PERT AND SIMULATION

David E. Douglas

ABSTRACT

PERT (Program Evaluation and Review Technique) is a network planning technique used to plan, schedule, and control projects. Unlike the CPM (Critical Path Method) which assumes actual project activity times are deterministic, PERT views the actual performance time for an activity as a random variable. The conventional PERT procedure ignores all subcritical paths which leads to an optimistically biased estimate of the expected earliest occurrence time for the network events. The most promising approach to solving this problem (called the merge event bias problem) appears to be simulation.

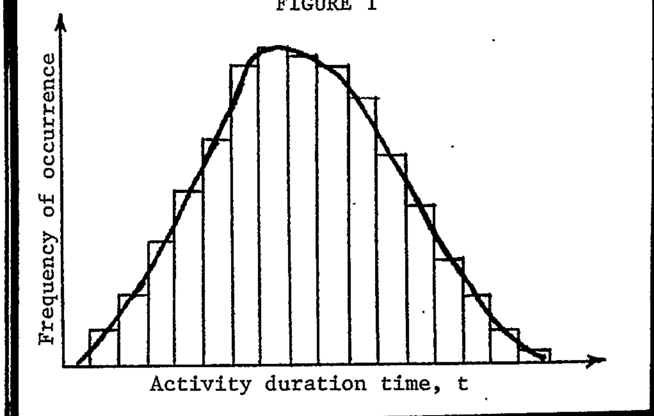
INTRODUCTION TO THE PERT STATISTICAL APPROACH

Unlike the traditional scheduling systems of using a fixed time for each task, the PERT statistical procedure utilizes probability theory for managerial decision making. In the PERT system three time estimates are obtained for each activity—an optimistic, a pessimistic, and a most likely time. This range of times provides a measure of the uncertainty associated with the actual time required to perform the activity sometime in the future. It is possible to derive the probabilities of finishing a project on or before scheduled dates of the probabilities of finishing milestone events on or before scheduled dates. These statements of the possible range of times and the probabilities associated with each results in a meaningful and potentially useful management tool.

EMPERICAL FREQUENCY DISTRIBUTIONS

To provide a basic background in probability and statistics, is is logical to begin with observations from some measurable quality subject to random or chance variation. Consider, for example, an activity that has been performed a large number of times under essentially the same conditions. If one counts the number of times the activity required for each duration time, the resulting data can be displayed in a frequency distribution as shown in Figure 1. If an infinite number of observations could be taken and the width of the time intervals are narrowed to approach zero, the distribution would merge into some smooth curve. This curve is referred to as the theoretical probability density of the random variable. Since the distribution represents the proportion of time that specified activity duration times occur, the total area under the curve is exactly one. Thus, the area under the curve between any two values of t directly provides the probability that the random variable (activity duration time) will fall in this area.

FIGURE 1



MEAN AND STANDARD DEVIATIONS OF DISTRIBUTIONS

The mean and standard deviation are the two most common measures to describe an empirical frequency distribution quantitatively. The mean is a measure of central tendency to location and the standard deviation measures the spread or dispersion in the distribution. These measures are illustrated in Figure 2.

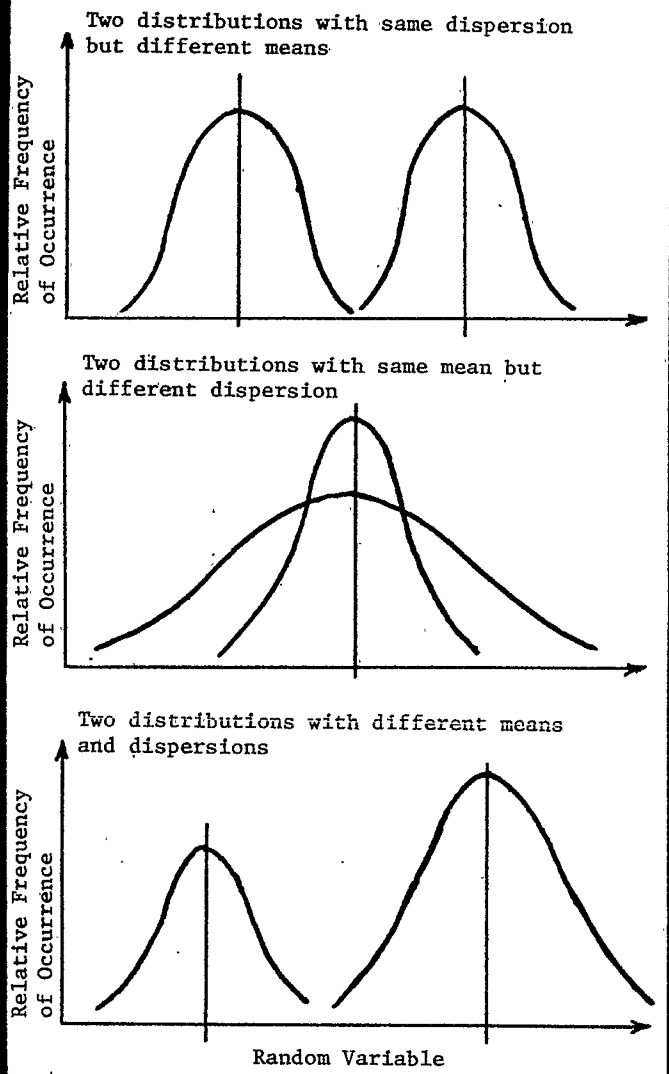
If a sample of n observations are taken from a distribution such as the one shown in Figure 1, and the n observations are denoted by $t_1, t_2, \ldots, t_n$, then the mean and standard deviation are defined as follows:

