COMPUTING DURATION, SLACK TIME, AND CRITICALITY UNCERTAINTIES IN PATH-INDEPENDENT PROJECT NETWORKS

Ryan G. Rosandich, Ph.D., University of Minnesota Duluth
Santiago Erquicia, University of Minnesota Duluth

Abstract
Monte Carlo simulation is the currently accepted method for quantifying uncertainty in projects. It is the goal of the research presented in this paper to develop a purely computational technique, based on traditional probability theory, for quantifying project uncertainty with accuracy equal to or greater than that of Monte Carlo simulation. Series and parallel operators have been developed for combing task uncertainties in project networks. The operators were used to compute overall project uncertainty given individual task uncertainty, and to calculate slack and the degree of criticality for each task. Results equaled or exceeded the accuracy of Monte Carlo simulation for the networks investigated.

Introduction
There are no facts about the future. This simple statement seems obvious, yet engineers and managers are always trying to predict or manipulate the future in some way. In a typical engineering management program, for example, courses are taught with topics like forecasting, operations management, and project management in which predicting, planning, and even controlling future events are discussed.

Dealing with the uncertainty inherent in future events is especially challenging in the field of project management. A project manager’s job is to plan and carry out a complex sequence of tasks, and to do so on schedule and within budget. The schedule and budget are based on estimates of how long tasks will take to accomplish and how much various materials will cost, and include a great deal of uncertainty.

In addition to those uncertainties, project managers face external or environmental causes of uncertainty such as limited resources, unpredictable weather, and variable funding, to name a few. The management of project uncertainty is further complicated by the fact that the risks are not independent. A variation in the schedule, for example, will often have an effect on resources and cost.

The management of the unknowns and/or uncertainties involved with project management is known as *project risk management* (PRM), and it is currently one of the main areas of interest in the project management community (Raz and Michael, 2001). The Project Management Institute (PMI), the largest professional organization in the world dedicated to project management, has identified risk management as one of the eight main areas of the Project Management Body of Knowledge (PMI, 2000), and one of the most active PMI Specific Interest Groups (SIG) is the one dedicated to risk management. The Risk Management SIG lists 108 risk management software tools on its website (RiskSIG, 2003), and a recent survey of research on risk management (Williams, 1995) identified 241 references.

Risk management experts Chapman and Ward have recently argued that the term ‘risk’ should be replaced with ‘uncertainty’ in project management (Ward and Chapman, 2003). They argue that risk management is too narrowly focused on events, and that risk always has a negative connotation. They contend that replacing PRM with the broader project uncertainty management (PUM) will allow project managers to better understand the sources of uncertainty in projects, and to take advantage of the opportunities related to uncertainty as well as dealing with the threats.

There is great interest in improving PUM by quantifying the effect of uncertainty on project outcomes. Projects could be managed more effectively if knowledge of uncertainty could be incorporated into the project planning, monitoring, and control processes.

A popular approach to the quantification of uncertainty utilizes Monte-Carlo simulation. With this approach the expected values of duration and cost for each task are replaced with probability distributions that reflect the uncertainty in those estimates, and the project is executed repeatedly in simulated time to determine the effect of the task uncertainties on project outcomes.

After many simulations probability distributions can be determined for outcomes like total project cost and project duration. Those distributions reflect the level of uncertainty in the entire project, and represent a far more realistic predictor of actual project outcomes.

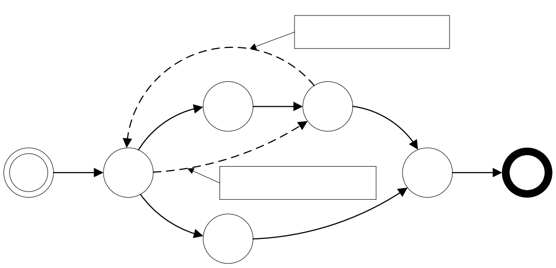
Simulation-based methods are widely used in project planning, and most risk management software packages rely on Monte-Carlo simulation. Simulations can involve thousands of iterations of thousands of calculations, however, so they are complex and time consuming to perform. For this reason they are used primarily in the planning and decision making stages of a project, and not during actual project execution. It is desirable to develop a method of calculating project uncertainty that is computationally simpler, so it could be integrated with project management software and used in all phases of project management.

