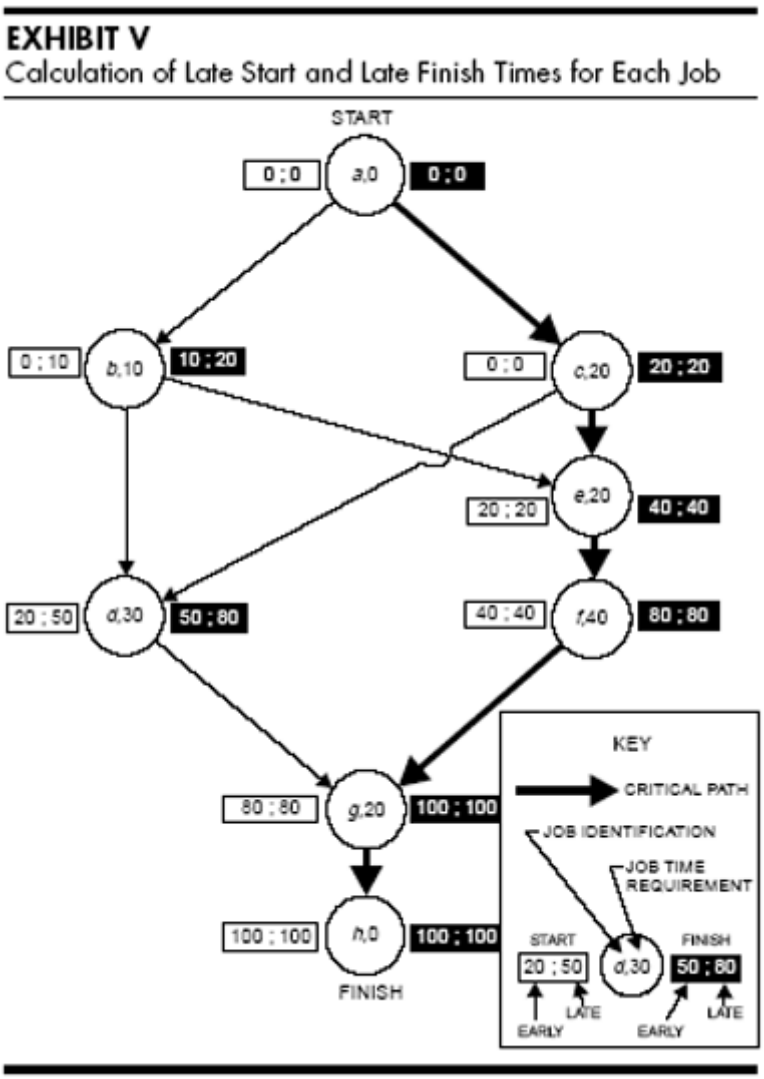


Exhibit V Calculation of Late Start and Late Finish Times for Each Job

Concept of Slack

Examination of Exhibit V reveals that some jobs have their early start equal to late start, while others do not. The difference between a job's early start and its late start (or between early finish and late finish) is called total slack (TS). Total slack

represents the maximum amount of time a job may be delayed beyond its early start without necessarily delaying the project completion time.

We earlier defined critical jobs as those on the longest path through the project. That is, critical jobs *directly* affect the total project time. We can now relate the critical path to the concept of slack.

Finding the Critical Path

If the target date (T) equals the early finish date for the whole project (F), then all critical jobs will have *zero* total slack. There will be at least one path going from Start to Finish that includes critical jobs only, i.e., the *critical path*.

If T is greater (later) than F, then the critical jobs will have total slack equal to T minus F. This is a minimum value; since the critical path includes only critical jobs, it includes those with the smallest TS. All noncritical jobs will have *greater* total slack.

In Exhibit V, the critical path is shown by darkening the arrows connecting critical jobs. In this case there is just one critical path, and all critical jobs lie on it; however, in other cases there may be more than one critical path. Note that T = F; thus the critical jobs have zero total slack. Job *b* has TS = 10, and job *d* has TS = 30; either or both of these jobs could be delayed by these amounts of time without delaying the project.

Another kind of slack is worth mentioning. Free slack (FS) is the amount a job can be delayed without delaying the early start of any other job. A job with positive total slack may or may not also have free slack, but the latter never exceeds the former. For purposes of computation, the free slack of a job is defined as the difference between the job's EF time and the *earliest* of the ES times of all its immediate successors. Thus, in Exhibit V, job *b* has FS of 10, and job *d* has FS of 30. All other jobs have zero free slack.

