

## A DYNAMIC CRASHING METHOD FOR PROJECT MANAGEMENT USING SIMULATION-BASED OPTIMIZATION

Michael E. Kuhl  
Radhamés A. Tolentino-Peña

Industrial & Systems Engineering Department  
Rochester Institute of Technology  
Rochester, NY 14623 USA

### ABSTRACT

A dynamic simulation-based crashing method is introduced in this research to evaluate project networks and determine the optimum crashing configuration that minimizes the average project cost due to lateness penalties and crashing costs. This dynamic approach will let the user evaluate the project network to determine a crashing strategy at the beginning of the project and also during the life of the project. By reevaluating the project network possible adjustments to the crashing strategy may be identified and implemented. The output of the method includes a distribution of the project completion time, a distribution of the project total cost, and the project cost savings.

### 1 INTRODUCTION

Project management is a tool that is used by many companies to help improve performance and competitiveness. Projects and their execution, in general, require resources. Project management, which is characterized by techniques intended to provide a better use of project resources (Kerzner 2003), can positively impact the profitability of a company.

An important aspect of project management is risk management. Different types of risk are present in any given project, but the emphasis of this research will be focused on schedule/time risk and associated costs. The schedule/time risk essentially implies not completing project activities on time, resulting in a late completion of the project. Late project completion generally has negative effects for the company such as penalty costs and customer dissatisfaction. If a project is running late project managers might be able to bring the project back on track by incorporating additional resources (Eisner, 2002). In project management, this method of mitigating risk is known as *crashing*.

Rosenau and Githens (2005) state crashing is “spend[ing] more money on the project in order to speed

up accomplishment of scheduled activities.” Since crashing a project represents additional costs, crashing decisions need to be made in a cost-effective way. When crashing a project the tradeoff between the crashing cost and the penalty cost needs to be evaluated. A typical scenario involving a project that has potential for being completed late (resulting in a penalty), and may benefit from crashing is illustrated in Figure 1. As crashing of activities is implemented, the total cost of crashing plus the penalty cost may initially decrease. As the crashing amount is increased, diminishing returns will be realized until a point where the total cost may begin to increase. The objective is to determine the optimal crashing point (indicated by the arrow) where the total cost will be minimized.

The crashing method is focused on reducing the time of the activities on the critical path. The critical path is the one that can cause a delay of the project because there is no slack on that path. The traditional method of crashing CPM/PERT networks only considers average activity times for the calculation of the critical path, ignoring the uncertainty related with the duration of the activities. Consequently, other paths that may have a high probability of becoming critical are ignored. As a way to overcome this issue, simulation can be used to model the stochastic nature of the durations of the activities. Incorporating stochastic durations in the crashing process allows the generation of the project completion time distribution and enables the analysis of the real effect that a specific crashing configuration may have on the project.

Several simulation based crashing methods are described in the literature (Bissiri and Dunbar 1999, Haga and Marold 2004, Haga and Marold 2005). These methods are heuristics that are developed to return satisfactory solutions but not necessarily an optimal solution.

The goal of this research is to develop a method that allows project managers to make optimal dynamic, data driven crashing decisions that minimize the average project

cost (the project cost in this research is the sum of crashing costs and penalty costs).

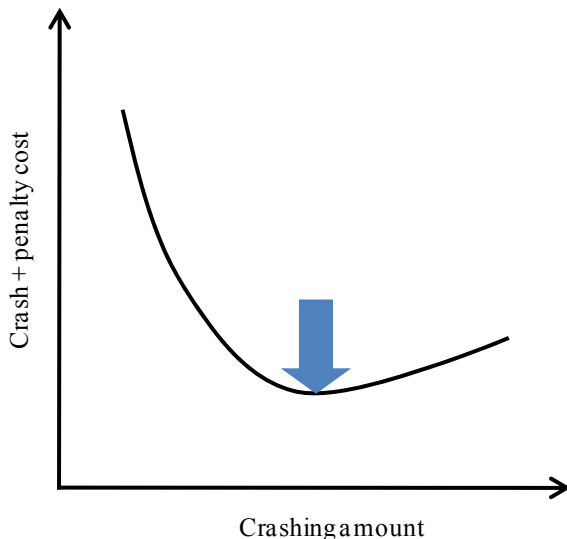


Figure 1: Relationship between crashing amount and the total cost (crash + penalty).