$$\text{mean} = \bar{t} = \frac{t_1 + t_2 + \ldots + t_n}{n} = \frac{\sum t}{n}$$

$$\text{standard deviation} = s_t = \sqrt{\frac{(t_1-\bar{t})^2+(t_2-\bar{t})^2+\ldots+(t_n-\bar{t})^2}{n-1}} = \sqrt{\frac{\sum(t-\bar{t})^2}{n-1}}$$

The variance, the standard deviation squared, is also required for the PERT method.

FIGURE 2



CENTRAL LIMIT THEOREM

Perhaps the most important theorem in all of statistics is the Central Limit Theorem. In the PERT context Moder and Phillips define it as follows:

"Suppose m independent tasks are to be performed in order; (one might think of these as the m tasks which lie on the critical path of a network). Let t_1, t_2,...t_n be the times at which these tasks are actually completed. Note that these are random variables with true means $t_{e1}, t_{e2}, \ldots t_{e3}$, and true variances V_{t1}, $V_{t2}, \ldots V_{t3}$, and these specific tasks are unknown until these specific tasks are actually performed. Now define T to be the sum:

$$T = t_1 + t_2 + \ldots + t_m$$

and note that T is also a random variable and thus has a distribution. The Central Limit Theorem states that if m is large, say four or more, the distribution of T is approximately normal with mean E and variance V_t given by

$$E = t_{e1} + t_{e2} + \ldots + t_{em}$$

$$V_t = V_{t1} + V_{t2} + \ldots + V_{tm} \text{ " } (11)$$

The normal distribution is a well known distribution which has a characteristic symmetrical bell shape as shown in Figure 3. Other areas under the normal curve can be looked up in a normal curve table in any statistics book. Readers who have not had a basic course in statistics may wish to refer to a more complete treatment of this subject given in text books on statistics. (5, 10)

FIGURE 3

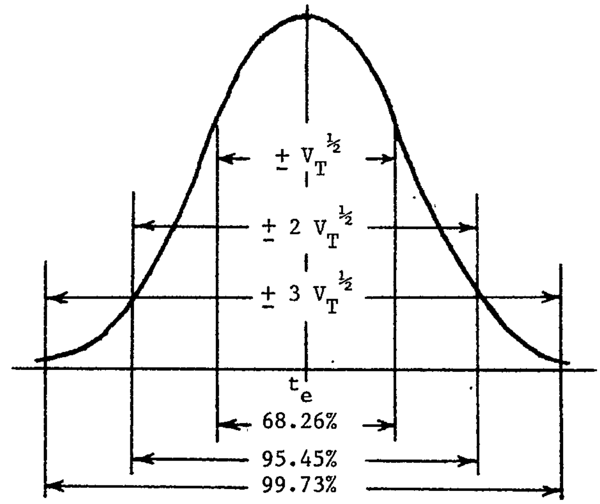


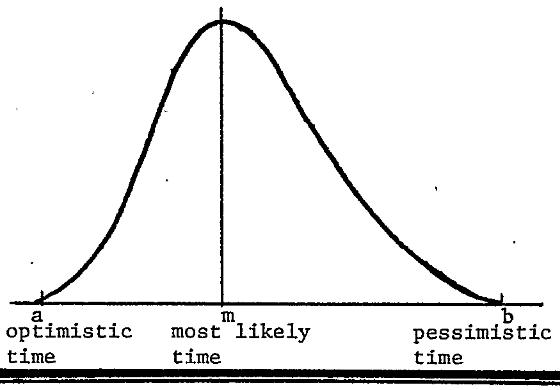
ILLUSTRATION OF THE "CONVENTIONAL"
PERT STATISTICAL APPROACH

The PERT system utilizes the expected values, t_e from hypothetical distributions which have already been illustrated. Since PERT is used primarily for projects whose activities are subject to considerable variability, it also utilized the standard deviations. The traditional method of obtaining an estimate for expected activity duration time t_e, and the standard deviation, s_t requires three time estimates for each activity:

a = optimistic performance time
b = most likely performance time
c = pessimistic performance time

Figure 4 illustrates a hypothetical distribution and three time estimates. Note this figure shows a and b as the original 0 and 100 percentiles of the distribution. Moder and Phillips (11) propose the use of 5 and 95 percentiles.