Project Networks and Uncertainty

Project managers have traditionally relied upon network methods like the Program Evaluation and Review Technique (PERT) and the Critical Path Method (CPM) for planning and managing projects. Many books on project management present the PERT/CPM techniques (e.g. Cleland and King, 1988; Kerzner, 2001; Meridith and Mantel, 2000; Rosenau, 1992; Stuckenbruck, 1981), and virtually all project management software packages implement some form of PERT/CPM. These techniques rely on estimates of the expected duration and expected cost of each project task as input, and generate project schedules and budgets based on the logical predecessor/successor relationships between the tasks.

Network structure. Formally, a PERT/CPM project network consists of a set of tasks (nodes) connected by predecessor-successor relationships usually represented as arrows. A project network has two unique nodes, a Start (initial) node that has no predecessors and one or more successors, and an End (terminal) node that has one or more predecessors and no successors. All of the other nodes in a project network are called internal nodes and they represent tasks. Each task must have at least one predecessor (no spontaneous tasks) and one successor (no dead ends). Each internal node is usually assigned a value or weight. In the case of a project network, the node value is the task duration.

Exhibit 1. Project Network Structure.



A path is said to exist between two nodes A and B if node B can be reached form node A by following predecessor-successor relationships. In the example network shown in Exhibit 1, two paths exist between the Start and End nodes (Start-A-C-D-E-End and Start-A-B-E-End). The length of a path is the sum of the durations of each of the tasks encountered as the path is traversed. In a project network, the longest path from the Start to the End node is called the *critical path*.

A legitimate PERT/CPM network is also *acyclic* and *compact* (Nasution, 1994). In an acyclic network, no path exists that passes through any task more than once. This eliminates the possibility of loop-backs. In Exhibit 1, the link from task D to task A is not allowed because it is a loop-back relationship that creates a cycle.

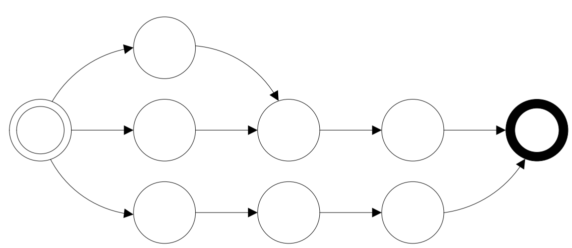
A compact network contains no redundant relationships. This means that only immediate predecessors are shown in a project network. In Exhibit 1 the link from task A to task D is not allowed because it represents a redundant relationship.

Quantification of uncertainty. Probability distributions are usually used to represent uncertain task durations. For this research uncertain task durations have been represented by discrete random variables with a minimum duration, a maximum duration, and several duration increments between the minimum and maximum values. The number of duration increments is determined by a value defined as the *precision*. The probability of each possible duration occurring is determined by a probability distribution that is assigned to the particular task, and as usual the total probability of all possible outcomes must be 1. The triangle distribution was used to represent task duration uncertainty in this research, but any finite distribution could be used.

It is often desirable to compute the effects of individual task uncertainties on the overall project outcome. For example, given triangle probability distributions that represent uncertain task durations, it is desirable to combine those individual uncertain durations in some way to compute the overall project duration, and the uncertainty therein. Two operators, a series operator and a parallel operator, have been developed to help accomplish this.

Two tasks are defined as being in *series* if the first task has the second task as its only successor, and the second task has the first task as its only predecessor. For example, tasks E and F in Exhibit 2 are in series.

Exhibit 2. Example Project Network.



When task durations are certain, the series result can be determined by simply adding the two durations. When durations are represented by discrete random variables, however, all possible outcomes must be determined, along with the probability of each. The possible outcomes are determined by adding all of the possible combinations, while the joint probabilities of each of the combinations are determined by multiplying the marginal probabilities for each task.

