

Simulator for Optimization of Software Project Cost and Schedule

¹P.K. Suri and ²Bharat Bhushan

¹Department of Computer Science and Applications,
Kurukshetra University, Kurukshetra (Haryana), India

²Department of Computer Science and Applications,
Guru Nanak Khalsa College, Yamuna Nagar (Haryana), India

Abstract: Each phase of the software design consumes some resources and hence has cost associated with it. In most of the cases cost will vary to some extent with the amount of time consumed by the design of each phase. The total cost of project, which is aggregate of the activities costs will also depends upon the project duration, can be cut down to some extent. The aim is always to strike a balance between the cost and time and to obtain an optimum software project schedule. An optimum minimum cost project schedule implies lowest possible cost and the associated time for the software project management. In this research an attempt has been made to solve the cost and schedule problem of software project using PERT network showing the details of the activities to be carried out for a software project development/management with the help of crashing, reducing software project duration at a minimum cost by locating a minimal cut in the duration of an activity of the original project design network. This minimal cut is then utilized to identify the project phases which should experience a duration modification in order to achieve the total software duration reduction. Crashing PERT networks can save a significant amount of money in crashing and overrun costs of a company. Even if there are no direct costs in the form of penalties for late completion of projects, there is likely to be intangible costs because of reputation damage.

Key words: Software, PERT, crashing, beta distribution, scheduling, activity cost, cost slope

INTRODUCTION

In this recent era of corporate management confusion and reorganization, the time and cost issues of traditional project management have become even more critical. The traditional project management models with ever-present budget overruns and late completion dates have reduced the concept's credibility and injured the productivity and reputation of almost every organization-from small business to large corporations.

The success of any project is very much dependent upon the quality of the planning, scheduling and controlling of the various phases of the project. Program Evaluation and Review Technique (PERT) is an aid to management in expediting and controlling the resources to meet the scheduled completion date of the projects which involve high degree of uncertainty.

Activity durations are estimates of the actual time required and there is liable to be a significant amount of uncertainty associated with the actual durations. To incorporate uncertainty in the scheduling process is to

apply the critical path scheduling process and then analyze the results from a probabilistic perspective. This process is referred to as the PERT (Programmed evaluation and Review Technique) scheduling or evaluation method. Using expected activity durations and critical path scheduling, a critical path of activities can be identified^[1]. The critical path is then used to analyze the duration of the project in incorporating the uncertainty of activities durations along the critical path. The expected project duration is equal to the sum of the expected durations of the activities along the critic.

Assuming that the activity durations follow a probabilistic beta distribution under a restrictive condition, the probability density function of a beta distribution for a random variable x is given by:

$$f(x) = k(x-a)^{\alpha}(b-x)^{\beta}$$
$$a \leq x \leq b \quad \alpha, \beta > -1$$

where, k is a constant which can be expressed in terms of α and β .

Corresponding Author: P.K. Surim, Department of Computer Science and Applications, Kurukshetra University, Kurukshetra (Haryana), India

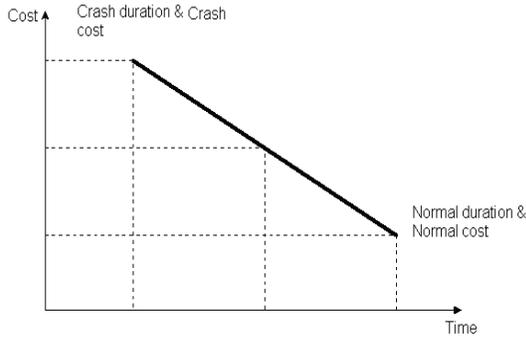


Fig. 1: Linear time and cost trade-off for an activity

For a beta distribution in the interval $a \leq x \leq b$ having a modal value m , the mean is given by:

$$\mu = \frac{a + (\alpha + \beta)m + b}{\alpha + \beta + 2}$$

with additional condition on the beta distribution is:

$$\alpha + \beta = 4(\alpha = \beta = 2) \text{ and } \sigma = (b - a)/6$$

If a , m and b are the optimistic, most likely and pessimistic activity durations denoted by $a_{i,j}$, $m_{i,j}$ and $b_{i,j}$ respectively for each activity (i, j) duration then.