FIGURE 4



The equations for estimating the mean, variance, and standard deviation using the estimates of a, m, and b are:

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

$$s_t = V_T^{\frac{1}{2}} = \sqrt{\left(\frac{b - a}{6}\right)^2}$$

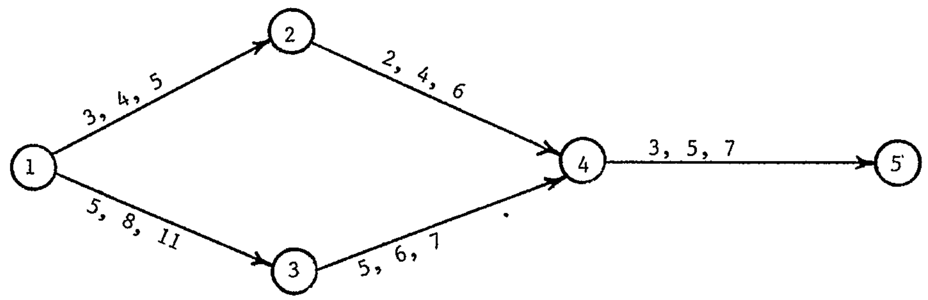
THE CRITICAL PATH

Consider the network shown in Figure 5 with its corresponding estimates of a, m, and b. Defining the critical path as the longest path through the network, the forward path computations can be calculations as in Figure 5. Note the expected activity time durations, t_e, are summed along the critical path and the variance to any node (circle) is also summed along the critical path for merge events such as 4. The Central Limit Theorem provides a normal distribution with mean of 20 and variance of 1.555 for completion of this particular network. With this information, probability computations are possible for any range of desired completion times. For example, the probability of completing on or before day 21 (computing Z and looking up in normal tables) is:

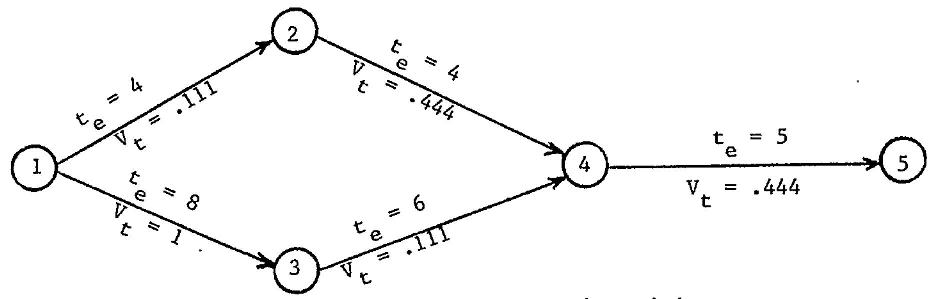
$$Z = \frac{21-20}{\sqrt{1.555}} = 0.80, \quad \text{Probability} = 0.7881 \text{ or } 79\%$$

Of course, probabilities for other values of interest are possible.

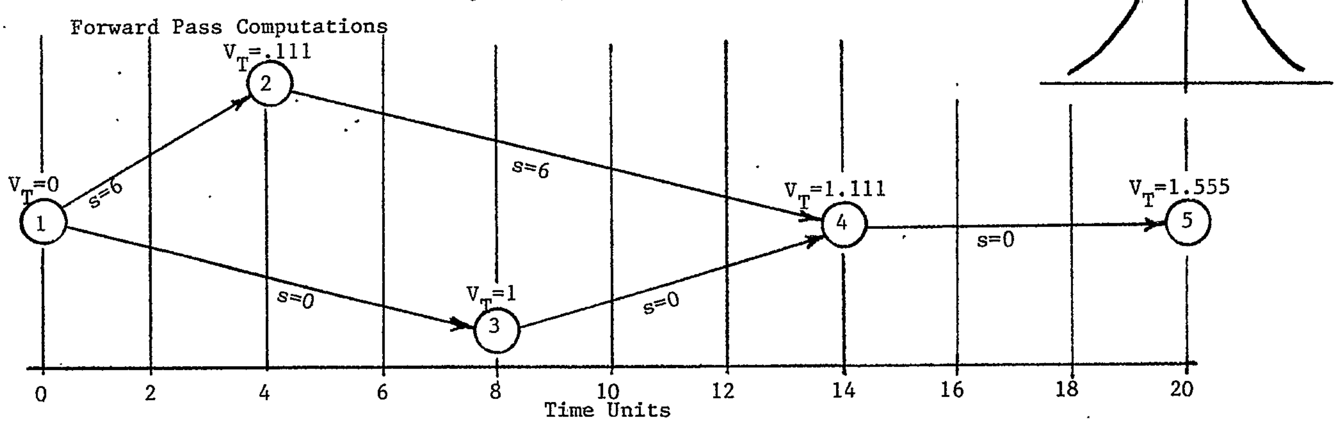
FIGURE 5



Basic network with three time estimates, a, m, and b



Basic network with t_e and V_t computed for each activity



THE MERGE EVENT BIAS PROBLEM

In the conventional PERT approach, all subcritical paths are ignored in making the critical path calculations. Because of this, the earliest (expected) occurrence time for the network is always optimistically biased. However, if the longer path leading to a merge event is much longer than the second longest path and/or the variance of the activities on the longest path is small, the bias can be ignored. For example, consider the PERT example in Figure 5. The longest path is much longer than the other path, 14 versus 8, therefore the bias will be insignificant and can be ignored.

MAGNITUDE OF BIAS

One notable study of the magnitude of bias was made by MacCrimmon and Ryavec (8). They considered two of the more important factors that affect the bias, the number of parallel paths through a portion of a network, and the closeness of the expected finish times at merge events of the parallel paths. MacCrimmon and Ryavec (8) illustrated the effect of the first of these two factors with the networks shown in Figure 6.