A simplified example is shown in Exhibit 3. Task E has possible durations of 10, 11, and 12 days with the marginal probabilities shown, and task F has possible durations of 9, 10, and 11 days, each with the probabilities shown. The resulting series outcomes, along with the probability of each, are shown in the table. Note that some outcomes (e.g. 20, 21, and 22) occur more than once, so the total probability of that outcome must be determined by adding.

Exhibit 3. Example Series Computation.

E \ F	**9 (.30)**	**10 (.40)**	**11 (.30)**
10 (.20)	19 (.06)	20 (.08)	21 (.06)
11 (.50)	20 (.15)	21 (.20)	22 (.15)
12 (.30)	21 (.09)	22 (.12)	23 (.09)

All possible outcomes, and their probabilities, are shown in Exhibit 4. This represents the total duration of the series combination of tasks A and B, and the uncertainty therein.

Exhibit 4. Results of Series Computation.

Duration	**Probability**
19	.06
20	.23
21	.35
22	.27
23	.09

Two tasks are defined as being in *parallel* if they have identical predecessors and successors. For example, tasks A and B in Exhibit 2 are in parallel.

When task durations are certain, the parallel result can be determined by simply taking the maximum of the two durations. As in the series case, however, all possible outcomes must be determined when durations are represented by discrete random variables, along with the probability of each. The possible outcomes are determined by taking the maximum of all of the possible combinations, while the joint probabilities are determined once again by multiplying the marginal probabilities for each task.

A simplified example is shown in Exhibit 5. Task A has possible durations of 10, 11, and 12 days with the marginal probabilities shown, and task B has possible durations of 9, 10, and 11 days, each with the probabilities shown. The resulting parallel outcomes, along with the probability of each, are shown in the table. As before, the probabilities of outcomes that occur more than once are determined by adding.

Exhibit 5. Example Parallel Computation.

A \ B	**9 (.30)**	**10 (.40)**	**11 (.30)**
10 (.20)	10 (.06)	10 (.08)	11 (.06)
11 (.50)	11 (.15)	11 (.20)	11 (.15)
12 (.30)	12 (.09)	12 (.12)	12 (.09)

All possible outcomes, and their probabilities, are shown in Exhibit 6. This represents the total duration of the parallel combination of tasks A and B, and the uncertainty therein.

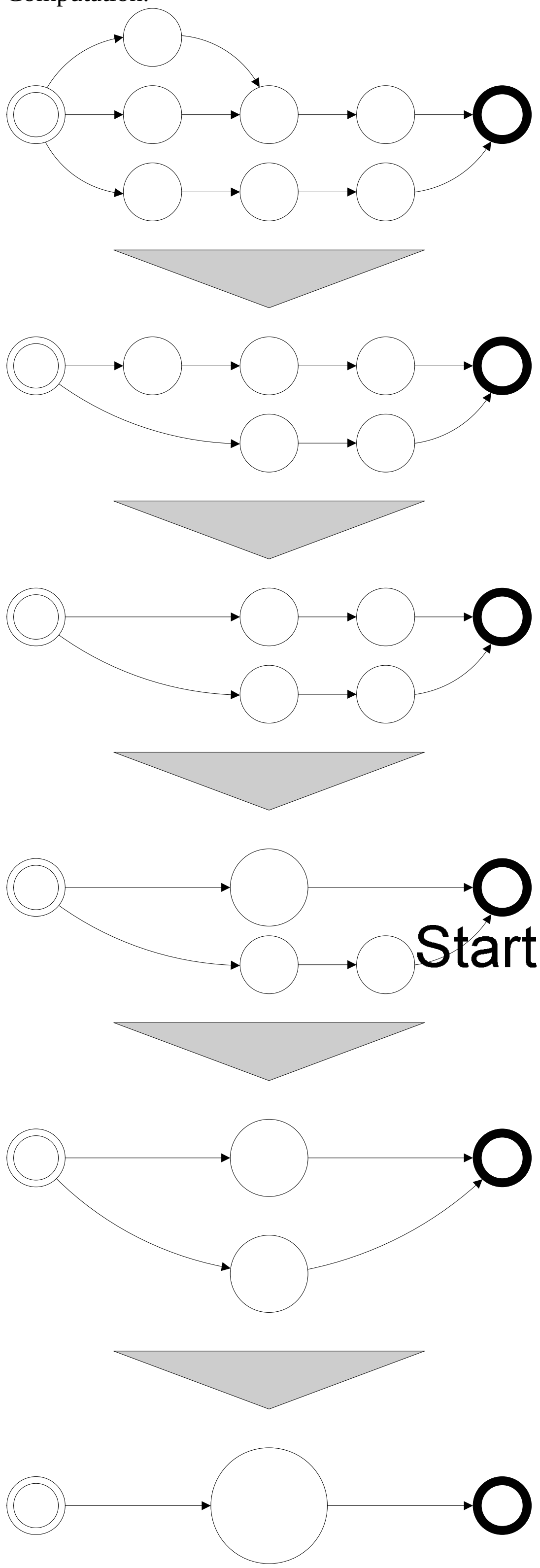
Exhibit 6. Results of Parallel Computation.