The mean and variance for each activity (i, j) duration are computed as:

$$\mu(i, j) = \frac{1}{6}(a_{ij} + 4m_{ij} + b_{ij})$$

And:

$$\sigma^2(i, j) = \frac{1}{36}(b_{ij} - a_{ij})^2$$

The use of these optimistic, most likely and pessimistic estimates stems from the fact that these are thought to be easier for managers to estimate subjectively^[2].

Activity time-cost relationship: A simple representation of the possible relationship between the duration of an activity and its direct costs appears in Fig. 1. Shortening the duration on an activity will normally increase its direct cost.

A duration which implies minimum direct cost is called the normal duration and the minimum possible time to complete an activity is called crash duration, but at a maximum cost. The linear relationship shown above between these two points implies that any intermediate duration could also be chosen.

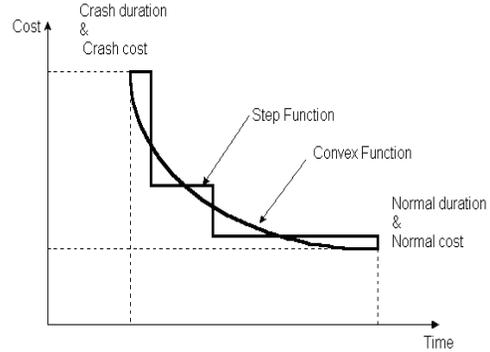


Fig. 2: Non-linear time and cost trade-off for an activity

It is possible that some intermediate point may represent the ideal or optimal trade-off between time and cost for this activity. The slope of the line connecting the normal point (lower point) and the crash point (upper point) is called the cost slope of the activity. The slope of this line can be calculated mathematically by knowing the coordinates of the normal and crash points:

$$\text{Cost slope} = (\text{crash cost} - \text{normal cost}) / (\text{normal duration} - \text{crash duration})$$

As the activity duration is reduced, there is an increase in direct cost. A simple case arises in the use of overtime work and premium wages to be paid for such overtime. Also overtime work is more prone to accidents and quality problems that must be corrected, so indirect costs may also increase. So, do not expect a linear relationship between duration and direct cost but convex function as shown in Fig. 2.

Project time-cost relationship: Total project costs include both direct costs and indirect costs of performing the activities of the project. If each activity of the project is scheduled for the duration that results in the minimum direct cost (normal duration) then the time to complete the entire project might be too long and substantial penalties associated with the late project completion might be incurred. At the other extreme, a manager might choose to complete the activity in the minimum possible time, called crash duration, but at a maximum cost. Thus, planners perform what is called time-cost trade-off analysis to shorten the project duration. This can be done by selecting some activities on the critical path to shorten their duration. As the direct cost for the project equals the sum of the direct costs of its activities, then the project direct cost will increase by decreasing its duration. On the other hand, the indirect cost will decrease by decreasing the project duration, as the indirect cost are almost a linear function with the project duration.

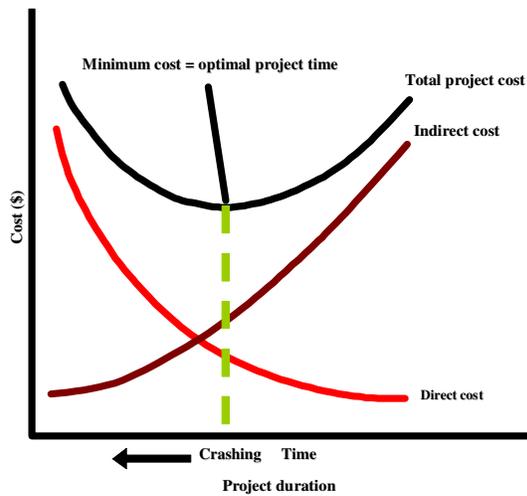


Fig. 3: Project time-cost relationship

Figure 3 shows the direct and indirect cost relationships with the project duration. The project total time-cost relationship can be determined by adding up the direct cost and indirect cost values together. The optimum project duration can be determined as the project duration that results in the least project total cost.

Steve and Dessouky^[3] described a procedure for solving the project time/cost tradeoff problem of reducing project duration at a minimum cost. The solution to the time & cost problem is achieved by locating a minimal cut in a flow network derived from the original project network. This minimal cut is then utilized to identify the project activities which should experience a duration modification in order to achieve the total project reduction.

