Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

TIME SEGMENTATION PARALLEL SIMULATION OF TANDEM QUEUES WITH MANUFACTURING BLOCKING

Mehdi Hoseyni-Nasab

DIMACS
Rutgers University
Piscataway, New Jersey 08854, U.S.A.

Sigrún Andradóttir

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332, U.S.A.

ABSTRACT

This paper studies the application of the time segmentation parallel simulation approach to efficient simulation of simple manufacturing queueing systems, namely systems of queues in tandem with manufacturing blocking at each station. We discuss the difficulties in the way of applying the time segmentation method to this class of manufacturing systems and demonstrate that the time segmentation method can efficiently provide very close approximate simulation results.

1 INTRODUCTION

Discrete event simulation has proven to be an effective tool in the study of complex real world systems. However, because of the complexity of many discrete event systems, obtaining reliable and accurate simulation results for such systems may require a large amount of computational resources. The goal of a parallel simulation method is to improve the efficiency of simulation experiments by distributing the computational load of these experiments among multiple processors. In this paper we discuss a parallel simulation method, namely the time segmentation approach, that can be used to simulate long sample paths of several classes of discrete event systems. Furthermore, we investigate the application of the time segmentation approach to the simulation of a class of simple manufacturing queueing systems.

In recent years, several parallel simulation methods have been proposed and extensively studied. Most of these parallel simulation methods can be classified into two main categories: *distributed simulation methods* and *time parallel simulation methods*. In distributed simulation, the system under study is partitioned into a number of subsystems and a separate processor is assigned to simulate the activities of each subsystem. The simulation algorithm establishes a communication mechanism between different processors to guarantee that valid sample paths are being simulated. In time parallel simulation, the goal is to distribute the computational load of simulating a long sample path between multiple processors. This task is accomplished by partitioning the time horizon of the sample path into smaller segments and assigning a processor to each segment. Processors that are assigned to simulate different segments of a sample path need to communicate to guarantee that a valid sample path is being simulated.

It is clear that the choice of an appropriate parallel simulation method for a specific system is directly related to the characteristics of the system under study. Comprehensive surveys and comparative analyses of recent developments in parallel simulation can be found in Fujimoto and Nicol (1994) and Chapter 2 of Hoseyni-Nasab (1998), among others.

The time segmentation approach is a time parallel simulation method that was originally proposed by Andradóttir and Ott (1995). This method can be applied to efficiently simulate long sample paths of many different models of discrete event systems. Applications of this method to efficient simulation of queueing models of communication networks have been studied by Andradóttir and Ott (1995) and Hoseyni-Nasab and Andradóttir (1996, 1997). In this paper, we describe the time segmentation method and study its application to parallel simulation of a tandem network of queues with manufacturing blocking at each station (i.e., a transfer line). We show that despite the fact that our model does not fully satisfy all the conditions required for valid application of the time segmentation method, the time segmentation method can be used to produce approximate simulation results efficiently.

The outline of this paper is as follows: In Section 2 we present the time segmentation method and discuss the conditions that are required for valid application of this method. In Section 3 we propose a model for transfer lines, present an algorithm for simulating sample paths of this model, and discuss the difficulties in the way of applying the

time segmentation method to this model. Finally, Section 4 contains the results of a number of simulation experiments that indicate that the time segmentation method can be used to simulate long sample paths of transfer lines and produce exact or approximate simulation results efficiently.

2 THE TIME SEGMENTATION APPROACH

In this section we present the time segmentation method and describe how this method can be used to efficiently simulate long sample paths of discrete event systems. We will also discuss factors that affect the applicability and efficiency of the method. For more details on the time segmentation approach, the reader is referred to Andradóttir and Ott (1995).

Suppose that we would like to generate a sample path of a discrete event system S on the interval $[0, T]$. For all $t \geq 0$ and all sample paths A of the system S, let $N_A(t)$ denote the state of the system at time t in sample path A. Suppose that it is possible to generate multiple valid sample paths of the system using a common sequence of potential events. Furthermore, suppose that there exist sample paths l and u, generated using a common sequence of potential events, such that by the time sample paths l and u *couple* (i.e., when $N_l(t) = N_u(t)$), all other sample paths that are being generated with the same sequence of potential events will have also coupled with sample paths l and u. Moreover, suppose that after the coupling of the sample paths l and u, the state of the system in all sample paths that are generated with a common sequence of potential events is identical. We refer to sample paths l and u as the bounding sample paths.

Now, suppose that it is possible to conduct the simulation of multiple sample paths of the system using a common sequence of potential events in such a way that if $N_A(0) \leq N_B(0)$, where A and B are two arbitrary sample paths of the system, then $N_A(t) \leq N_B(t)$, for all $t > 0$ (we refer to this property as the *monotonicity condition*). Then, the existence of bounding sample paths l and u is guaranteed if we assume that there exist initial states $N_l(0)$ and $N_u(0)$ such that $N_l(0) \leq N_A(0) \leq N_u(0)$, for any sample path A. Clearly, these two properties imply that the states of the system in sample paths l and u bound the state of the system in all sample paths simulated with the same sequence of potential events, at all times, from below and above, respectively. This guarantees that by the time sample paths l and u couple, all other sample paths that are being generated with the same sequence of potential events will have also coupled with sample paths l and u.

