

CRASHING PERT NETWORKS: A SIMULATION APPROACH

Wayne A. Haga

Assistant Professor of Computer Information Systems
Metropolitan State College of Denver
P.O. Box 173362, Campus Box 45
Denver, CO 80217-3362
(303)-556-4698
haga@mscd.edu

Tim O'keefe

Associate Professor of Computer Information Systems
University of North Dakota
Gamble Hall RM 365
Grand Forks, North Dakota 58201
(701)-777-2517
timothy_okeefe@und.nodak.edu

Paper presented at the 4th International conference of the
Academy of Business and Administrative Sciences Conference
Quebec City, Canada
July 12-14, 2001

Abstract

The Project Evaluation and Review Technique (PERT) has been used as a tool for project management for over four decades. To reduce a projects completion time, a technique called “crashing” is performed, which involves bringing in additional resources for activities along the critical path of the network. The traditional method of crashing PERT networks ignores the stochastic nature of activity times, reducing the stochastic model to a deterministic model and simply using activity time means in calculations. The project is arbitrarily crashed to some desired completion date, without consideration for what the penalty for late project completion is. Additionally, the method ignores the fact that reducing some activity times may reduce the mean project completion time more than others, due to bottlenecks. The authors created a computer simulation model to determine the order in which activities should be crashed as well as the optimal crashing strategy for a PERT network to minimize the expected value of the total (crash +overrun) cost, given a specified penalty function for late completion of the project. This work was initially explored by Haga (1998) as part of his unpublished Doctoral dissertation.

Introduction

Project management consists of planning, designing, and implementing a set of activities to accomplish a particular goal or task. For many years, two of the most popular approaches to project management have been the Critical Path Method (CPM) and the Project Evaluation and Review Technique (PERT). J.E. Kelly of Remington-Rand and M.B. Walker of Dupont developed CPM in the 1950's to assist in scheduling maintenance shutdowns of chemical processing plants. Pert was developed shortly after by the U.S. Navy to manage the development of the Polaris missile. The original PERT Navy report (1958) does not identify the names of the developers.

Both PERT and CPM are network based techniques. The critical path method can be summarized as follows:

1. Determine or estimate precedence relationships and duration times for each activity required to complete the project.
2. Construct a network model from the relationships and duration times. Arrows are used to represent activities and nodes used to represent the starting and ending point of an activity. Two different activities are not allowed to have the same starting and ending nodes, so occasionally dummy activities are required in construction of the network.
3. For each activity i , calculate the earliest possible start time (ES_i) and earliest completion time (EC_i), by making a "forward pass" through the network. The earliest starting time of all activities which have no predecessors are set to zero. The earliest completion time of each activity is equal to the earliest completion time plus the activity duration time (T_i). The earliest start time for activities with predecessors is equal to the maximum of the earliest completion times of all immediate predecessors to the activity. The completion time for the project is equal to the maximum of the latest completion time of all activities.
4. The latest start time (LS_i) and latest completion time (LC_i) of all activities are calculated in a "backward pass" through the network. The latest completion time of activities with no successors are set equal to the completion time of the project. The latest start time of an activity is calculated by subtracting its activity time from its latest completion time. The latest completion time for an activity is equal to the minimum of the latest start time of all activities which succeed it.
5. Calculate the slack (S_i) for each activity by subtracting the earliest start time (ES_i) from the latest start time (LS_i). The slack for each activity represents the amount of time an activity may be delayed from its earliest start time without delaying the project.
6. Identify the critical path, which is made up of activities for which the slack is equal to zero.

It is often necessary or desirable to shorten the duration of a project. Crashing refers to reducing the duration of an activity by devoting additional resources to the activity, typically at a higher cost. For each activity that can be crashed, a time/cost tradeoff function is specified. The activity on the critical path with the minimum cost per time period (known as the time/cost slope) to crash is selected, the activity time is reduced, and the network is reanalyzed. The process is continued until the project will be completed within the desired time frame, the cost of further crashing outweighs the benefit, or all activities have been crashed to their shortest possible duration.

PERT can be considered an extension of CPM. Instead of using a fixed time estimate for each activity, activity times are assumed to follow the generalized Beta distribution. Thus, PERT is a probabilistic network model whereas CPM is a deterministic network model.