Significance of Slack

When a job has zero total slack, its scheduled start time is automatically fixed (that is, ES = LS); and to delay the calculated start time is to delay the whole project. Jobs with positive total slack, however, allow the scheduler some discretion in setting their start times. This flexibility can usefully be applied to smoothing work schedules. Peak loads that develop in a particular shop (or on a machine, or within an engineering design group, to cite other examples) may be relieved by shifting jobs on the peak days to their late starts. Slack allows this kind of juggling without affecting project time.[3]

Free slack can be used effectively at the operating level. For example, if a job has free slack, the foreman may be given some flexibility in deciding when to start the job. Even if he delays the start by an amount equal to (or less than) the free slack, the delay will not affect the start times or slack of succeeding jobs (which is not true of jobs that have no free slack). For an illustration of these notions, we return to our house-building example.

Back to the Contractor

In Exhibit VI, we reproduce the diagram of house-building jobs, marking the ES and LS to the left, and the EF and LF to the right of each job (for example, "0;3" and "4;7" on either side of the *b,* 4 circle). We assume that construction begins on day zero and must be completed by day 37. Total slack for each job is not marked, since it is evident as the difference between the pairs of numbers ES and LS or EF and LF. However, jobs that have positive free slack are so marked. There is one critical path, which is shown darkened in the diagram. All critical jobs on this path have total slack of three days.

