A UNIT BASED CRASHING PERT NETWORK FOR OPTIMIZATION OF SOFTWARE PROJECT COST

Priti Singh
Astt. Prof.-MCA Department
OCM, Bhopal
preitisingh@rediffmail.com

Florentin Smarandache
Chair of Department of Mathematics
University of New Maxico,Gallup U.S.A.
fsmarandache@yahoo.com

Dipti Chauhan
Student M.Tech (CSE) VIth Sem
B.U.I.T, Bhopal
dipti.c_sisodia@rediffmail.com

Amit Bhaghel
Asst Proff. CS-Department
B.U.I.T, Bhopal
amit_kumar_baghel@rediffmail.com

Abstract:

Crashing is a process of expediting project schedule by compressing the total project duration. It is helpful when managers want to avoid incoming bad weather season. However, the downside is that more resources are needed to speed-up a part of a project, even if resources may be withdrawn from one facet of the project and used to speed-up the section that is lagging behind. Moreover, that may also depend on what slack is available in a non-critical activity, thus resources can be reassigned to critical project activity. Hence, utmost care should be taken to make sure that appropriate activities are being crashed and that diverted resources are not causing needless risk and project scope integrity. In this paper we want to present a technique called "Unit Crashing" to reduce the total cost of project. Unit Crashing means to crash the project duration by one unit (day) instead of crashing it completely. This technique uses an iterative approach to perform unit crashing until all activities along the critical path are crashed by desired amount. The output of this method will reduce the cost of project, and is useful at places where cost is of major consideration. 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.

Keywords: Crashing, Uncrashing, PERT, Cost Slope

Introduction:

Complex projects require a series of activities, some of which must be performed sequentially and others that can be performed in parallel with other activities. This collection of series and parallel tasks can be modeled as a network.

In 1957 the Critical Path Method (CPM) was developed as a network model for project management. CPM is a deterministic method that uses a fixed time estimate for each activity. While CPM is easy to understand and use, it does not consider the time variations that can have a great impact on the completion time of a complex project.The *Program Evaluation and Review Technique* (PERT) is a network model that allows for randomness in activity

completion times. PERT was developed in the late 1950's for the U.S. Navy's Polaris project having thousands of contractors. It has the potential to reduce both the time and cost required to complete a project.

Steps in the PERT Planning Process

PERT planning involves the following steps:

1. Identify the specific activities and milestones.
2. Determine the proper sequence of the activities.
3. Construct a network diagram.
4. Estimate the time required for each activity.
5. Determine the *critical path*.
6. Update the PERT chart as the project progresses.

1. Identify Activities and Milestones

The activities are the tasks required to complete the project. The milestones are the events marking the beginning and end of one or more activities. It is helpful to list the tasks in a table that in later steps can be expanded to include information on sequence and duration.

2. Determine Activity Sequence

This step may be combined with the activity identification step since the activity sequence is evident for some tasks. Other tasks may require more analysis to determine the exact order in which they must be performed.

3. Construct the Network Diagram

Using the activity sequence information, a network diagram can be drawn showing the sequence of the serial and parallel activities. For the original activity-on-arc model, the activities are depicted by arrowed lines and milestones are depicted by circles or "bubbles".If done manually, several drafts may be required to correctly portray the relationships among activities. Software packages simplify this step by automatically converting tabular activity information into a network diagram.

4. Estimate Activity Times

Weeks are a commonly used unit of time for activity completion, but any consistent unit of time can be used.A distinguishing feature of PERT is its ability to deal with uncertainty in activity completion times. For each activity, the model usually includes three time estimates:

- *Optimistic time* - generally the shortest time in which the activity can be completed. It is common practice to specify optimistic times to be three standard deviations from the mean so that there is approximately a 1% chance that the activity will be completed within the optimistic time.
- *Most likely time* - the completion time having the highest probability. Note that this time is different from the *expected time*.
- *Pessimistic time* - the longest time that an activity might require. Three standard deviations from the mean is commonly used for the pessimistic time.

PERT assumes a beta probability distribution for the time estimates. For a beta distribution, the expected time for each activity can be approximated using the following weighted average:

Expected time = (Optimistic + 4 x Most likely + Pessimistic) / 6

This expected time may be displayed on the network diagram.