In PERT, three time estimates are required for each activity. The time estimates represent a pessimistic time (a), an optimistic time (b), and a most likely time (m) for duration of the activity. The pessimistic time is assumed to equal the 0.00 fractile of the distribution, the optimistic time equal to the 1.00 fractile, and the most likely time equal to the mode. The mean completion time of each activity is then estimated as

$$t_e = (a + 4m + b) / 6$$

and the standard deviation as

$$s = (b - a) / 6.$$

Using the estimated mean of each activity time, the network is analyzed in the same manner as the CPM method. The PERT method assumes that the sum of the mean completion times of activities on the critical path is normally distributed. This allows the calculation of the probability of completing the project within a given time period.

The PERT method uses only the activity time means in calculating the critical path, reducing the stochastic model to a deterministic model. A single critical path is thus calculated and used, whereas in reality there may be numerous possible critical paths realized. For a large network the probability that any given path may be the critical one may be very small. As a result, the traditional PERT method of calculating the time to complete the project is almost always too low. The extent to which PERT underestimates completion times is examined for a variety of networks by Klingel (1966).

The PERT method of crashing a network is identical to the CPM method, using the mean completion time of each activity and ignoring its distribution.

A more detailed explanation of the PERT and CPM methods can be found in any textbook on Project Management, Operations Research, or Production and Inventory Control. There are numerous software packages available to perform PERT, and/or CPM analysis.

Research Questions

The crashing algorithm for the CPM model is not suitable for the PERT model for the following reasons:

1. If there is a single critical path, there is still only a 50% probability of the project being completed by the target date, since mean activity times are used to calculate the completion time of the project. If there are numerous possible critical paths the probability may be much less than 50%. This may be costly if there are penalties for late completion of the project. The complete distribution of project completion time needs to be considered when crashing.
2. Since there may be numerous possible critical paths, crashing a given activity by one time period will not necessarily reduce the completion time of the project by one time period. The expected reduction in project completion time must be considered in addition to the time/cost slope when selecting an activity to be crashed.

An improved method for crashing PERT networks could save a company a significant amount of money in crashing and overrun costs. Even if there are not direct costs in the form of penalties for late completion of projects, there is likely to be costs due to loss of business because of a damaged reputation.

Literature Review

Since the development of CPM and PERT in the 1950's, the techniques have been the subject of hundreds of research papers. Research has generally been focused on PERT, since the deterministic CPM presents few problems of interest. The research on PERT can be organized into five general categories. The first category includes research on the error and bias due to PERT assumptions made. The second category involves finding the distribution function $F(T)$ of project completion times through exact analysis, approximation, and bounding methods. Third, Monte Carlo sampling has been used to study the distribution of PERT networks. The fourth category involves resource allocation problems and load leveling. The fifth area of research is the crashing of PERT networks. This literature review will focus on the research most directly related to this study; Monte Carlo simulations and crashing of PERT networks.

Van Slyke (1963) was the first of many researchers to apply Monte Carlo simulations to study PERT. Van Slyke demonstrated several advantages of using simulation including more accurate estimates of the true project length, flexibility in selecting any distribution for activity times, and the ability to calculate "criticality indexes" which are the probability of various activities being on the critical path.

Pritsker and Happ (1966) developed a modification of PERT called the Graphical Evaluation and Review Technique (GERT). GERT allowed activity times to follow several different distributions. Project completion time distributions were computed through Monte Carlo simulation. Kennedy and Thrall (1976) developed a modification of GERT called Project

Length Analysis and Evaluation Technique (PLANET). PLANET added the ability to calculate the probability of activities being critical and find the distribution of completion times for each activity.

Ameen (1987) developed Computer Assisted PERT Simulation (CAPERTSIM), a simulation program developed as a teaching tool to teach project management techniques. Students used the program to evaluate decision making under uncertainty and cost-time relationships and trade-offs. Ameen reported students reacted very favorably to participating in the computer-assisted PERT simulation project.