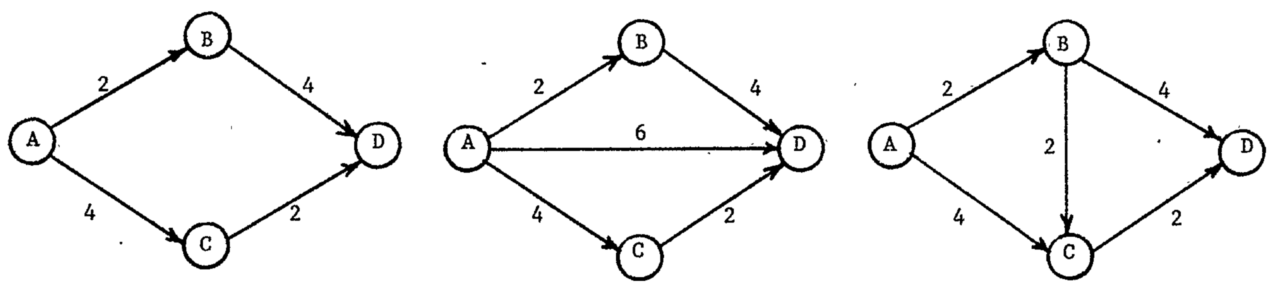
The particular discrete distribution for each of the activities in Figure 6 is shown below and can be identified by the corresponding mean shown on the network activities.

t	Probability	t	Probability
1	1/4	2	1/4
2	1/2	4	1/2
3	1/4	6	1/4
Mean = t_e = 2		Mean = t_e = 4	
Std Dev = V_t = 0.707		Std Dev = $V_t^{1/2}$ = 1.414	

Although an extreme case in that all the parallel paths are equal, one can conclude that the bias increases as the number of parallel paths increase.

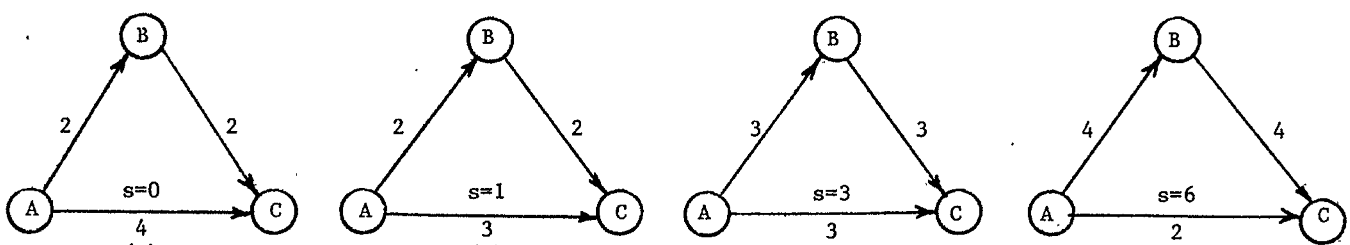
The effect of slack on Merge Event Bias is illustrated in Figure 7. All activities in Figure 7 are assumed to be normally distributed with standard deviation equal to 1 and mean as shown. From this example, one can conclude that the bias increases as the length of the parallel paths become equal. Other Merge Event Bias studies have been conducted by Klingel (7), Clark (2), Fulkenson (6), Clingen (3), Elmaghraby (4), and Charnes and Cooper (1).

FIGURE 6



PERT(mean)	6.00	6.00	6.00
Exact(mean)	6.89	7.34	7.07
Error	-12.9%	-18.2%	-15.2%

FIGURE 7



Ratio of Lengths path AC / path ABC	(a) 1/1	(b) 3/4	(c) 1/2	(d) 1/4
PERT(mean)	4.00	4.00	6.00	8.00
Exact(mean)	4.69	4.30	6.03	8.00
Error	-17%	-8%	-.5%	0.0%

RULES OF THUMB

From a table of studies derived by Clark (2), (giving the expected value of the greatest of a finite set of random variables), a useful rule of thumb is stated as in Moder and Phillips:

"If the difference between the expected complete times of the two merging activities being considered is greater than the larger of their respective standard deviations, then the bias correction will be small; if the difference is greater than two standard deviations, the bias will be less than a few percent and can be ignored. If there are more than two merging activities, this rule should be applied to the two with the latest expected finish times." (11)

Figure 7 depicts the validity of this rule. In part b, the difference between the expected finishes is less than one standard deviation and is greater than two standard deviations in part c. The corresponding bias is 8% and 0.5% respectively.

If the rule of thumb indicates a correction for bias is required, some method should be implemented. Although several analytical methods have been proposed, simulation seems to have more promise for practical networks.

SIMULATION APPROACH TO MERGE EVENT BIAS PROBLEM

The simulation of a network not only gives unbiased estimates of the mean and variance of the project duration (along with the distribution of total project time), but also gives a "criticality" of an activity (the probability of an activity being on the critical path). This useful measure allows management to focus attention to activities with a high criticality index. It should be noted that the probability of an activity being on the critical path does not correlate too well with slack as calculated using the conventional PERT scheme.