Rehab and Carr^[4] described the typical approach that construction planners take in performing time-Cost Trade-off (TCT). Planning focuses first on the dominant characteristics and is then fine-tuned in its details. Planners typically cycle between plan generation and cost estimating at ever finer levels of detail until they settle on a plan that has an acceptable cost and duration. Computerized TCT methods do not follow this cycle. Instead, they separate the plan into activities, each of which is assumed to have a single time-cost curve in which all points are compatible and independent of all points in other activities' curves and that contains all direct cost differences among its methods.

Pulat and Horn^[5] described a project network with a set of tasks to be completed according to some precedence relationship, the objective is to determine efficient project schedules for a range of project

realization times and resource cost per time unit for each resource. The time-cost tradeoff technique is extended to solve the time-resource tradeoff problem. The methodology assumes that the project manager's (the decision maker) utility function over the resource consumption costs is linear with unknown weights for each resource. Enumerative and interactive algorithms utilizing Geoffrion's $P(\lambda)$ approach are presented as solution techniques. It is demonstrated that both versions have desirable computational times.

Walter *et al.*^[6] described the application of advanced methods of process management, especially in those fields in which activity durations can be determined only vaguely, while at the same time a highly competitive market enforces strict completion schedules through the implementation of penalties. The technique presented is a new PERT-based, hybridized approach using simulated annealing and importance sampling to support typical process re-engineering, which focuses on the efficient allocation of extra resources in order to achieve a more reliable performance without changing the precedence-successor-structure. The technique is most suitable for determining a time-cost trade-off based on practice-relevant assumptions.

Marold^[7] used 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. Three extreme network types are examined, each with two different penalty functions.

Van Slyke^[8] demonstrated several advantages of applying simulation techniques to PERT, 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. Van Slyke was the first to apply Monte Carlo simulations to PERT.

Ameen^[9] developed Computer Assisted PERT Simulation (CAPERTSIM), an instructional tool to teach project management techniques. Students used the program to evaluate decision-making under uncertainty and cost-time relationships.

Coskun^[10] 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. This formulation allows a desired probability of completion within a target date to be entered.

Ramini^[11] proposed an algorithm for crashing PERT networks with the use of criticality indices. Apparently he did not implement the algorithm, as no results were ever reported. His method does not allow for bottlenecks. Bottlenecks traditionally have multiple feeds into a very narrow path that is critical to the project's completion. Bottlenecks are the favored locations for project managers to build time buffers into their estimates, yet late projects still abound because of deviation from timetables and budgets.

Johnson and Schon^[12] used simulation to compare three rules for crashing stochastic networks. He also made use of criticality indices.

Badiru^[13] 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 ranges of activities are extended. This allows some probability of generating activity times above the pessimistic time and below the optimistic time. Feng *et al.*^[14] presented a hybrid approach that combines simulation techniques with a genetic algorithm to solve the time-cost trade-off problem under uncertainty.

Grygo^[15] pointed out that the habit of project managers building time buffers into non-critical paths that feed into critical ones in a project network has resulted in almost late completion of projects. The corporations are dealing firmly with time overruns that cripple their budgets, damage their reputations and tax their cash flows with paid-out penalties. It is estimated that 50 percent of the software projects that are successfully completed, are not as successful as they should be.

Jorgensen^[16] emphasized that the simulation approach can be used for management of any project but he time estimates for project management of information systems are still less accurate than any other estimates in the project management cycle.

MATERIALS AND METHODS

Method for computation of optimum cost and schedule:

Initialization steps:

Step 1: Each activity is assumed to have a known Normal cost if completed in a Normal time and a (larger) Crash Cost if completed in a (shorter) Crash

time. Compute the cost (i.e., cost per unit time) for each activity according to the following formula:

Cost slope = (Crash cost-Normal cost)/(Normal time-Crash time)

Step 2: Enumerate all the paths through the project network and list them with their normal time durations in the path list. Identify the critical path(s) as those with longest duration and mark the critical activities.

Step 3: Identify the normal project duration, the normal project cost and the normal critical path.

Iteration steps:

Step 4: Select that subset of critical activities which, when compressed in parallel, enable all current critical paths to become shorter and do so at the least group marginal cost, where the group marginal cost for a subset of critical activities is the sum of the marginal costs for activities in the group.

Step 5: Compress the selected critical activities until one or both of the following two conditions occurs: (i) one (or more) of the compressed activities becomes fully crashed (i.e., is reduced to crash time); or (ii) a new path becomes critical.