Badiru (1991) reported development of another simulation program for project management, called STARC. STARC allows the user to calculate the probability of completing the project by a specified deadline. It also allows the user to enter a "duration risk coverage factor". This is a percentage over which the time range of activities are extended. This allows there to be some probability of generating activity times above the pessimistic time and below the optimistic time.

Additional authors which have studied various PERT problems via simulation include Klingel (1966), Gray (1969), Burt (1971), Herbert (1979), Schonberger (1981), and Dodin (1984), and Kidd (1986).

The traditional method of crashing PERT networks has been to convert the model to a deterministic model by using the means of the activity times. The network is then crashed in a series of iterative steps until the expected completion time of the project is acceptable, the cost of crashing exceeds the benefits, or all activities have been crashed as much as possible.

Benson and Sewall (1972) report on FASNET, a program which uses the CPM model and allows data to be entered as the project progresses. The network is reanalyzed each time new data is entered. Moore et al. (1978) and Hannan (1978) reformulate the problem using goal programming. Goal programming is a modification of linear programming which can solve problems with multiple objectives. This allows goals in addition to cost minimization to be added to the problem. Because of conflicting objectives, not all goals are achieved completely. Foldes and Soumis (1993) present a reformulation of crashing networks when the cost-time tradeoff is represented by a non-linear, non-differentiable convex function.

Very little research has been conducted which addresses crashing in a stochastic environment. Herbert (1979) suggests that the reason for the lack of such research stems from the fact that it "seems unlikely that the effort required to gather the data and or estimates need to specify stochastic tradeoff functions for each activity in a large project could be justified on the basis of anticipated improvement in performance." Antill and Woodhead (1982) suggest a different reason:

It is obvious that PERT compression yields indefinite results, indicating only the areas within the project that can be considered for probable compressions. In effect, the overall compression characteristics of the network model are too vague to provide other than an approximate forecast of the time-cost behavior. Compression therefore cannot be applied with confidence to PERT networks as it can to a CPM model.

The first approach to address the problem of crashing under stochastic conditions was made by Coskun (1984). Coskun formulated the problem as a chance constrained linear programming (CCLP) problem. CCLP is a method of attempting to convert a probabilistic mathematical programming formulation into an equivalent deterministic formulation. Coskun's formulation ignored the assumed beta distribution of activity times. Instead activity times were assumed to be normally distributed with the mean and standard deviation of each known. The formulation allows a desired probability of completion within a target date to be entered. Coskun concluded that "While the solution of the CLLP formulations of the optimal PERT compression problem provides a wealth of information with significant managerial implications, the computational efforts necessary to solve the CCLP are no greater than those necessary to solve the deterministic compression problem."

Johnson and Schou (1990) used simulation to compare three rules for crashing stochastic networks:

1. Select the lowest cost slope activity or activities that will shorten the critical path(s). Rule 1 is the CPM developed rule for crashing a project. It reflects the common approach taken to expediting in the literature.
2. Select the activity with the highest criticality index. In the case of ties, choose the alternative that costs least per unit of time to reduce. The criticality index should ideally be recomputed at each step in crashing a project.
3. Select the least cost/day activity first. This rule is a combination of the first two rules. It reflects the idea of selecting the least cost expected value. The procedure follows Rule 2 in calculation a criticality index. The criticality index would then be multiplied by the number of days the expected time of an activity can be reduced. This yields an expected number of days that the critical path can be shortened. This expected value is then divided into the total incremental cost of expediting the activity. Theoretically, the criticality index should be regenerated and the computations repeated at each step in crashing the project.

They concluded Rule 3 provided the lowest cost of crashing the network. Although the differences in cost were small in the examples, the authors argued that "the greater size of 'real life' problems and the likelihood of multiple critical paths would likely lead to larger differentials in the expected cost of different rules."

Ramini (1986) also proposed an algorithm for crashing PERT networks incorporating the use of criticality indices. Apparently he did not implement the algorithm, as no results are reported. The authors also believe his method does not take into account bottlenecks.

Methodology

A new proposed method of crashing PERT networks was compared with the traditional PERT method. The new method used a linear penalty function for late completion of a project. Each activity in the network that had not been crashed to its limit was tested to determine which would reduce the total crash + overrun cost the most. A simulation program was written in the C++ programming language to study the benefits of using simulation and a cost/penalty function to determine an optimal crash strategy. The bases for comparisons used were (a) mean completion time of the project and (b) total (crash + overrun) mean cost.