To calculate the variance for each activity completion time, if three standard deviation times were selected for the optimistic and pessimistic times, then there are six standard deviations between them, so the variance is given by:

$$\text{Variance} = [\ (\ \text{Pessimistic} - \text{Optimistic}\ ) / 6\ ]^2$$

5. Determine the Critical Path

The critical path is determined by adding the times for the activities in each sequence and determining the longest path in the project. The critical path determines the total calendar time required for the project. If activities outside the critical path speed up or slow down (within limits), the total project time does not change. The amount of time that a non-critical path activity can be delayed without delaying the project is referred to as *slack time*.

If the critical path is not immediately obvious, it may be helpful to determine the following four quantities for each activity:

- ES - Earliest Start time
- EF - Earliest Finish time
- LS - Latest Start time
- LF - Latest Finish time

These times are calculated using the expected time for the relevant activities. The earliest start and finish times of each activity are determined by working forward through the network and determining the earliest time at which an activity can start and finish considering its predecessor activities. The latest start and finish times are the latest times that an activity can start and finish without delaying the project. LS and LF are found by working backward through the network. The difference in the latest and earliest finish of each activity is that activity's slack. The critical path then is the path through the network in which none of the activities have slack.

The variance in the project completion time can be calculated by summing the variances in the completion times of the activities in the critical path. Given this variance, one can calculate the probability that the project will be completed by a certain date assuming a normal probability distribution for the critical path. The normal distribution assumption holds if the number of activities in the path is large enough for the central limit theorem to be applied.

Since the critical path determines the completion date of the project, the project can be accelerated by adding the resources required to decrease the time for the activities in the critical path. Such a shortening of the project sometimes is referred to as *project crashing*.

6. Update as Project Progresses

Make adjustments in the PERT chart as the project progresses. As the project unfolds, the estimated times can be replaced with actual times. In cases where there are delays, additional resources may be needed to stay on schedule and the PERT chart may be modified to reflect the new situation.

Benefits of PERT

PERT is useful because it provides the following information:

Expected project completion time.

- Probability of completion before a specified date.
- The critical path activities that directly impact the completion time.

- The activities that have slack time and that can lend resources to critical path activities.
- Activities start and end dates.

Crashing:

Crashing refers to a particular variety of project schedule compression which is performed for the purposes of decreasing total period of time (also known as the total project schedule duration). The diminishing of the project duration typically take place after a careful and thorough analysis of all possible project duration minimization alternatives in which any and all methods to attain the maximum schedule duration for the least additional cost The objective of crashing a network is to determine the optimum project schedule. Crashing may also be required to expedite the execution of a project, irrespective of the increase in cost. 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

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

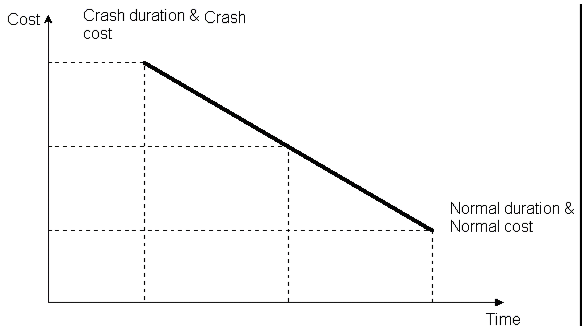

Fig. 1: Linear time and cost trade-off for an activity

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:
Cost slope = (crash cost-normal cost)/ (normal duration 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.

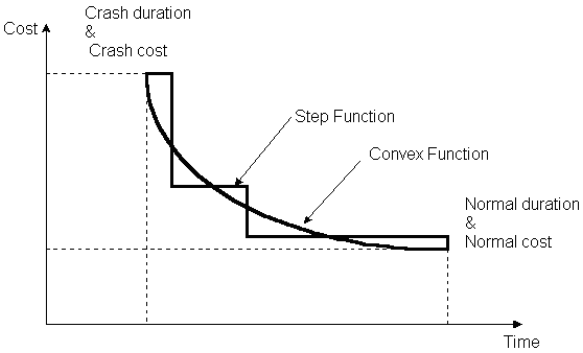

Fig. 2: Non-linear time and cost trade-off for an activity

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.