Step 6: Record the selected activities, number of time periods compressed, the new project duration, the group marginal cost for the selected activities, the added cost resulting from the compression, the new total direct cost and the new critical path (if any) as items in the breakpoint table for this iteration. Update the compression availabilities and the path list to reflect the reduction in path lengths resulting from the selected compression.

Step 7: Repeat steps 4 through 6 until all activities on some (any) critical path become fully crashed. At this point the breakpoint table is complete, as no further time reduction is possible. Plot the time-cost trade-off graph by linear interpolation between the time/cost pairs which occur in each row of the breakpoint table.

Algorithm:

- 1) Read total number of activities N
- 2) Read total number of nodes M
- 3) Read starting node s[i] of each activity (i,j)
- 4) Read finish node f[i] of each activity (i,j)
- 5) Read OPT[i], MOST[i] and PRE[i] for each activity (j,j)
- 6) Compute MUE[i] and SIGMA[i] for each (i,j)
- 7) Compute T[i], expected time for each activity using normal distribution:

$$T[i] = MUE[i] + SIGMA[i] * (SUM - 6)$$

where, SUM denotes the sum of 12 random numbers

- 8) Compute total normal and crash cost of the project.
- 9) Compute cost slope for each activity
- 10) Crash the activity duration up to their crashing time having the minimum cost slope i.e.,

- Search the activity having minimum cost slope
- Reduce the normal time up to their crash time.
- Compute critical path and project duration by PERT technique.
- Compute total cost of the project

i.e., total cost = cost before crashing + number of days crashed * crashing cost of the crashed activity + indirect cost * total update duration of project

- 11) Repeat step 10 for each activity
- 12) Draw a graph between total cost and duration of the project made in step 10 to obtain the total minimum project cost.
- 13) Stop

An EDP department has various activities to meet the requirements of various departments of a company. The job of EDP department may be divided into various heads in connection with the software development and equipment installation. These tasks can be further divided into various phases. These various phases can be represented in the form of a network, showing them as activities with their time estimations as optimistic, most likely and pessimistic time estimate for the activities to be required for the completion of a software project. These times are used to calculate expected time. The expected time is the average time an activity will take if it were to be repeated on large number of times and is based on the assumption that the activity time follows beta distribution.

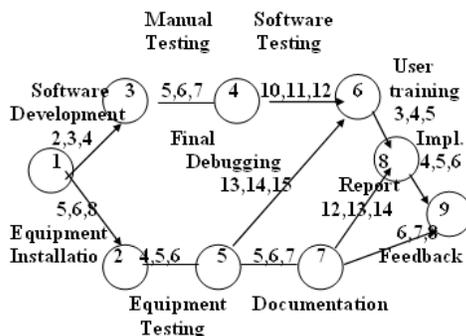


Table 1: Expected time is computed using normal distribution

Activity No.	Opt. time (days)	M. likely time (days)	Pessi time (days)	Expected time (days)
1-2	5	6	8	6
1-3	2	3	4	3
2-3	0	0	0	0
2-5	4	5	6	5
3-4	5	6	7	6
4-6	10	11	12	11
5-6	13	14	15	14
5-7	5	6	7	6
6-8	3	4	5	4
7-8	12	13	14	13
7-9	6	7	8	7
8-9	4	5	6	5

Table 2: Cost slope is computed

Activity	Cost	Crash time (days)	Crash cost (Rs)	Cost slope
1-2	70	3	10	10.000
1-3	60	2	80	20.000
2-3	0	0	0	0.000
2-5	90	1	115	6.250
3-4	25	5	30	1.250
4-6	10	8	160	7.500
5-6	12	10	160	3.330
5-7	12	6	140	6.660
6-8	10	5	100	0.000
7-8	6	7	74	1.270
7-9	6	5	120	20.000
8-9	6	4	78	6.000

Table 3: The results of Project time Vs cost analysis.

Total duration	Total cost (Rs)
35	1215
34	1219
32	1239
28	1204
25	1292
24	1242
20	1227
17	1217

This algorithm is implemented in C++ language for above mentioned network with various input data as shown in Table 1 and 2. The results are shown in Table 3 and a graph is plotted between time and project cost.

RESULTS AND DISCUSSION

In Table 3, the results shows how the total cost of the project is reduced as the total duration is crashed. The minimum cost will be Rs. 1204 for the project duration of 28 days. If we further reduce the days the cost follows up and down nature.

The following graph depicts the results obtained.