Consider a project with n activities, each of which follows a generalized beta distribution. Precedence and time estimates (a , b , and m) are known for each activity. A linear time/cost trade-off function is specified for each activity. A linear penalty function for late completion of the project is specified.

Each simulation trial will consist of two steps:

1. Randomly generate n activity duration times from the appropriate distributions.
2. Compute and record the project duration time, crash costs, and overrun costs.

Through preliminary runs it was determined that 10,000 runs were required to obtain stable results.

The two methods of crashing compared were as follows:

Method 1: The traditional PERT method is used to crash the network to its target completion date.

Method 2: Each activity in the network that had not been crashed to the maximum was tested to determine which would reduce the mean expected total (crash + overrun) cost the most. Crashing was continued as long as the total cost continued to decline.

Test Data

Three fictitious networks were created so as to represent three extreme cases:

Data Set #1: A network with a very dominant critical path (indicated with bold lines). The PERT diagram is shown in Figure 1. Crash costs were randomly generated ranging from \$1,000 to \$9,000 per week. The penalty function was specified at \$10,000 per week.

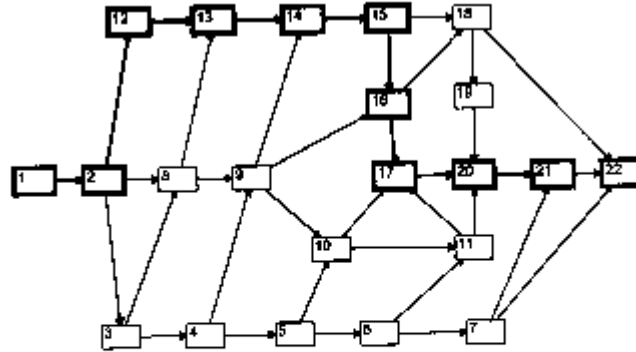


Figure 1. PERT Network for Data Set #1

Data Set #2: A network with all paths of equal length. The PERT diagram for date set #2 is shown in Figure 2. Crash costs were randomly generated ranging from \$5,000 to \$15,000 per day. The penalty function was specified at \$50,000 per week.

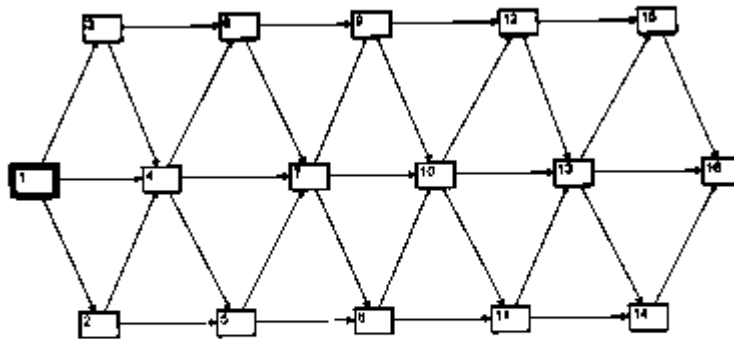


Figure 2. PERT Network for Data Set #2

Data Set #3: A network with two major bottlenecks. The PERT diagram is shown in Figure 3. Since the purpose of this third set of test data was to demonstrate the effect of bottlenecks, crash costs were fixed at \$10,000 per week. The penalty function was specified at \$40,000 per week.