## 2 RELATED WORK

### 2.1 Determining Project Completion Times

The Critical Path Method (CPM) and the Project Evaluation and Review Technique (PERT) methods have been used since the 1950s to estimate the completion time of a project. CPM is a deterministic approach to calculate the duration of a project, and PERT is a probabilistic approach that enhances CPM by considering uncertainty in activity durations by calculating the probability to complete the project by a given time (Lee and Arditi 2006). Hillier and Lieberman (2001) state that even when the original versions of CPM and PERT have some significant differences, with time they have been considered as one technique called CPM/PERT. The PERT method considers the mean and variance of each activity to describe its duration and to represent the uncertainty associated with it; however, PERT only considers the mean times to calculate the critical path, ignoring the variances, thus making a deterministic analysis (Ahuja et al. 1994).

The research of Lu and AbouRizk (2000) presents a CPM/PERT simulation model that incorporates the discrete event modeling approach and a simplified critical activity identification method. Lee (2005) presents a software tool, SPSS, which can be used to determine the probability associated with the completion of the project by a target date specified by the user. Lee and Arditi (2006) describe a new simulation system, S3, which is an improvement over SPSS. An advantage of S3 over SPSS is that S3 calculates

a confidence interval for the project mean duration and also determines the minimum number of simulation runs necessary to have a better estimator of the mean project duration. Simmons (2002) and Pritsker (1986) also describe simulation models that evaluate project networks. These simulation models provide a histogram of the project completion time distribution, which can be used to perform risk analysis.

### 2.2 Simulation-based Crashing Methods

Bissiri and Dunbar (1999) present a method to crash a project network. They suggest the use of simulation to obtain the average time of each activity, the critical path, and the near critical paths. A near critical path in this model is a path which length is smaller than the original completion date but it is larger than the target completion date after crashing. After the path information is collected a linear program is applied to determine the crashing strategy.

Haga (1998) along with Haga and Marold (2004), and Haga and Marold (2005) present a series of papers involving heuristic crashing methods for project management utilizing simulation.

Haga and Marold (2004), propose a simulation-based method that deals with the time-cost trade-off involved with crashing a project. The authors state that “the complete distribution of project completion time needs to be considered when crashing”. The method that they proposed is a two steps approach. The first step is to apply the traditional PERT method to crash the project, and the second step consists in testing each activity that had not been crashed to the upper crashing limit to determine if crashing that activity further reduces the average total cost of the project. The authors considered two sources that can increase the cost of the project, which are crashing costs and overrun costs.

Haga and Marold (2005) developed a method to monitor and control a project. The output of this method is a list of dates at which the project manager “should review the project to decide if activities need to be crashed”. These dates are called crashing points, and they are determined by a backward run through the project network. The crashing points are established at the beginning of the project and they remain fixed during the entire project life.

## 3 METHODOLOGY

The purpose of this research is to develop a dynamic simulation-based analysis method capable of evaluating project networks to answer the following questions:

- What activities should be crashed in order to minimize the average project cost?
- To what extent should the activities be crashed?

- How often should the project network be reevaluated?

The overall procedure is presented in two phases. Phase I considers the evaluation of the project prior to the start of the project. This phase will produce an optimal crashing strategy (with respect to information available prior to the start of the project) as well as recommendations for reevaluation during the Phase II where the dynamic crashing portion of the method is implemented.

This procedure is designed to evaluate the impact that crashing each activity (by integer time units) has on the average project cost. The method is intended to be robust and produce optimum results for analyzing project networks with only one dominant critical path or multiple critical paths.