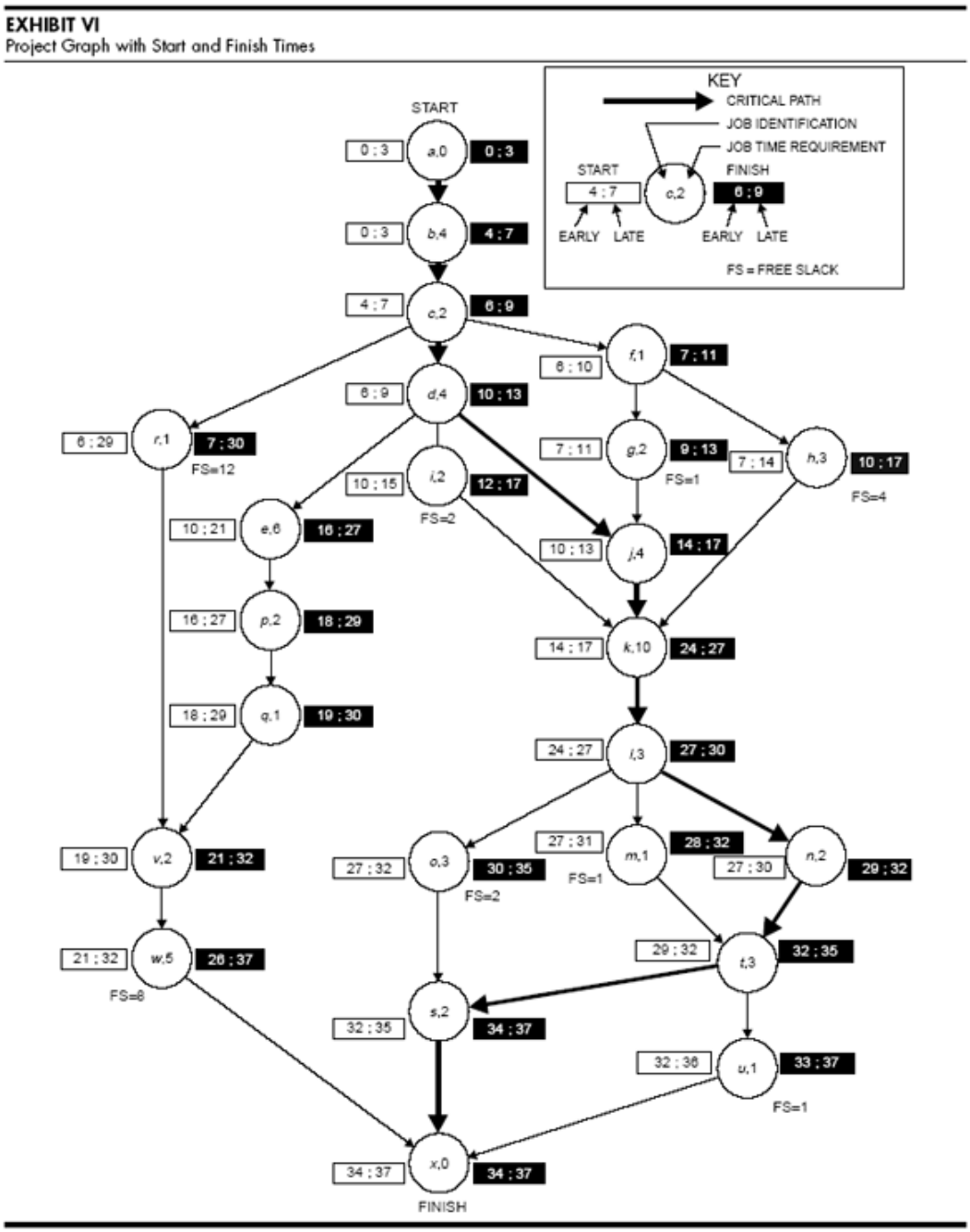


Exhibit VI Project Graph with Start and Finish Times

Several observations can be drawn immediately from the diagram:

(1) The contractor could postpone starting the house three days and still complete it on schedule, barring unforeseen difficulties (see the difference between early and late times at the Finish). This would reduce the total slack of all jobs by three days, and hence reduce TS for critical jobs to zero.

(2) Several jobs have free slack. Thus the contractor could delay the completion of *i* (rough wiring) by two days, *g* (the basement floor) by one day, *h* (rough plumbing) by four days, *r* (the storm drains) by 12 days, and so on—without affecting succeeding jobs.

(3) The series of jobs *e* (brickwork), *p* (roofing), *q* (gutters), *v* (grading), and *w* (landscaping) have a comfortable amount of total slack (nine days). The contractor can use these and other slack jobs as "fill in" jobs for workers who become available when their skills are not needed for currently critical jobs. This is a simple application of workload smoothing: juggling the jobs with slack in order to reduce peak demands for certain skilled workers or machines.

If the contractor were to effect changes in one or more of the critical jobs, by contrast, the calculations would have to be performed again. This he can easily do; but in large projects with complex sequence relationships, hand calculations are considerably more difficult and liable to error. Computer programs have been developed, however, for calculating ES, LS, EF, LF, TS, and FS for each job in a project, given the set of immediate prerequisites and the job times for each job.[4]

Handling Data Errors

Information concerning job times and predecessor relationships is gathered, typically, by shop foremen, scheduling clerks, or others closely associated with a project. It is conceivable that several kinds of errors may occur in such job data:

1. The estimated job times may be in error.

2. The predecessor relationship may contain cycles: e.g., job *a* is a predecessor for *b, b* is a predecessor for *c,* and *c* is a predecessor for *a*.

3. The list of prerequisites for a job may include more than the immediate prerequisites; e.g., job a is a predecessor of *b, b* is a predecessor of *c,* and a and *b* both are predecessors of *c*.

4. Some predecessor relationships may be overlooked.

5. Some predecessor relationships may be listed that are spurious.

How can management deal with these problems? We shall examine each briefly in turn.

Job Times. An accurate estimate of total project time depends, of course, on accurate job-time data. CPM eliminates the necessity (and expense) of careful time studies for *all* jobs. Instead the following procedure can be used:

- Given rough time estimates, construct a CPM graph of the project.

- Then those jobs that are on the critical path (together with jobs that have very small total slack, indicating that they are nearly critical) can be more closely checked, their times re-estimated, and another CPM graph constructed with the refined data.

- If the critical path has changed to include jobs still having rough time estimates, then the process is repeated.

In many projects studied, it has been found that only a small fraction of jobs are critical; so it is likely that refined time studies will be needed for relatively few jobs in a project in order to arrive at a reasonably accurate estimate of the total project time. CPM thus can be used to reduce the problem of Type I errors at a small total cost.

Prerequisites. A computer algorithm has been developed to check for errors of Types 2 and 3 above. The algorithm (mentioned in footnote 4) systematically examines the set of prerequisites for each job and cancels from the set all but immediate predecessor jobs. When an error of Type 2 is present in the job data, the algorithm will signal a "cycle error" and print out the cycle in question.