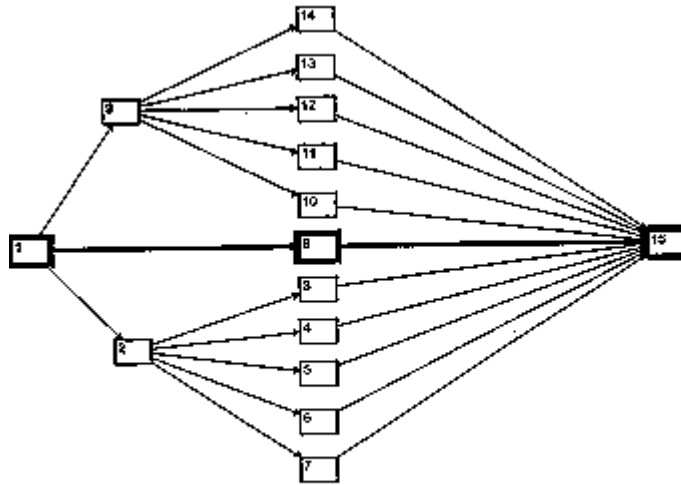


Figure 3. PERT network For Data Set #3

The maximum number of weeks an activity could be crashed was arbitrarily set to three weeks for every activity in all three networks.

Results

The following results were first reported by Haga (1998) as part of his unpublished Doctoral dissertation.

The results for Data Set #1 are summarized in Table 1. The results of the simulation show an expected mean completion time of 179.9 weeks, almost identical to the expected completion date of 180 days using the traditional PERT calculation method. This was expected since the network was designed with a single very dominant critical path and a target completion date equal to this mean. When crashed to the optimal expected crash + overrun cost, the mean completion time is reduced to 176.9 weeks, a 1.7% reduction. This resulted in a 31.1% reduction in total costs. Figure 4 illustrates the distribution of completion times for the two methods. As one would expect, the entire distribution shifted to the left under Method 2.

Table 1. Summary of Results for Data Set #1

	Method 1	Method 2
Mean Completion Time	179.90	176.90
Percent Change		-1.7%
Standard Deviation	7.69	7.72
Percent Change		0.4%
Mean Total Cost	30,312.00	20,883.00
Percent Change		-31.1%

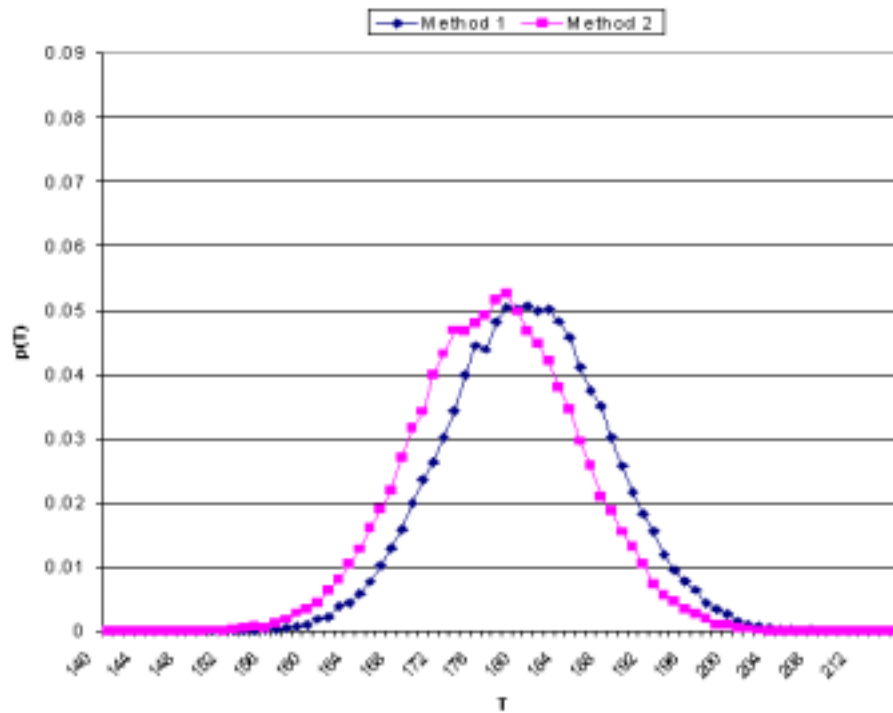


Figure 4. Distribution of Completion Times for Data Set #1

The results for Data Set #2 are summarized in Table 2. The simulated mean completion time for Data Set #2 was 106.7 weeks, 11.7 weeks greater than the expected completion time of 95 weeks under the traditional PERT method calculation. This was to be expected given that every path was of equal length, and each had an equal probability of becoming critical. Method 2 resulted in the network being crashed to a mean completion time of 100.1 weeks, a 6.1% reduction. This resulted in a 6.6% reduction in mean total costs. The smaller reduction in savings can be explained by the fact that crashing a single activity will simply increase the probability of another of the equally long paths becoming critical. Thus, reducing the completion time by a full week required crashing numerous activities. Figure 5 illustrates the distribution of completion times for the two methods. Once again, the entire distribution simply shifted to the left.