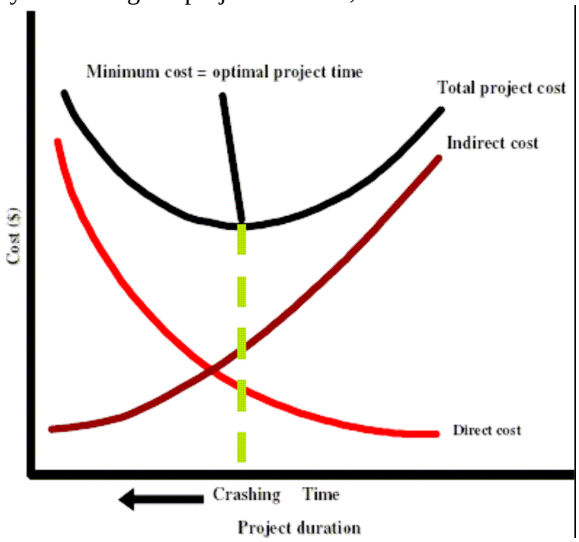

Fig. 3: Project time-cost relationship

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.

Literature review:

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 (λ) 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 (CAPERTSlM), an instructional tool to teach project management techniques. 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 Method:

Step1: Calculate Earliest time Estimates for all the activities. It is calculated as

T_E = Maximum of all ($T_E{}^j + t_E{}^{ij}$) for all i , j leading into the event.

where $T_E{}^j$ is the earliest expected time of the successor event j.

$T_E{}^i$ is the earliest expected time of the predecessor event i. and

$t_E{}^{ij}$ is the expected time of activity ij.

Step2: Calculate Latest time Estimates for all the activities. It is calculated as

T_L = Minimum of all (T_L^i - t_E^{ij}) for all i, j leading into the event

where T_L^i is the latest allowable occurrence time for event i.

T_L^j is the latest allowable occurrence time for event j and

t_E^{ij} is the expected time of activity ij.

Step3: After knowing the T_E and T_L values for the various events in the network, the critical path activities can be identified by applying the following conditions:

1) T_E and T_L values for the tail event of the critical activity are the same i.e.,
   $T_E^i = T_L^i$.
2) T_E and T_L values for the head event of the critical activity are the same i.e.,
   $T_E^j = T_L^j$.
3) For the critical activity, $T_E^j - T_E^i = T_L^j - T_L^i$

Step4: Find the project cost by the formula

Project cost = (Direct cost + (Indirect cost*project duration))

Step5: Find the minimum cost slope by the formula

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

Step6: Identify the activity with the minimum cost slope and crash that activity by one day. Identify the new critical path and find the cost of the project by formula

Project Cost= ((Project Direct Cost +
Crashing cost of crashed activity) +
(Indirect Cost*project duration))

Iteration Step:

Step7: In the new Critical path select the activity with the next minimum cost slope, and crash by one day, and repeat this step until all the activities along the critical path are crashed upto desired time.

Step8: At this point all the activities are crashed and further crashing is not possible. The crashing of non critical activities does not alter the project duration time and is of no use.

Step9 To determine optimum project duration, the total project cost is plotted against the duration time given by figure 4.

Further modification: Uncrashing

Step10 Now see if the project cost can be further reduced without affecting the project duration time. This can be done by uncrashing the activities which do not lie along the critical path. Uncrashing should start with an activity having the maximum cost slope. An activity is to be expanded only to the extent that it itself may become critical, but should not affect the original critical path.

Proposed Work:

Step1: Find Earliest time estimates for all the activities, it is denoted as T_E

Step2: Find latest time estimates for all the activities, it is denoted as T_L

Step3: Determine the Critical Path.

Step4: Compute the cost slope (i.e., cost per unit time) for each activity according to the following formula:

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

Step5: Among the critical path identify the activity with the minimum cost slope, and crash the activity by 1 day.

Step6: Calculate the project cost. Identify new critical path.

Project Cost= ((Project Direct Cost + Crashing cost of crashed activity) + Indirect Cost*project duration))

Step7: Now in the new critical path select the activity with the next minimum cost slope, and crash by one day.

Step8: Repeat this process until all the activities in the critical path have been crashed by 1 day.

Step9: Once all the activities along the critical path are crashed by one day, Repeat the process again i.e. goes to step5.

Step10: Find the minimum project cost and identify the activities which do not lie along the critical path

Step10: Now perform uncrashing. i.e uncrash the activities which do not lie along the critical path.

For Example:

To explain the process of crashing a network to reach the optimum project schedule, let us consider the network shown in figure 1. With each activity is associated normal direct cost and crash direct cost, the normal duration time and crash duration time. The complete data is given in table 1.The network has been drawn for normal conditions and the times shown along the arrows are normal duration times.