Since simulation appears to be the most promising solution to the Merge Event Bias problem, this paper will demonstrate this simulation methodology using a procedural programming language (FORTRAN), and a special simulation language, GERT (Graphical Evaluation and Review Technique), designed to work with networks. Although not well suited for network simulation, the popular simulation language GPSS (General Purpose Simulation System) can be used to simulate networks. For an example of simulation of a network using GPSS, see Schriber (15). Another popular simulation language, GASP IV, can be used to simulate networks. See Pritsker (13) for an example.

NETWORK SIMULATION USING FORTRAN

For illustrative purposes, the network shown in Figure 8 was simulated 10,000 times. Each activity was assumed to be normally distributed with means and standard deviations as shown. Rather large standard deviations were used to demonstrate the wide range of possible finish times. Figure 9 shows the FORTRAN program and the output. The output provides the number of times (out of 10,000) that each activity was on the cirtical path, thus, providing the probability of each activity being on the critical path. The distribution of finish times for the network is also provided. For this example, the mean finish time is 77.74 and standard deviation of 7.62. Assuming a normal distribution the range of finish times can be from 77.74 - 3(7.62) to 77.74 + 3(7.62) or from 54.88 to 100.60.

FIGURE 8

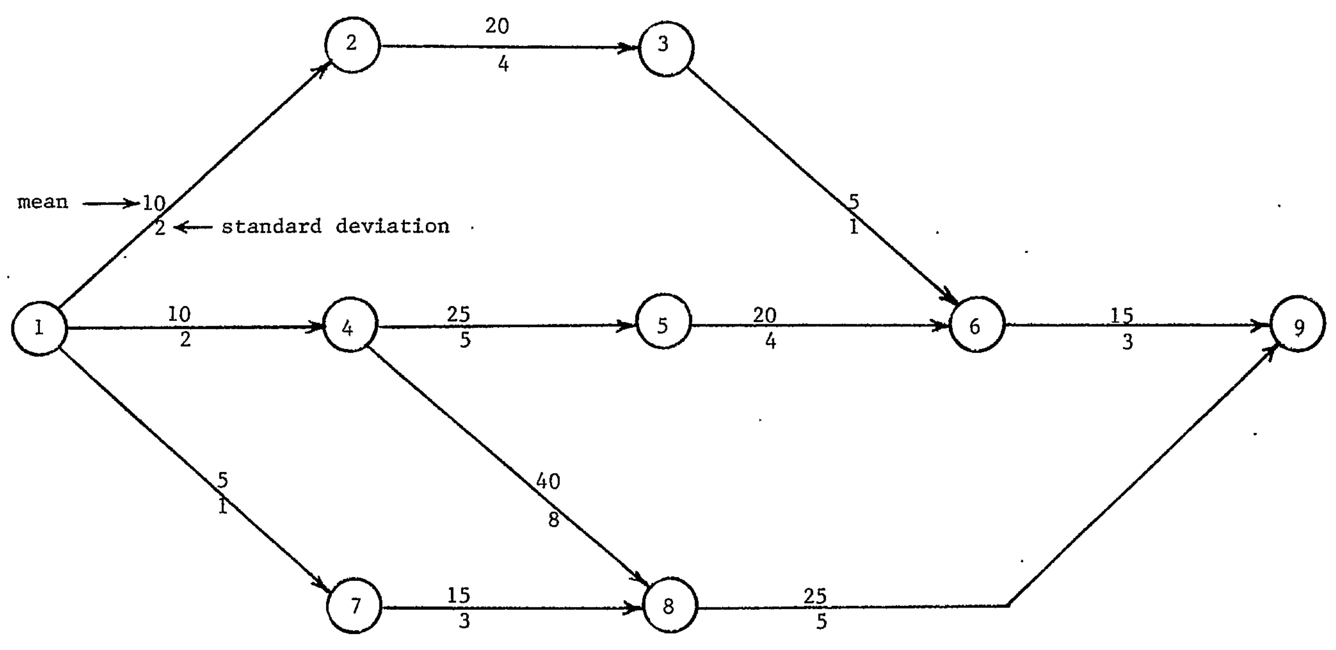


FIGURE 9

```
                REAL NODE(100),LS(100),LF(100)
                DIMENSION I(100),J(100),TIME(100),NMCT(100),ES(100),EF(100),
               1          TOTSLK(100),XMEAN(100),SD(100),FIN(10000)
                ISEED=4262923
                READ(5,501)NN,NACT
            501 FORMAT(2I5)
                WRITE(6,501)NN,NACT
                DO 25 L=1,NACT
                READ(5,502)I( L),J(L),XMEAN(L),SD(L)
                WRITE(6,502)I(L),J(L),XMEAN(L),SD(L)
            502 FORMAT(2I5 ,2F10.0)
                NMCT(L)=0
             25 CONTINUE
          C---------SIMULATE FOR 10000 TIMES
                DO 600 L=1,10000
          C-----------GET VALUE FROM NORMAL
                DO 30 K=1,NACT
                R1=RANDU(ISEED)
                R2=RANDU(ISEED)
                V1=(-2.0*ALOG(R1))**0.5*COS(6.283*R2)
                TIME(K)=V1*SD(K)+XMEAN(K)
             30 CONTINUE
                TOTMAX=0.
          C---------SET TIME AT NODES EQUAL TO ZERO
                DO 100 K=1,NN
            100 NODE(K)=0.0
          C---------FORWARD PASS TO GET EARLY STARTS FOR NODES
                DO 150 K=1,NACT
                INODE=I(K)
                JNODE=J(K)
                ES(K)=NODE(INODE)
                TEMP=NODE(INODE)+TIME(K)
                IF(TEMP .GT. NODE(JNODE))NODE(JNODE)=TEMP
            150 CONTINUE
          C---------LOOP TO GET EARLY FINISHES, CALCULATE EF
                DO 200 K=1,NACT
                EF(K)=ES(K)+TIME(K)
                IF(EF(K) .GT. TOTMAX)TOTMAX=EF(K)
            200 CONTINUE
                FIN(L)=TOTMAX
          C---------SET ALL NODES TO EARLY FINISH OF NETWORK
          C---------THEN MAKE BACKWARD PASS
                DO 250 K=1,NN
            250 NODE(K)=TOTMAX