Duration	**Probability**
10	.14
11	.56
12	.30

A procedure has been developed that applies the series and parallel operators repeatedly to a network to compute the overall project duration and uncertainty. The procedure first searches for a series relationship between two tasks, and computes the result if one is found. The network is then searched for a parallel relationship between two tasks, and again the result is computed if such a relationship is found. The procedure is repeated until neither a series nor parallel relationship is found.

The result of applying the procedure to the network in Exhibit 2 is shown in Exhibit 7. Note that tasks are combined in serial and/or parallel until the network is reduced to a single task. The duration of this task is the overall project duration, and the uncertainty in this task represents the overall uncertainty in the project duration.

Exhibit 7. Illustration of Network Uncertainty Computation.



Computation of slack times. In a project network without uncertainty, slack is defined as the difference in duration between the critical path and a given path. Tasks on the critical path have zero slack, while all others have positive slack. In an uncertain network the computation of slack becomes far more complex. There may be many possible critical paths in an uncertain network, each with some probability of being critical.

In order to calculate the slack for a particular task, its uncertain duration must be compared with all other possible parallel paths through the network. To accomplish this, one more operator, the *negate* operator, is required. This operator essentially reverses the time direction of a task duration and its uncertainty, so uncertain durations can be 'subtracted' in series as well as added.

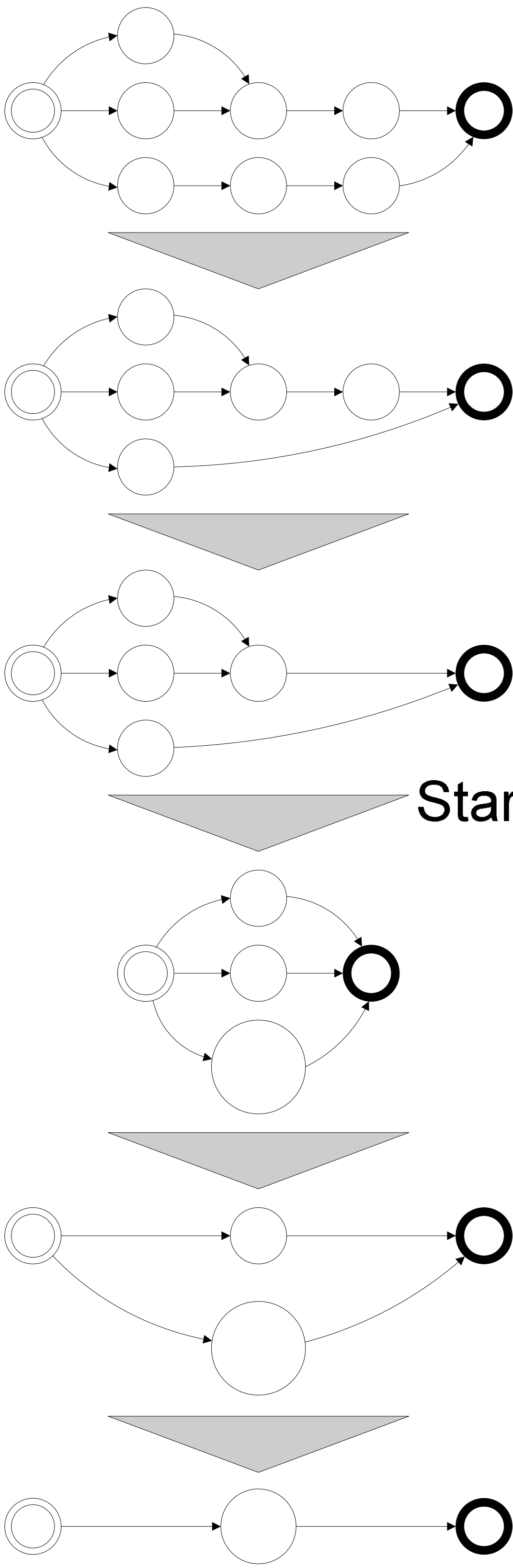
Consider, for example, the calculation of slack for task B in the project network shown in Exhibit 2. This task can be compared with task A to see the probability that task B will be critical or have slack, but the effect of the uncertainties in the other tasks (C, D, E, F, and G) must also be considered. This is accomplished by first combining tasks E, F, and G in series, and then combining the result in series with the negation of tasks D and C. The resultant task is then effectively in parallel with task A, so they can be combined using the parallel operator. The result can then be compared with task B, again using the negation operator, to determine the slack for task B. This process is illustrated graphically in Exhibit 8.