The output of the method includes a distribution of the project completion time, a distribution of the project total cost, the activities to crash and the extent of the crashing, confidence and tolerance intervals on the project completion time, and the time points at which the project network might be reevaluated.

In the next sections, Phase I and Phase II of the procedure are presented. Although the methods are designed to be used together to maximize the benefit of the method, Phase I can be applied independently from Phase II at the start of the project with Phase II being optional.

### 3.1 Phase I: Optimal Crashing Method Applied Prior to the Start of the Project

The objective of Phase I of the procedure is to obtain the optimal crashing configuration prior to the start of the project that will minimize the total project cost with respect to crashing costs and penalty costs. Phase I involves the following steps:

1. Construct a simulation model of the project network.
2. Identify the potential/feasibility of crashing each activity in the network and the related costs.
3. Utilize a stochastic simulation optimization tool such as Industrial Strength COMPASS (ISC) to determine the optimal project crashing configuration.
4. Proceed to Phase II or implement the optimal crashing solution.

The simulation model is used to determine the project duration and the additional project cost (crash + penalty costs). The first step in using the simulation model is to input the data that describes the project network, which consists of the probability distribution functions (PDF) that represent the activity durations, the crashing cost per time unit for each activity, the predecessors of each activity, and

the target completion time. Although activity times could follow any probability distribution if the appropriate parameters that define the PDF are known, using the beta distribution to represent activity times is common in the field of project management and we will keep with this convention in this paper. The duration of each activity is defined by three estimates consisting of the optimistic, most likely, and pessimistic duration times; these estimates are used to estimate the parameters of the general beta distribution, from which the activity durations are sampled.

Once the information that describes the project network is defined in the simulation model the activity times will be generated, and the starting and completion times of each activity will be calculated. The starting time of each activity will be equal to the time at which all its predecessors are completed. The completion time of each activity is represented by the following expression:

$$ct_i = st_i + t_i - x_i,$$

where  $ct$  represents the activity completion time,  $st$  represents the activity starting time,  $t$  represents the activity duration, and  $x$  represents the number of time units by which the activity is crashed.

After calculating the completion time for each activity the project duration is calculated; the project duration is equal to the longest activity completion time. There is a penalty cost associated with a late completion of the project, and for some projects there is an additional profit associated with early completion. It is necessary to incorporate in the simulation model the functions that represent the penalty cost or additional profit. In this phase of the research linear functions are considered. Finally, the total cost is calculated, which is equal to the crashing cost plus the penalty cost. (For the examples presented in this paper, the simulation model is developed using the C++.)

The simulation model is used to generate the distribution of the project cost and project completion time when no crashing is applied to the project network; the distribution of project cost is the baseline used to determine the level of risk associated with penalty costs.

The simulation model uses integer decision variables that represent the number of time units by which an activity is crashed; a particular set of values for these integer decision variables represents a crashing configuration. The simulation model interacts with an optimization engine with the purpose of determining the crashing configuration with the minimum average total cost.

The optimization engine used in this step of the methodology is Industrial Strength COMPASS (ISC) (Xu et al. 2007). ISC is a tool which is derived from the COMPASS framework developed by Hong and Nelson (2006) for locally convergent, discrete optimization-via-simulation (DOvS). To utilize ISC the C++ simulation model is integrated into the ISC code. ISC requires inputs such as an in-

itial solution, the range of possible values for each decision variable (crashing amounts), and the confidence level desired for the solution; these inputs must be provided in a separate text file from which ISC reads them. (For a complete list of the inputs required to use ISC please refer to Xu et al. 2007). ISC searches the feasible region defined by the potential activities that can be crashed and returns an optimal solution within the specified tolerance.

Next we present an example illustrating the Phase I method.