Table 2. Summary of Results for Data Set #2

	Method 1	Method 2
Mean Completion Time	106.70	100.10
Percent Change		-6.1%
Standard Deviation	4.57	4.55
Percent Change		-.4%
Mean Total Cost	877,874.00	819,664.00
Percent Change		-6.6%

The results for Data Set #3 are summarized in Table 3, and the distribution of completion times shown in Figure 6. The mean completion time was reduced 6.9% from 55.4 weeks to 51.5 weeks. This resulted in the total cost being reduced by 10.4%. As the authors expected, the activities creating the two bottlenecks were crashed to the limit immediately.

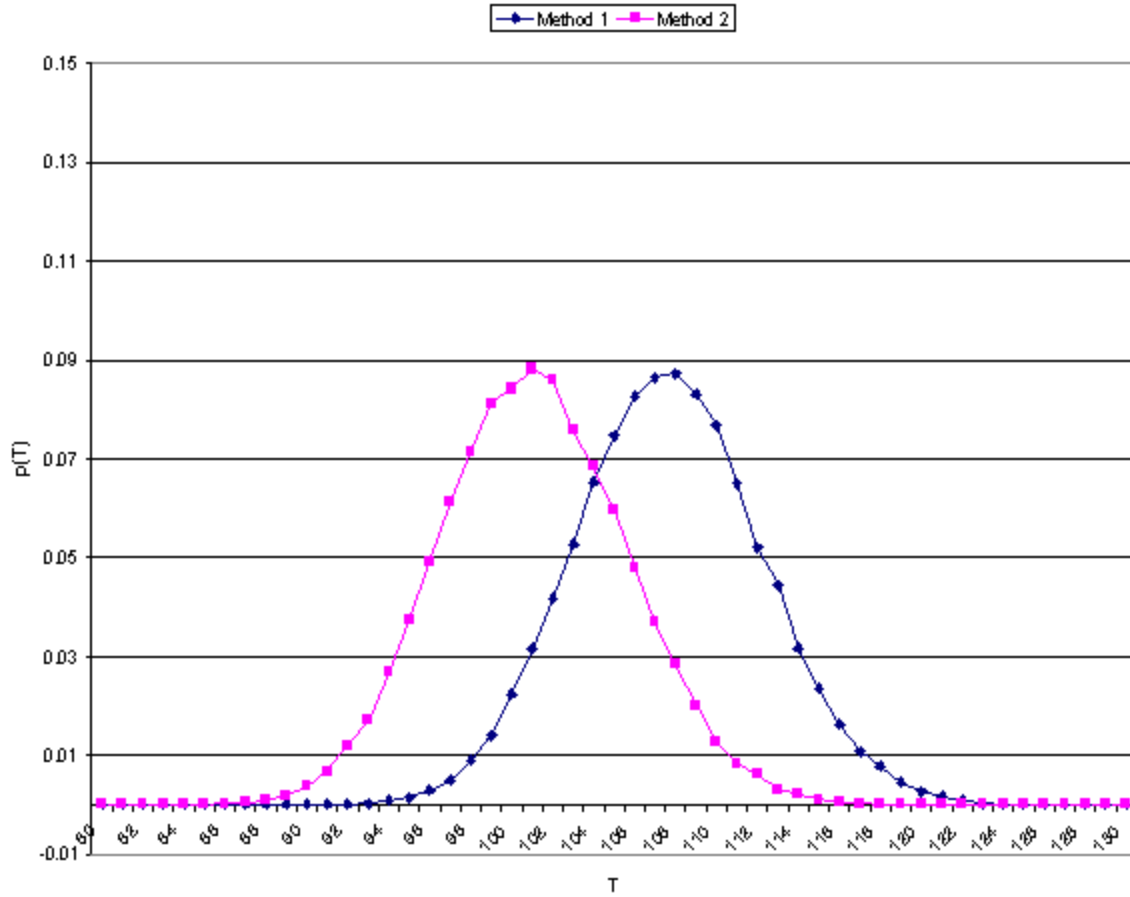


Figure 5. Distribution of Completion Times for Data Set 2