Because slack is the difference between two uncertain task durations it can take on both positive and negative values. Tasks with completely negative slack are always critical, and tasks with completely positive slack are never critical. Tasks with slack on both sides of zero have some probability of being critical, and some probability of being non-critical. The area beneath the negative portion of the slack curve is defined as *criticality*, the probability of the task being critical.

Evaluation of the computational methodology. The computational technique described above was tested against Monte Carlo simulation. The method was tested on many networks, and used to calculate many uncertain quantities including task start times, task end times, task slack and criticality, and overall project duration. The results of these computations were compared to the simulation results for the same networks.

Start AB C

Exhibit 8. Illustration of Slack Computation.



Results and Conclusions

The results of applying the computational method to the example network shown in Exhibit 2 are discussed next. These results are representative of the results achieved with the many other network configurations that were tested.

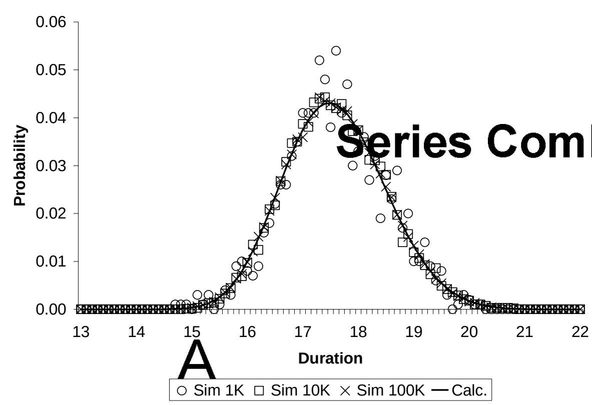
Triangle probability distributions were used to represent the uncertainties in the task durations. The minimum duration, maximum duration, and most likely duration for each task are shown in Exhibit 9.

Exhibit 9. Minimum, Most Likely (ML) and Maximum Values for Tasks Used in Example.

Task	Min.	ML	Max.
A	5	6	8
B	4	5	7
C	4	5	7
D	4	5	7
E	4	5	7
F	4	5	7
G	5	6	8

Project duration. The overall duration of the project and its uncertainty were calculated using the series and parallel operators as previously described in Exhibit 7. The calculated results were then compared to the results of three Monte Carlo simulations, with 1000, 10000, and 100000 iterations. The results of this comparison are shown graphically in Exhibit 10. The mean absolute differences between each of the simulated results and the calculated results were also generated, and they are shown in Exhibit 11.

Exhibit 10. Calculated and Simulated Results of Project Duration Uncertainty.