### 3.2 Phase I: Example

The following project network, is based on an example presented by Haga (1998), to illustrate Phase I of the crashing method. Table 1 shows the 36 activities in the network along with the precedence relationships. The project network is depicted graphically in Figure 2. In this example, the network contains only 1 possible critical path which will be the focus of this example. The activities on this critical path each have a potential of crashing up to 3 time units. The respective parameters of the activity time distributions and unit crashing costs are shown in Table 2. The target completion time of the project is time 180, and the equation defining the penalty cost for late completion is

$$P = \begin{cases} 0, & \text{if } T \leq 180 \\ 10(T - 180), & \text{if } T > 180, \end{cases}$$

where  $T$  is the resulting completion time of the project.

ISC was used to obtain the optimal crashing configuration which is to crash activity 29 three time units. The original project without crashing and the project with the optimal crashing configuration were each simulated 50,000 times to produce the distribution of completion time (Figure 3) and the cumulative distribution of the total project cost (Figure 4). In addition, Table 3 provides the average and standard deviation of the project duration and project cost. The optimal crashing configuration generated by ISC is consistent with the one presented by Haga (1998).

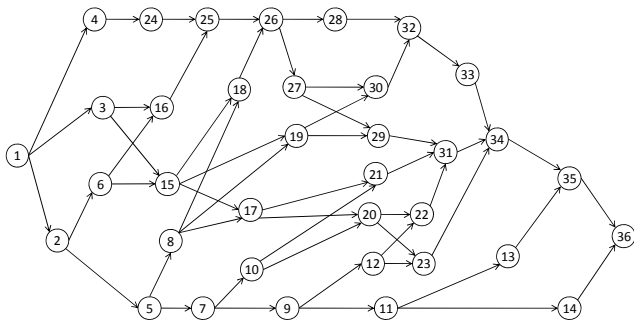


Figure 2: Project network used for example (based on Haga 1998).

Table 1: Dependency relationships for the project network used for the example.

Activity	Predecessors	Activity	Predecessors
1	-	19	8, 15
2	1	20	10, 17
3	1	21	10, 17
4	1	22	12, 20
5	2	23	12,20
6	2	24	4
7	5	25	16, 24
8	5	26	18, 25
9	7	27	26
10	7	28	26
11	9	29	19, 27
12	9	30	19, 27
13	11	31	21, 22, 29
14	11	32	28, 30
15	3, 6	33	32
16	3, 6	34	23, 31, 33
17	8, 15	35	13, 34
18	8, 15	36	14, 35

Table 2: Minimum (a), most likely (ml), and maximum (b) duration and crashing cost for each activity.

Activity	a	ml	b	Crash cost
1	10	20	30	9
4	12	14	16	8
24	14	18	22	4
25	12	18	30	6
26	10	20	30	9
27	8	12	16	9
29	18	25	32	1
31	10	20	30	8
34	15	20	25	9
35	6	12	18	4

Table 3: Summarized comparison between no crashing and optimal crashing.

	Duration		Cost	
	Average	Std. Dev.	Average	Std. Dev.
Original	179.997	7.63	30.51	44.88
Optimal	176.997	7.63	20.93	34.54

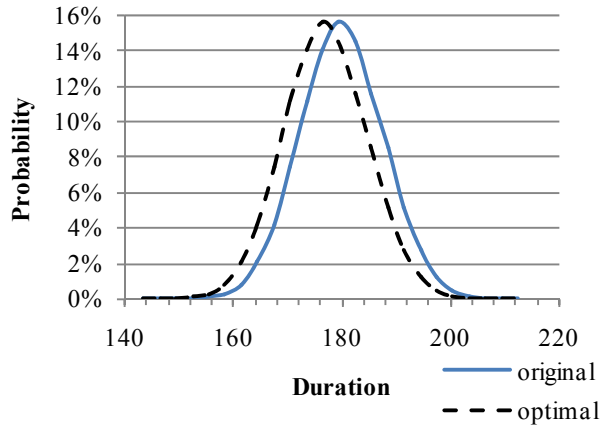


Figure 3: Project completion times comparison – Original versus Optimal.