Table 3. Summary of Results for Data Set #3

	Method 1	Method 2
Mean Completion Time	55.40	51.50
Percent Change		-6.9%
Standard Deviation	2.80	2.75
Percent Change		-1.8%
Mean Total Cost	234,571.00	210,197.00
Percent Change		-10.4%

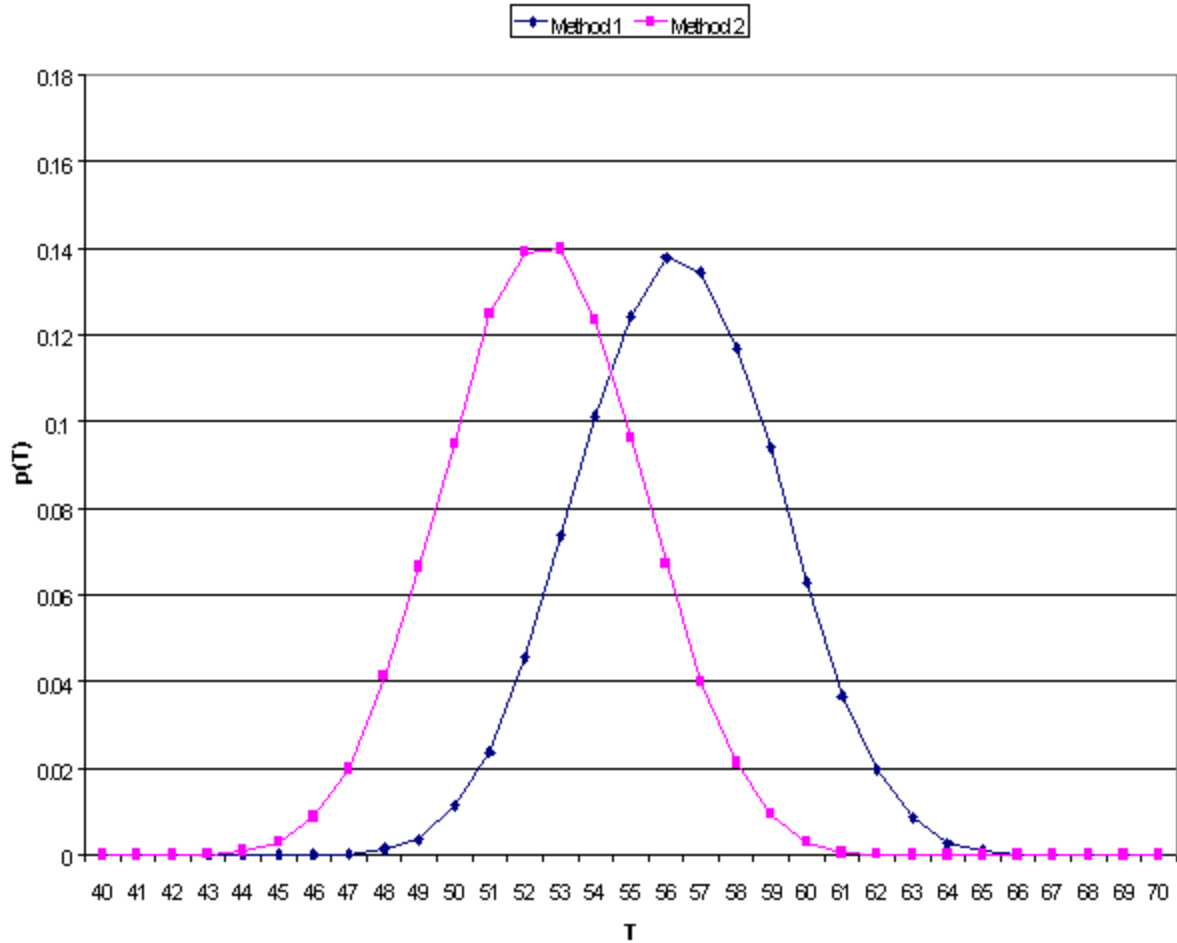


Figure 6. Distribution of Completion Times for Data Set #3

Standard deviations of project completion times were also calculated, but were not significantly different between the two methods. This was expected since it was assumed that crashing an activity reduced the mean completion time of the activity, but did not change the distribution of completion times.