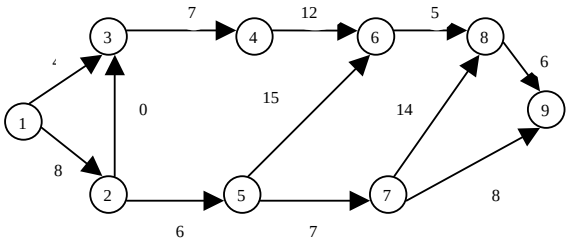

Figure 1

Activity	Normal		Crash		Δt	Δc	Δc/Δt
	Time	cost	Time	cost			
1--2	8	7000	3	10000	5	3000	600
1--3	4	6000	2	8000	2	2000	1000
2--3	0	0	0	0	0	0	0
2--5	6	9000	1	11500	5	2500	500
3--4	7	2500	5	3000	2	500	250
4--6	12	10000	8	16000	4	6000	1500
5--6	15	12000	10	16000	5	4000	800
5--7	7	12000	6	14000	1	2000	2000
6--8	5	10000	5	10000	0	0	0
7--8	14	6000	7	7400	7	1400	200
7--9	8	6000	5	12000	3	6000	2000
8--9	6	6000	4	7800	2	1800	900

Table 1

Result of Calculations based on Unit Crashing

Activity crashed	Weeks saved	Project duration	Direct cost	Indirect cost	Total cost
Nil	0	41	86500	41000	127500
7---8	1	40	87500	40000	127500
2--5	1	39	87200	39000	126200
1--2	1	38	87800	38000	125800
8--9	1	37	88700	37000	125700
5--6	0	37	89500	37000	126500
7--8	1	36	89700	36000	125700
1--2	1	35	90300	35000	125300
8--9	1	34	91200	34000	125200
1--2	1	33	91800	33000	124800
1--2	1	32	92400	32000	124400
3--4	0	32	92650	32000	124650
7--8	0	32	92850	32000	124850
2--5	1	31	93350	31000	124350
3--4	0	31	93600	31000	124600
2--5	1	30	94100	30000	124100
2--5	0	30	94600	30000	124600
1--2	0	30	94700	30000	124700
1--3	1	29	95700	29000	124700
2--5	0	29	96200	29000	125200
4--6	1	28	97700	28000	125700
2--5	0	28	98200	28000	126200
4--6	1	27	99700	27000	126700
5--6	0	27	100500	27000	127500
4--6	1	26	102000	26000	128000
5--6	0	26	102800	26000	128800
4--6	0	26	104300	26000	130300
7--8	1	25	104500	25000	129500
Uncrashing		30	93400	30000	123400

Table:2

Results and discussion:

In Table 2 the results shows how the total cost of the project is reduced as the total duration is crashed. Before uncrashing the minimum cost of project is Rs. 124100 for the project duration of 30 days and after uncrashing the minimum cost will be Rs. 123400 for the project duration of 30 days. The following graph depicts the results obtained.

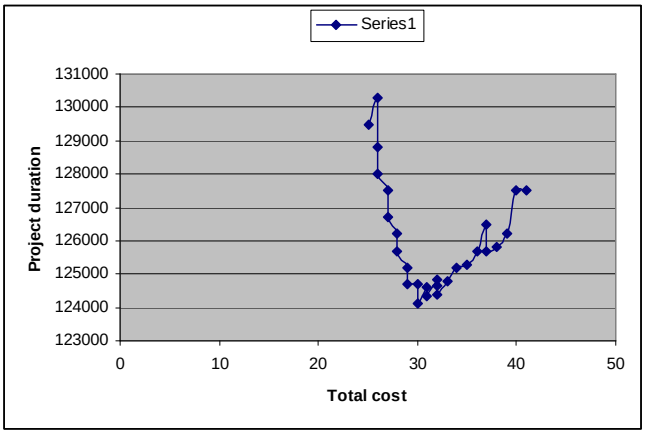

Figure: 4 Project duration Vs cost analysis

Conclusion:

In this paper the algorithm proposed for unit crashing reduces the cost of project. When activities are crashed by one day then only the crashing cost corresponding to one day is increased thereby reducing the project duration as well as cost. A C++ program is been developed to achieve the above results. This approach is well suitable for places where cost is of major consideration.

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.
17. P.K. Suri and Bharat Bhushan Dept of Computer Science and Applications, Kurukshetra University, Kurukshetra (Haryana), India Department of Computer Science and Applications, Guru Nanak Khalsa College, Yamuna Nagar (Haryana), India 2008 Simulator for Optimization of Software Project Cost and Schedule Journal of Computer Science 4 (12): 1030-1035, 2008 ISSN 1549-3636 © 2008 Science Publications.