Wrong or Missing Facts. Errors of Types 4 and 5 cannot be discovered by computer routines. Instead, manual checking (perhaps by a committee) is necessary to see that prerequisites are accurately reported.

Cost Calculations

The cost of carrying out a project can be readily calculated from the job data if the cost of doing each job is included in the data. If jobs are done by crews, and the speed with which the job is done depends on the crew size, then it is possible to shorten or lengthen the project time by adding or removing men from crews. Other means for compressing job times might also be found; but any speedup is likely to carry a price tag. Suppose that we assign to each job a "normal time" and a "crash time" and also calculate the associated costs necessary to carry the job in each time. If we want to shorten the project, we can assign some of the critical jobs to their crash time, and compute the corresponding direct cost. In this way it is possible to calculate the cost of completing the project in various total times, with the direct costs increasing as the over-all time decreases.

Added to direct costs are certain overhead expenses which are usually allocated on the basis of total project time. Fixed costs per project thus decrease as project time is shortened. In ordinary circumstances a combination of fixed and direct costs as a function of total project time would probably fall into the pattern shown in Exhibit VII. The minimum total cost (point A) would

likely fall to the left of the minimum point on the direct cost curve (point B) indicating that the optimum project time is somewhat shorter than an analysis of direct costs only would indicate.

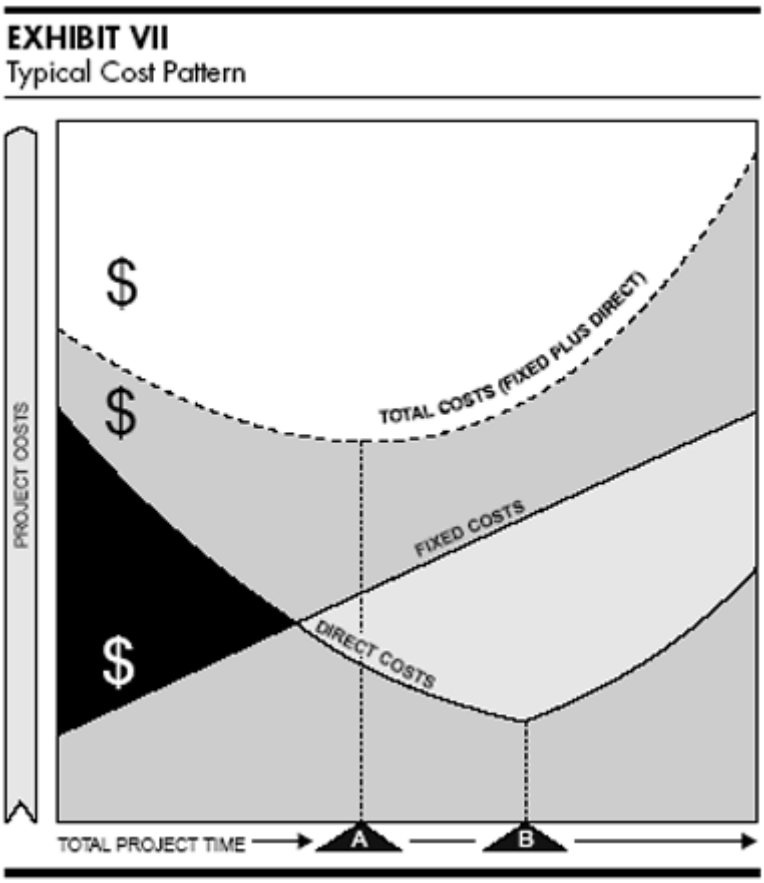


Exhibit VII Typical Cost Pattern

Other economic factors, of course, can be included in the analysis. For example, pricing might be brought in:

A large chemical company starts to build a plant for producing a new chemical. After the construction schedule and completion date are established, an important potential customer indicates a willingness to pay a premium price for the new chemical if it can be made available earlier than scheduled. The chemical producer applies techniques of CPM to its construction schedule and calculates the additional costs associated with “crash” completion of jobs on the critical path. With a plot of costs correlated with

total project time, the producer is able to select a new completion date such that the increased costs are met by the additional revenue offered by the customer.