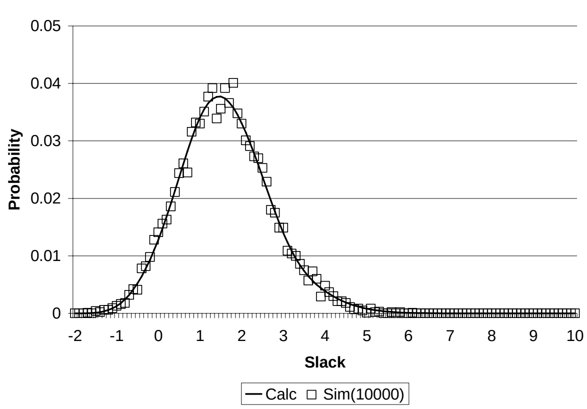
Note that the results agree more closely, and the differences become smaller, as the number of iterations in the simulation increases. This would suggest that the calculated result, generated in a single pass, is more accurate then the Monte Carlo simulation with many iterations. It would also suggest that, as the number of iterations becomes very large, the simulation results in fact approach the calculated result as a limit.

Exhibit 11. Mean Absolute Difference Comparison Between Calculated and Simulated Results.

Iterations	Mean Absolute Difference
1,000	.001722
10,000	.000515
100,000	.000194

Slack times. The uncertain slack times, and their related criticalities, were calculated using the series, parallel, and negate operators as previously illustrated in Exhibit 8. The calculated results for task B were then compared to the results of a Monte Carlo simulation with 10000 iterations. The results of this comparison for the task B slack computation and simulation are shown graphically in Exhibit 12.

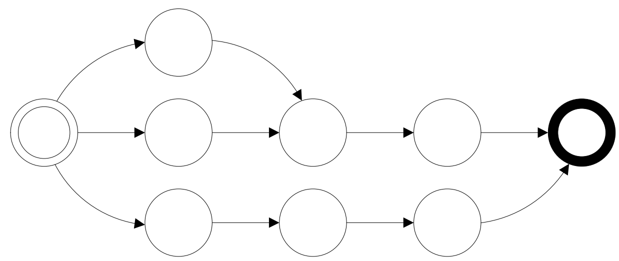
Exhibit 12. Calculated and Simulated Results of Slack for Task B.



The computed slack results for each of the tasks were also examined in order to determine the effects of the series and parallel relationships on slack times. The criticality results are shown in Exhibit 13. The results indicated that, for task in series, slack times and criticalities are identical. For example, tasks E, F, and G are all in series and each has a criticality of 48.5%

The results also indicated that when tasks are in parallel, criticality divides among the various parallel paths. For example, tasks A and B are in parallel and have criticalities of 45.4% and 6.1% respectively. The total criticality for the parallel combination is the sum of the individual criticalities, or 51.5%. Note also that the total criticality at the Start and End nodes sums to 100%.

Exhibit 13. Criticality Results for All Network Tasks.



Limitations. A major limitation of the computational method described herein is the fact that the method requires that the project network be completely simplified using the series and parallel operators. Although this is possible for many practical networks, there are also many networks for which simplification is not possible due to path dependencies.

Exhibit 14. Project Network with Path Dependencies.