                DO 300 K=1,NACT
                M=NACT+1-K
                INODE=I(M)
                JNODE=J(M)
                LS(M)=NODE(JNODE)-TIME(M)
                TEMP=NODE(JNODE)-TIME(M)
                IF(TEMP .LE. NODE(INODE))NODE(INODE)=TEMP
            300 CONTINUE
          C---------CALCULATE LATE FINISHES
          C---------CALCULATE TOTAL SLACK
                DO 400 K=1,NACT
                LF(K)=LS(K) + TIME(K)
                TOTSLK(K)=LF(K)-EF(K)
                IF(LF(K) .EQ. EF(K))TOTSLK(K)=0.
                IF(TOTSLK(K) .LE. 0.0)NMCT(K)=NMCT(K)+1
            400 CONTINUE
            600 CONTINUE
                WRITE(6,601)
            601 FORMAT(1H1,'I-NODE     J-NODE     NUM-ON-CP')
                DO 650 K=1,NACT
                WRITE(6,602)I(K),J(K),NMCT(K)
            602 FORMAT(1X,I6,5X,I6,5X,I9)
```

FIGURE 9 (Continued)

```
  650 CONTINUE
      X=10000.
      SUMSQ=0.
      SUM=0.
      DO 750 L=1,10000
      SUM=SUM+FIN(L)
      SUMSQ=SUMSQ+FIN(L)**2
  750 CONTINUE
      AVG=SUM/X
      STDEV=SQRT((X*SUMSQ-SUM*SUM)/(X*(X-1.)))
      WRITE(6,605)AVG,STDEV
  605 FORMAT(//////////,1X,'AVERAGE=',F10.2,
     1   //,1X,'STANDARD DEVIATION=',F10.2)
      STOP
      END

      FUNCTION RANDU(ISEED)
      IY=ISEED*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      ISEED=IY
      RANDU=YFL*0.4656613E-09
      RETURN
      END
```

I-NODE	J-NODE	NUM-ON-CP
1	2	6
1	4	9994
1	7	0
2	3	6
3	6	6
4	5	3292
4	8	6702
5	6	3292
6	9	3298
7	8	0
8	9	6702

AVERAGE= 77.74

STANDARD DEVIATION= 7.62

NETWORK SIMULATION USING GERT

One simulation approach which appears to be free of the shortcomings of the conventional PERT statistical system is GERT (Graphical Evaluation and Review Technique). GERT, developed by Alan Pritsker, is a technique for analyzing stochastic networks, and differs in many respects from PERT. Notable among the differences are the following.

* PERT requires that all activities be completed before the project can be completed (deterministic branching), while GERT associates with each branch a probability that the branch will be taken (probabilistic branching).

* PERT requires that all activities leading to a node be completed before the node is realized, while GERT allows the user to specify the number of required activity completions before the node is realized. This number of completions may be less than, equal to, or greater than the number of activities terminating at a node (see looping below).

* PERT allows no activity to be repeated (no looping), while GERT allows looping (as in rework, for a production network).

* PERT allows only one outcome or project completion node, while GERT recognizes the possibility of multiple outcomes (such as success or failure).

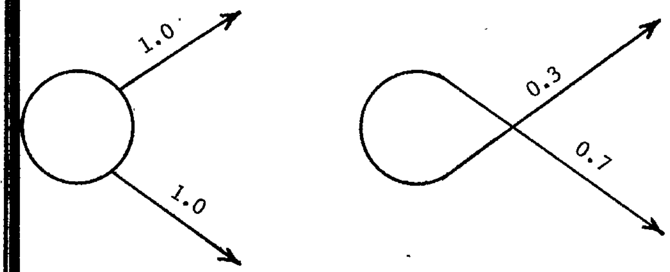
* In PERT, the critical path is always the path with the longest expected elapsed time, even though it is recognized that variation in the activity times does exist, as evidenced by the use of the three time estimates. When GERT is used for a "PERT" network, paths other than the PERT "critical path" may become critical.

GERT SYMBOLOGY

The power of GERT is shown in the symbols used in the technique. The basics of the symbology will be given by way of review and to show contrast with PERT.