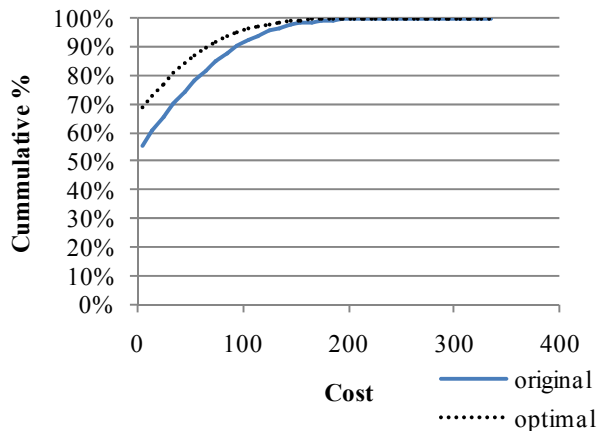


Figure 4: Project costs comparison – Original versus Optimal.

### 3.3 Phase II: Dynamic Crashing

In Phase I, prior to the start of the project an initial optimal crashing configuration is obtained by analyzing the entire project network. This initial optimal crashing configuration considers the uncertainty associated with the duration of all

the activities of the project. As activities are completed the overall uncertainty about the project completion time is reduced, and as a result the initial optimal solution might change. In order to take into account the effect that the uncertainty reduction has on the project completion time and the crashing configuration, the Phase II, dynamic crashing method, is used. The purpose of the dynamic method is to determine the optimal crashing configuration for the remaining activities.

After the project begins, the following steps make up the dynamic method. We will assume that the initial reevaluation points will be the crashing points identified in Phase I.

1. As the project progresses, when the first activity that requires crashing as identified by Phase I is encountered, begin the reevaluation process.
2. Determine which activities are completed or in process when the reevaluation point is reached. For those in process, estimate the remaining processing time.
3. Reevaluate the remaining project network using the simulation model and ISC as described in Phase I.
4. Implement the project network under the new crashing configuration and continue until either
  - a) the next activity that requires crashing is encountered and go to step 2; or
  - b) the project is complete.

For each iteration of steps 2-4 of the dynamic crashing procedure, a new crashing configuration for the remainder of the project will be identified that takes into account the sunk activity times and costs associated with the activities in progress and the activities that have been completed.

### 3.4 Dynamic Crashing Example (Phases I and II)

To illustrate the dynamic crashing method the following project network shown in Figure 5 is used. The activity durations are represented by beta distributions; the minimum (a), most likely (b), and maximum (b) duration for each activity, as well as randomly generated crashing cost are shown in Table 4. The penalty for late completion is equal to 40 cost units per time unit. The target completion time is set to 70 time units.

The project network is evaluated when no crashing is applied (by running 10,000 replications), and the average cost is 21.21 cost units with a variance of 1556.38. The optimal crashing configuration provided by ISC is that activity 2 should be crashed two time units, and activity 9 should be crashed one time unit. In this case the average cost is 14.39 cost units, and the variance is 417.87.

To illustrate Phase II, the dynamic crashing involving reevaluation during the project, we have conducted 10 trials. Each trial represents a single realization of the project

evaluated without crashing, with Phase I crashing only, and with the Phase II crashing. The detailed description for Trial 1 is as follows.

In Trial 1, a realization of the project activity times is generated (via simulation) and applying the dynamic method algorithm, the first reevaluation point will be the start time of activity 2. By the start time of activity 2, activity 1 is completed, and there are no activities in process. The duration of activity 1, which is equal to the start time of activity 2, is equal to 10.21 time units. The network is reevaluated assuming the duration of activity one as being deterministic (a sunk cost); the new optimal crashing configuration is that activities 2 and 9 should be crashed two time units each. This new solution confirms that activity 2 should be crashed, and suggests that activity 9 should also be crashed but by two time units instead of by one time unit as initially suggested. The new average cost is 14.53 with a variance of 278.51.