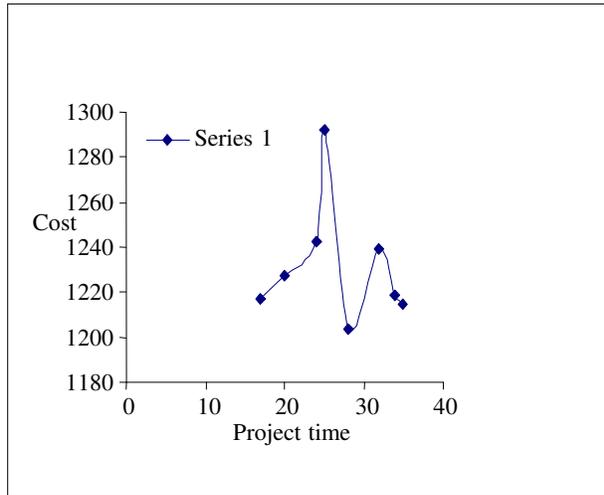


Fig. 4: Project time Vs cost analysis

CONCLUSION

The simulation approach is one of a variety of tools we can use to bring projects back under control and reinforce the use of project management in organizations. The use of simulation to crash project management networks in order to reduce time and cost overruns is a worthy endeavor. The project manager, in collaboration with the IT division, can routinely submit each developed network to crashing (using the simulation program), before major work commences.

The above results shows that the optimization of time and cost process technique can be incorporated as a standard procedure for every project, regardless of its scope, because the simulation program remains the same, project after project. The above results reflect that the time spent on the actual crashing is minimal and the project management schedule can be reduced to a minimum optimum level to save time and money.

We can also further reduce the cost of those activities which are not on critical path but can't further reduce optimal schedule.

In an era where Rapid Application Development (RAD) is the de facto systems development model, the possibility of reducing delays by using simulation for crashing the existing network holds great promise.

REFERENCES

1. Bratley, P., B.L. Fox and L.E. Schrage, 1973. A Guide to Simulation. Springer-Verlag.

2. Elmaghraby, S.E., 1977. Activity Networks: Project Planning and Control by Network Models. John Wiley, New York.
3. Steve, P. Jr. and M.I. Dessouky, 1977. Solving the project time/cost tradeoff problem using the minimal cut concept. *Manage. Sci.*, 24: 393-400.
4. Rehab, R. and R.I. Carr, 1989. Time-cost trade-off among related activities. *J. Construct. Eng. Manage.*, 115: 475-486.
5. Pulat, P.S. and S.J. Horn, 1996. Time-resource tradeoff problem [project scheduling]. *IEEE Trans. Eng. Manage.*, 43: 411-417.
6. Walter, J.G., C. Strauss and M. Toth, 2000. Crashing of stochastic processes by sampling and optimization. *Bus. Process Manage. J.*, 6 : 65-83.
7. Kathryn, A.M., 2004. A simulation approach to the PERT/CPM: time-cost trade-off problem. *Project Manage. J.*, 35: 31-38.
8. Van Slyke, R.M., 1963. Monte carlo methods and the PERT problem. *Operat. Res.*, 33: 141-143.
9. Ameen, D.A., 1987. A computer assisted PERT simulation. *J. Syst. Manage.*, 38: 6-9. <http://portal.acm.org/citation.cfm?id=34619.34620>.
10. Coskun, O., 1984. Optimal probabilistic compression of PERT networks. *J. Construct. Eng. Manage.*, 110: 437-446. <http://cedb.asce.org/cgi/WWWdisplay.cgi?8402651>.
11. Ramini, S., 1986. A simulation approach to time cost trade-off in project network, modeling and simulation on microcomputers. *Proceedings of the Conference*, pp: 115-120.
12. Johnson, G.A. and C.D. Schou, 1990. Expediting projects in PERT with stochastic time estimates. *Project Manage. J.*, 21: 29-32.
13. Badiru, A.B., 1991. A simulation approach to Network analysis. *Simulation*, 57: 245-255.
14. Feng, C.W., L. Liu and S.A. Burns, 2000. Stochastic construction time-cost tradeoff analysis. *J. Comput. Civil Eng.*, 14: 117-126.
15. Grygo, E., 2002. Downscaling for better projects. *InfoWorld*, 62-63.
16. Jorgensen, M., 2003. Situational and task characteristics systematically associated with accuracy of software development effort estimates. *Proceedings of the Information Resources Management Association International Conference*, Philadelphia, PA.