The type of node symbol most familiar to PERT users is the round node, with branches leading from it. This symbol is used in the same manner with GERT to indicate a deterministic branch where all activities leaving the node must be taken, as shown in Figure 10, on the left.

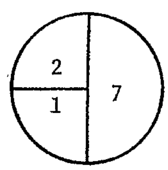
FIGURE 10



By contrast, the node on the right in Figure 10 indicates a probabilistic branch. It is drawn with a point on the output side and the probabilities emanating from the node sum to one. On any one pass through the node, only one branch may be taken.

The input side of a GERT node likewise has some distinct symbols attached to it. This is illustrated in Figure 11.

FIGURE 11

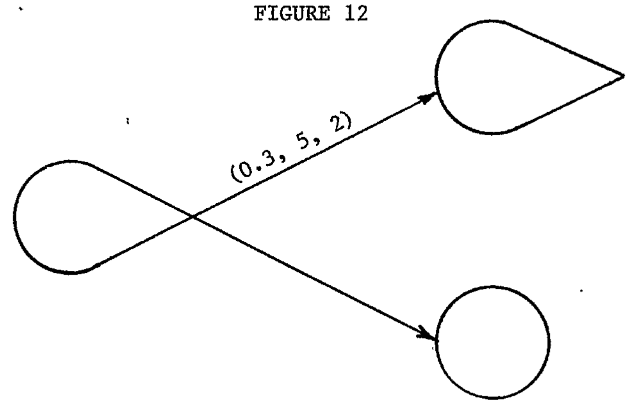


As discussed before, the roundness of the right side of the node indicates a deterministic node. The number in the upper left (2 in Figure 11) is the number of activities leading into the node that must be completed before the node is realized for the first time. The number in the lower left (1 in Figure 11) is the number of activity completions needed before the node is realized the second and succeeding times. The number on the right side of the node (7 in Figure 11) is simply the node identification number.

The activities in the GERT network must also be described. This is done by specifying three descriptors. They are, in order, (1) the probability that a given branch will be taken, (2) a number referencing a set of parameters for the time distribution of that activity, and (3) a code number specifying the time distribution. This is illustrated in Figure 12.

In Figure 12, there is a probability of 0.3 that the branch will be taken, the parameters associated with the time probability distribution are stored as set number 5, and the time distribution is type 2 (Normal distribution). In parameter set number 5 would be found the mean and standard deviation of the particular normal distribution associated with the branch. (GERT allows the selection of any one of eleven time distributions for an activity).

FIGURE 12

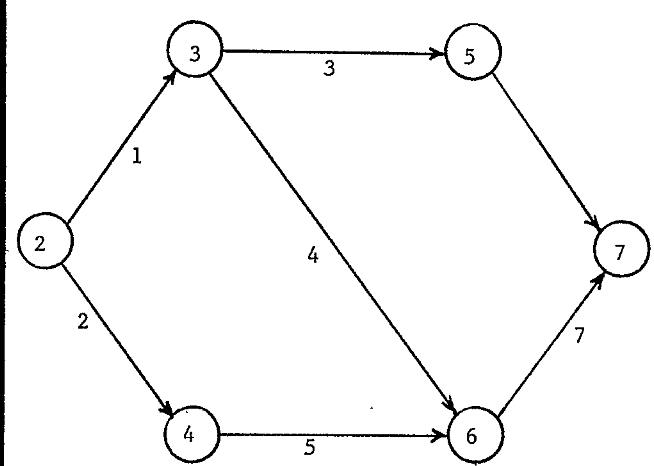


A PERT/GERT MODEL

As discussed previously, one of the major problems with the usual PERT network approach is that of merge event bias. The use of GERT in simulation of a network allows the addressing of the merge event bias problem, together with the gathering of much-needed network statistics. This can perhaps be best illustrated by using GERT to simulate a PERT network.

Moore and Clayton (12) use the small PERT network shown in Figure 13 to illustrate the use of GERT.

FIGURE 13



The usual PERT time estimates, again from Moore and Clayton (12) are

Activity	a	m	b
1	1.0	4.0	7.0
2	1.0	5.0	8.0
3	2.0	8.0	14.0
4	2.0	7.0	10.0
5	1.0	7.0	11.0
6	1.0	6.0	9.0
7	2.0	5.0	7.0

By using the procedure discussed previously for calculating PERT mean times and standard deviations, the following mean times for the three paths through the network were obtained. (12)

Path	Mean Times
1-3-6	17.67
1-4-7	15.50
2-5-7	16.33

The path 1-3-6 is then the critical path because it has the longest expected time to completion.

The same problem formulated as a GERT network becomes as shown in Figure 14. (12)

All nodes are coded as deterministic. The number in the upper left (number of completions for first realization) is equal to the number of activities leading to the node and, since the node can be realized only once per pass through the network (no looping), the number of completion for subsequent realizations is set to infinity. The numbers in squares on the activities simply identify the activities for GERT.

Once the problem has been set up as a GERT network, the network can be simulated the number of times desired using the GERTS-IIIZ simulator (available from Pritsker and Associates). GERT can then list a criticality index for each activity in the network by noting the relative frequency with which that activity is on the longest path. Further, the relative frequency of a path being the critical path can be found by noting the activity on that path with the lowest criticality index.