After crashing activity 2, the project proceeds. The next reevaluation point is the start time of activity 9. Just before the start of activity 9, activities 1 to 4 have been completed and activity 5 is in process. The network is reevaluated considering the duration of activities 1 to 5 as deterministic, and crashing activity 2 two time units. The new optimal crashing configuration indicates that activity 9 shouldn't be crashed, nor any other activity. The average cost is 6 cost units with a variance of 0; that cost is the result of crashing activity 2 twice and resulting in an on-time project.

Similarly, a total of ten trials of the dynamic method were performed. The results of each trial are shown in Table 5 and Table 6. Over these 10 trials, implementing Phase I alone provided an average cost savings of 36% over not crashing at all. The dynamic crashing method provided a cost reduction of 69% over not crashing at all and an additional 52% reduction over using the Phase I crashing method alone. These results demonstrate the types of benefits that can be obtained when the project network is dynamically crashed during the project life.

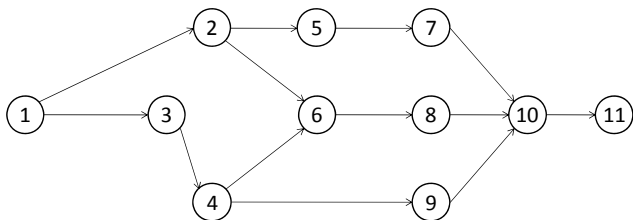


Figure 5: Project network used to evaluate the dynamic method.

Table 4: Minimum (a), most likely (ml), and maximum (b) duration and crashing cost for each activity.

Activity	a	ml	b	Crash Cost
A1	8	10	12	6
A2	6	10	14	3
A3	6	8	10	5
A4	10	15	20	4
A5	12	17	22	5
A6	3	5	7	8
A7	6	9	12	5
A8	4	6	8	5
A9	11	13	15	2
A10	13	15	17	8
A11	5	7	9	7

Table 5: Summary of results of the dynamic method implementation.

Trial	Project Cost		
	No Crashing	Static Crashing	Dynamic Crashing
1	0.0	8.0	6.0
2	18.2	8.0	7.0
3	0.0	8.0	0.0
4	82.0	50.0	43.1
5	61.8	29.8	17.0
6	0.0	8.0	6.0
7	85.6	53.6	17.0
8	45.9	8.0	9.0
9	104.8	72.8	12.0
10	0.0	8.0	6.0
<b>Average</b>	<b>39.8</b>	<b>25.4</b>	<b>12.3</b>

Table 6: Activities crashed in each trial of the dynamic method.

Trial	Crashing Amount		
	A2	A9	A11
1	2	0	0
2	1	2	0
3	0	0	0
4	2	2	0
5	2	2	1
6	2	0	0
7	2	2	1
8	1	0	0
9	2	3	0
10	2	0	0

#### 4 CONCLUSION

We have presented a simulation-based methodology to evaluate project networks and determine an optimal crashing strategy. The methodology has two phases: Phase I, crashing applied prior to the start of the project, and Phase II, dynamic crashing applied during the project life to update the crashing strategy. Applying Phase I to a project network reduces the average cost, and in certain cases the achieved average cost reduction might be enough for the decision makers; however, when Phase II is applied all the uncertainty that has been eliminated is taken into account to produce an updated crashing strategy, which generally yields lowest project costs. These methods utilize a proven stochastic optimization procedure that provides asymptotically optimal results which provides a significant contribution to the literature that currently consists primarily of heuristic methods.

The future research efforts will be focused on conducting a rigorous experimental performance evaluation, investigating alternative methods for determining reevaluation points for Phase II, and investigating the scalability of the computational methods for large project networks. In addition, we intend to generalize the approach to include alternative probability distributions for crashed activity times as opposed to the standard assumption in the literature of integer reductions in activity times for crashed activities.