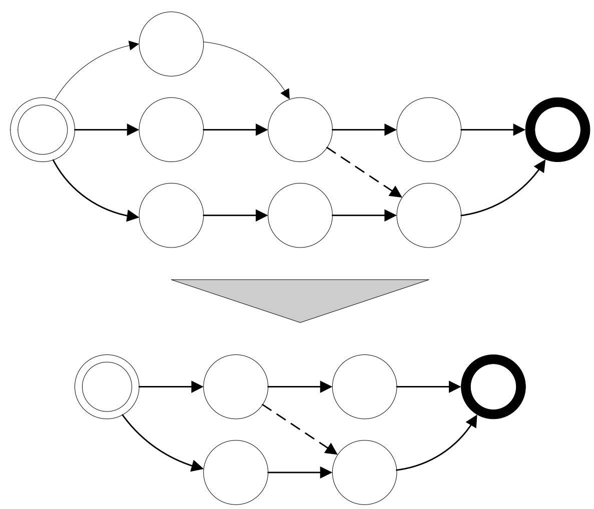


Exhibit 14 shows the example project network that has been used throughout this paper with one additional relationship added from task C to task G. Note that the network can only be partially simplified with the series and parallel operators, because task G participates in two paths. Task G could be duplicated (one copy in parallel with D and one in series with EF) and the resulting network could apparently be simplified, but when the final parallel combination was performed it would be between two dependent paths (both contain the uncertainty in task G), and the series and parallel operators assume independence. At this point some other technique such as joint probability computation or Monte Carlo simulation would have to be employed on the simplified network.

Summary. The conclusions based on the evaluation of the computational methodology presented herein can be summarized as follows:

- The method accurately computes the effect of uncertainties in individual task durations on the overall project duration in path-independent networks.
- The method accurately computes the uncertainty distributions in task slack times, and the resulting task criticality.
- Task criticality is in fact the area beneath the negative portion of the slack time distribution.
- Slack time distributions for tasks in series are identical.
- Criticality divides among parallel paths
- The total criticality for the project, as computed at the Start and End nodes, is 100%.
- The method does not work on project networks with path dependencies, but can be used to partially simplify those networks to facilitate further analysis.

Acknowledgments
This research was generously supported by a grant from the Northland Advanced Transportation Systems Research Laboratory, a cooperative research and education initiative of the Minnesota Department of Transportation, the University of Minnesota Center for Transportation Studies and its Intelligent Transportation Systems Institute, and the College of Science and Engineering at the University of Minnesota Duluth.

References

Cleland, D. I. and W. R. King (eds.), *Project Management Handbook*, second edition, Van Nostrand Reinhold (1988).

Kerzner, H. (ed.), *Project Management - A Systems Approach to Planning, Scheduling, and Controlling*, seventh edition, John Wiley and Sons (2001).

Meridith, Jack R. and Samuel J. Mantel Jr., *Project Management - A Managerial Approach*, fourth edition, John Wiley and Sons (2000).

Nasution, S. H., "Fuzzy Critical Path Method," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 24, pp. 48-57 (1994).

PMI, *A Guide to the Project Management Body of Knowledge*, Project Management Institute (2000).

Raz, T. and E. Michael, "Use and Benefits of Tools for Project Risk Management," *International Journal of Project Management*, Vol. 19 pp. 9-17 (2001).

Risk Specific Interest Group (RiskSIG) of the Project Management Institute, "Risk Management Tools", http://www.risksig.com/resource/index.htm (cited Feb. 2003).

Rosenau, Milton D., *Successful Project Management - A Step-by-Step Approach with Practical Examples*, second edition, Van Nostrand Reinhold (1992).

Stuckenbruck, Linn C. (ed.), *The Implementation of Project Management: The Professional's Handbook*, Addison-Wesley (1981).

Ward, S. and C. Chapman, "Transforming Project Risk Management into Project Uncertainty Management," *International Journal of Project Management*, Vol. 21 pp. 97-105 (2003).

Williams, T. M., "A Classified Bibliography of Recent Research Relating to Project Risk Management," *European Journal of Operational Research*, Vol. 85 pp. 18-38 (1995).

About the Authors

Dr. Ryan G. Rosandich is an Associate Professor of Mechanical and Industrial Engineering at the University of Minnesota Duluth. He received his Ph.D. in Engineering Management from the University of Missouri-Rolla in 1994, and M.S. in Computer Science from UMR in 1992. He received a Bachelor of Electrical Engineering from the University of Minnesota in 1980 and has 12 years of industrial experience as an engineer, project manager, and engineering manager for Boise Cascade Corporation. His research interests are in the areas of intelligent manufacturing, automated visual inspection, and project management.

Santiago Erquicia is an engineering management graduate student at the University of Minnesota Duluth. He received a Bachelor of Chemical Engineering from the Instituto Tecnológico de Buenos Aires, Argentina, in 1999 and has 4 years of experience as an operations business analyst and project manager at Transportadora de Gas del Norte, Argentina.