The GERTS-IIIZ simulator has the capability of gathering summary statistics. If the minimum, maximum, mean and standard deviation of time to completion of the project (sink node) are kept, together with a frequency distribution of these times, it is easy to see that such information would be significant to project management.

In the example just shown, for instance, Moore and Clayton found that standard PERT techniques were overly optimistic on mean time to project completion by a factor of 0.94 days or just over 5% (12). The GERTS-IIIZ approach also yielded a more accurate estimate of the time to completion(smaller range) than PERT.

Thus, through calculation of the criticality index for each activity, the capability of other automatic data gathering features for a network, and the printing out of a frequency distribution of project completion times, (all available through GERTS-IIIZ), it can be seen that GERT is a powerful tool for analysis of PERT-type networks, and essentially eliminates the merge event bias problem.

BIBLIOGRAPHY

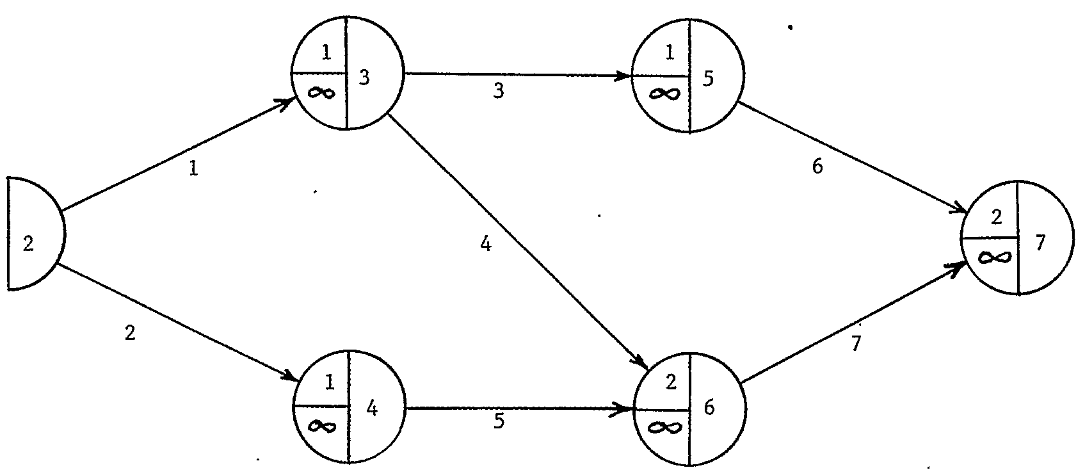
1. Charnes, A., W. W. Cooper, and G. L. Thompson, "Critical Path Analyses via Chance Constrained and Stochastic Programming." Operations Research, Vol. 12, No. 3, May-June (1964), pp. 460-70.

2. Clark, C. E., "The Greatest of a Finite Set of Random Variables." Operations Research, Vol. 9, No. 2, March-April (1961) pp. 145-162.

3. Clingen, C. T., "A Modification of Fulkerson's Algorithm for Expected Duration of a PERT Project when Activities Have Continuous D.F.." Operations Research, Vol. 12, No. 4, (1964) pp. 629-632.

4. Elmaghraby, S. E., "On the Expected Duration of PERT Type Networks." Management Science, Vol. 13, No. 5, January (1967) pp. 299-306.

FIGURE 14



5. Freund, J. E., and Williams, F. J. Modern Business Statistics, Pitman, 1959.

6. Fulkerson, D. R., "Expected Critical Path Lengths in PERT Networks." Operations Research, Vol. 10, No. 6, November-December (1962), pp. 808-817.

7. Klingel, A. R., "Bias in PERT Project Completion Time Calculation for a Real Network." Management Science, Vol. 13, No. 4, December 1966, pp. B-194-201.

8. MacCrimmon, K. R., and C. A. Ryavec, "An Analytical Study of the PERT Assumptions." Operations Research, Vol. 12, No. 1, January-February 1964, pp. 16-37.

9. McMillan and Gonzalez, Systems Analysis, A Computer Approach, Irwin Publishing Co., 1965.

10. Miller, I., J. E. Freund, Probability and Statistics for Engineers, Prentice-Hall, Inc., 1965.

11. Moder, J. J., and C. R. Phillips, Project Management with CPM and PERT, Reinhold Corp., 2nd ed., 1970.

12. Moore, L. J., and E. R. Clayton, GERT Modeling and Simulation: Fundamentals and Applications, Petrocelli/Charter, New York, 1976.

13. Pritsker, A. A. B., "GERT: Graphical Evaluation and Review Technique." The Rand Corp., RM-4973-NASA, Santa Monica, Calif., April 1966.

14. Pritsker, A. A. B., The GASP IV Simulation Language, John Wiley & Sons, New York, 1974.

15. Schriber, Thomas J., Simulation Using GPSS, John Wiley & Sons, New York, 1974.

16. Signal, C. E., and A. A. B. Pritsker, "SMOOTH: A Combined Continuous-Discrete Network Simulation Language." SIMULATION, Vol. 21, March 1974.