Summary, Conclusions, and Future Research

The purpose of this study was to explore the suitability of a new method to crash PERT networks. A simulation program was written in the C++ programming language and run on three hypothetical data sets.

For all data sets, the new method significantly reduced the mean total (crash + overrun) cost for the project. The savings varied from 10.4% to 31.1% over the traditional PERT method. The results were encouraging, and suggest further study of this technique is warranted.

Plans for future research include running the simulation at different levels of a linear penalty function. Other types of penalty functions such as an exponential function need to be explored as well. Testing on additional networks, preferably including real-world projects, is required. Algorithms to reduce the number of activities tested as possible choices for crashing need to be explored to reduce the simulation time required for a large PERT network.

References

- Ameen, D. A. (1987, April). A computer assisted PERT simulation. Journal of Systems Management, 6-9.
- Antill, J. M. & Woodhead, R. W. (1982). Critical path methods in construction practice, 3rd ed., New York: Wiley.
- Badiru, A. B. (1991). A simulation approach to network analysis. Simulation, 57,(4), 245-255.
- Benson, L. A. & Sewall, R. F. (1972). Dynamic crashing keeps projects moving. Computer Decisions, 4(2), 14-18.
- Burt, J. M. & Garman, M. B. (1971). Conditional Monte Carlo: A simulation technique for stochastic network analysis. Management Science, 18(3), 207-217.
- Coskun, O. (1984). Optimal probabilistic compression of PERT networks. Journal of Construction Engineering and Management. 110(4), 437-446.
- Dodin, B. M. (1984). Determining the K most critical paths in PERT networks. Operations Research, 32(4), 859-877.
- Foldes, S. & Sourmis F. (1993). PERT and crashing revisited: Mathematical generalizations. European Journal of Operational Research, 64, 286-294.
- Gray, C. F. & Reiman, R. E., PERT simulation: A dynamic approach to the PERT technique. Journal of Systems Management, 20, 18-23.
- Haga, W. A. (1998). Crashing PERT Networks. Unpublished Doctoral dissertation, Applied Statistics Department, University of Northern Colorado, Greeley, Colorado.
- Herbert, J. E. (1979). Applications of simulation in project management. IEEE 1979 Winter Simulation Conference., 211-219.
- Johnson, G. A. & Schou, C. D. (1990). Expediting projects in PERT with stochastic time estimates. Project Management Journal, 21(2), 29-32.
- Kennedy, K. W. & Thrall, R. M. (1976). PLANET: A Simulation approach to PERT. Computers and Operations Research, 3, 313-325.

Kidd, J. B. (1987). A comparison between the VERT program and other methods of project duration estimation. Omega, International Journal of Management Science, 15(2), 129-134.

Klingel, A. R. (1966). Bias in PERT completion times calculations for a real network. Management Science, 13(4), 476-489.

Moore, L. J. et al., (1978). Analysis of a multi-criteria project crashing model. AIIIE Transactions, 10(2), 163-169.

Navy Special Project Office, Bureau of Naval Weapons (1958). PERT Summary Report, Phase I. National Technical Information Service, AD-735902.

Pritsker, A. B. & Happ, W. W. (1966). GERT:Graphical evaluation and review technique, Part I, Fundamentals. Journal of Industrial Engineering, 17(5), 267-274.

Ramini S. (1986). A simulation approach to time-cost trade-off in project network. Modeling and Simulation on Microcomputers:1986 Proceedings of the Conference, 115-120.

Schonberger, R. J. (1981). Why projects are always late: A rationale based on manual simulation of a PERT/CPM network. Interfaces, 11(5), 66-70.

Van Slyke, R. M. (1963). Monte Carlo methods and the PERT problem. Operations Research, 13, 141-143.