#### REFERENCES

- Ahuja, H. N., S. P. Dozzi, and S. M. AbouRisk. 1994. *Project Management Techniques in Planning and Controlling Construction Projects*. 2nd Ed., Wiley, New York.
- Bissiri, Y., and S. Dunbar. 1999. Resource allocation model for a fast-tracked project. *International Conference on Intelligent Processing and Manufacturing of Materials*, 635-640.
- Eisner, H. 2002. *Essentials of Project and Systems Engineering Management*. 2nd Ed., Wiley, New York.
- Haga, W. A. 1998. Crashing PERT networks. Ph.D. Dissertation, University of Northern Colorado, Colorado.
- Haga, W. A., and K. A. Marold. 2004. A simulation approach to the PERT CPM time-cost trade-off problem. *Project Management Journal*, 35(2): 31-37.
- Haga, W. A., and K. A. Marold. 2005. Monitoring and control of PERT networks. *The Business Review*, 3(2): 240-245.
- Hillier, F. S., and G. J. Lieberman. 2001. *Introduction to Operations Research*. 7th Ed., McGraw-Hill, New York.
- Kerzner, H. 2003. *Project management: a systems approach to planning, scheduling, and controlling*. 8th Ed., Wiley, Hoboken, NJ.
- Lee, D. 2005. Probability of project completion using stochastic project scheduling simulation. *Journal of Construction Engineering and Management*, 131(3): 310-318.
- Lee, D., and D. Arditi. 2006. Automated statistical analysis in stochastic project scheduling simulation. *Journal of Construction Engineering and Management*, 132(3): 268-277.
- Lu, M., and S. M. AbouRizk. 2000. Simplified CPM/PERT simulation model. *Journal of Construction Engineering and Management*, 126(3): 219-226.
- Nelson, L.J. and B. L. Nelson. 2006. Discrete optimization via simulation using COMPASS. *Operations Research*, 54:115-129.
- Pritsker, A. A. B. 1986. *Introduction to Simulation and SLAM II*. 3rd Ed., Wiley & Sons, Inc, New York.
- Rosenau, M. D. and G. D. Githens. 2005. *Successful Project Management : a Step-by-step Approach with Practical Examples*. 4th Ed., Wiley, Hoboken, N.J.
- Simmons, L. F. 2002. Project management - critical path method (CPM) and PERT simulated with Process Model. *Proceedings of the 2002 Winter Simulation Conference*, Dec 8-11 2002: 1786-1788.
- Xu, J., B. L. Nelson, and L. J. Hong. 2007. *Industrial Strength COMPASS: A Comprehensive Algorithm and Software for Optimization via Simulation*. Website <<http://users.iems.northwestern.edu/~nelsonb/ISC/>>.

#### AUTHOR BIOGRAPHIES

**MICHAEL E. KUHL** is an Associate Professor in the Industrial and Systems Engineering Department at Rochester Institute of Technology. He has a Ph.D. in Industrial Engineering from North Carolina State University (1997). His research interests include simulation modeling and analysis with application to input modeling, healthcare, project management, and semiconductor manufacturing. He served as Proceedings Editor for the 2005 Winter Simulation Conference. He is currently president of the INFORMS Simulation Society, and a member of IIE and ASEE. His e-mail address is <[Michael.Kuhl@rit.edu](mailto:Michael.Kuhl@rit.edu)> and his web address is <[people.rit.edu/mekeie](http://people.rit.edu/mekeie)>.

**RADHAMÉS A. TOLENTINO-PEÑA** is a Master of Science candidate in Industrial Engineering in the Industrial and Systems Engineering Department at Rochester Institute of Technology. His research interests include the application of simulation and operations research methods to the areas of project management and logistics. He is a member of IIE, APICS, and SHPE. His e-mail address is <[radhames.tolentino@mail.rit.edu](mailto:radhames.tolentino@mail.rit.edu)>.