New Developments

Because of their great potential for applications, both CPM and PERT have received intensive development in the past few years. This effort is sparked, in part, because of the Air Force (and other governmental agency) requirements that contractors use these methods in planning and monitoring their work. Here are some illustrations of progress made:

One of the present authors (Wiest) has developed extensions of the work-load smoothing algorithm. These extensions are the so-called SPAR (for Scheduling Program for Allocating Resources) programs for scheduling projects having limited resources.

A contemporaneous development by C-E-I-R, Inc., has produced RAMPS (for Resource Allocation and Multi-Project Scheduling), which is similar but not identical.

The most recent version of PERT, called PERT/COST, was developed by the armed services and various businesses for use on weapon-systems development projects contracted by the government. Essentially, PERT/COST adds the consideration of resource costs to the schedule produced by the PERT procedure. Indications of how smoothing can be accomplished are also made. Other recent versions are called PERT II, PERT III, PEP, PEPCO, and Super PERT.

Conclusion

For the manager of large projects, CPM is a powerful and flexible tool, indeed, for decision making:

- It is useful at various stages of project management, from initial planning or analyzing of alternative programs, to scheduling and controlling the jobs (activities) that comprise a project.

- It can be applied to a great variety of project types—from our house-building example to the vastly more complicated design project for the Polaris—and at various levels of planning—from scheduling jobs in a single shop, or shops in a plant, to scheduling plants within a corporation.

- In a simple and direct way it displays the interrelations in the complex of jobs that comprise a large project.

- It is easily explainable to the layman by means of the project graph. Data calculations for large projects, while tedious, are not difficult, and can readily be handled by a computer.

- It pinpoints attention to the small subset of jobs that are critical to project completion time, thus contributing to more accurate planning and more precise control.

- It enables the manager to quickly study the effects of "crash" programs and to anticipate potential bottlenecks that might result from shortening certain critical jobs.

- It leads to reasonable estimates of total project costs for various completion dates, which enable the manager to select an optimum schedule.

Because of the above characteristics of CPM—and especially its intuitive logic and graphic appeal—it is a decision-making tool which can find wide appreciation at all levels of management.[5] The project graph helps the foreman to understand the sequencing of jobs and the necessity of pushing those that are critical. For the manager concerned with day-to-day operations in all departments, CPM enables him to measure progress (or lack of it) against plans and to take appropriate action quickly when

needed. And the underlying simplicity of CPM and its ability to focus attention on crucial problem areas of large projects make it an ideal tool for the top manager. On his shoulders falls the ultimate responsibility for over-all planning and coordination of such projects in the light of company-wide objectives.

1. *Proceedings of the Eastern Joint Computer Conference,* Boston, December 1–3, 1959; see also James E. Kelley, Jr., "Critical-Path Planning and Scheduling: Mathematical Basis," *Operations Research,* May–June 1961, pp. 296–320.

2. See Robert W. Miller, "How to Plan and Control With PERT," HBR March–April 1962, p. 93.

3. For a method for smoothing operations in a job shop, based on CPM and the use of slack, see F.K. Levy, G.L. Thompson, and J.D. Wiest, "Multi-Ship, Multi-Shop Production Smoothing Algorithm," *Naval Logistics Research Quarterly,* March 9, 1962.

4. An algorithm on which one such computer program is based is discussed by F.K. Levy, G.L. Thompson, and J.D. Wiest, in chapter 22, "Mathematical Basis of the Critical Path Method," *Industrial Scheduling* (see Authors' Note).

5. See A. Charnes and W.W. Cooper, "A Network Interpretation and a Directed Sub-Dual Algorithm for Critical Path Scheduling," *Journal of Industrial Engineering,* July–August 1962, pp. 213–219.

A version of this article appeared in the September 1963 issue of *Harvard Business Review*.

FL

Ferdinand K. Levy has just become Assistant Professor at Stanford University.

GT

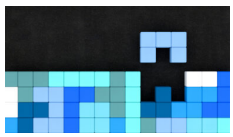
Gerald L. Thompson is Professor of Applied Mathematics and Industrial Administration, Carnegie Institute of Technology.

JW

Jerome D. Wiest is Assistant Professor, University of California at Los Angeles.

Recommended For You

The Four Phases of Project Management



Decision Trees for Decision-Making

3 Simple Habits to Improve Your Critical Thinking

AUDIO
Improve Your Critical Thinking at Work