Assuming that it is possible to simulate multiple sample paths of the system using a common sequence of potential events and that bounding sample paths l and u exist, we proceed as follows: We partition the time horizon of the simulation experiment into P equal segments, $[0, T/P], \ldots, [(P-1)T/P, T]$, where P is the number of available processors. Suppose that interval i refers to the interval $[(i-1)T/P, iT/P]$, for $1 \leq i \leq P$. Each processor is assigned to one segment of the sample path and is responsible for generating the sample path over a time period of length T/P corresponding to that segment. In order for a processor to initiate the simulation of the sample path on its segment, the processor needs to know the state of the system at the end of the sample path on the previous segment.

We start the simulation of the sample path on the interval $[0, T/P]$ from the true initial state of the sample path. On the other subintervals, the initial states of the sample path are not known at the beginning of the simulation period (note that all processors start processing at the same time). The processors assigned to segments $2, \ldots, P$ of the sample path start simulating bounding sample paths l and u using a common sequence of potential events for the sample paths of each interval. For $i = 2, \ldots, P$, let

$$T_c^i = \inf \left\{ t \in \left[\frac{(i-1)T}{P}, \frac{iT}{P} \right] : N_l(t) = N_u(t) \right\}$$

be the coupling time of sample paths l and u generated by processor i. If $T_c^i < iT/P$, then $N_l(t) = N_u(t)$, for all $t \in [T_c^i, iT/P]$. This means that after the bounding sample paths of the system on subinterval i couple, the processor corresponding to that interval can start collecting data (because the state of the system no longer depends on the initial state of the sample path). Therefore, the information collected on the interval $[T_c^i, iT/P]$ is valid data for the true sample path.

If sample paths l and u on interval $i-1$ couple prior to the end of that interval, then the real initial state of the sample path on interval i is the same as the final state of the coupled bounding sample paths on interval $i-1$. Therefore, to complete the sample path on interval i, we start the simulation from the true initial state (given by the final state of the coupled bounding sample paths on interval $i-1$) and simulate the sample path on the interval $[(i-1)T/P, \min\{T_c^i, iT/P\}]$ using the same sequence of potential events as the bounding sample paths of the interval.

On the other hand, if sample paths l and u on interval $i-1$ are not coupled prior to the end of that interval, then we again simulate two (bounding) sample paths l' and u' on interval i starting in the final states $N_l((i-1)T/P)$ and $N_u((i-1)T/P)$ of sample paths l and u on interval $i-1$ and using the same sequence of potential events as the one used to generate sample paths l and u on interval i. By repeating this process, possibly several times, and

combining the data collected on all the subintervals, we can generate a complete sample path of the system on the interval $[0, T]$.

In the above procedure, if all the bounding sample paths l and u couple prior to the end of their corresponding intervals, then the true initial state of the sample path on each subinterval will be known upon completing the simulation of the coupled bounding sample paths on the previous interval. Also, it is clear that if the coupling times are small, then a larger portion of the true sample path will be generated during the simulation of the bounding sample paths l and u of the subintervals. This suggests that the amount of time spent on generating a sample path of S on the interval $[0, T]$ using the time segmentation approach is an increasing function of the magnitude of the coupling times of the bounding sample paths of the system. Therefore, the magnitude of the coupling times can be used to measure the efficiency of the time segmentation approach.

3 SIMULATION MODEL AND ALGORITHM

In this section we present a model for transfer lines and present an algorithm that can be used to simulate multiple sample paths of this model using a common sequence of potential events. Furthermore, we discuss the extent to which this model satisfies the conditions required for the applicability of the time segmentation method.

Suppose that S is a tandem network of $n > 1$ single server queueing stations with an infinite buffer in front of station 1 and finite buffers in front of the other stations in the system. Let B_i be the size of the buffer in front of station i, for $i = 2, \ldots, n$. Moreover, suppose that we always have an unlimited number of jobs in the buffer in front of station 1 (i.e., as soon as we have a departure from station 1, the server at station 1 starts working on a new job). Finally, suppose that jobs leave the system after completing service at station n (i.e., there is always room for departing jobs after station n), and that stations 1 through $n-1$ operate under the following blocking mechanism:

Manufacturing Blocking: Upon completing service at station i, $i = 1, \ldots, n-1$, a job attempts to move to station $i+1$. If there is no room at station $i+1$ for this job (either in the buffer or in the service area), the job is forced to wait at station i until there is room for it at station $i+1$. The server of station i cannot operate during this waiting time (i.e., the server is blocked) and it will resume operating as soon as the departing job moves to station $i+1$, provided that there is a job waiting to enter service at station i.

We refer to a system S that satisfies all the above conditions as a transfer line.

The manufacturing blocking mechanism creates difficulties in the way of modeling and simulating the system S by time segmentation. In particular, knowing the number of jobs at the stations of a transfer line does not completely determine the state of the system at that time. Furthermore, the occurrence of an event at one station of a manufacturing queueing system can potentially affect the state of many other stations. Therefore, in order to capture the dynamics of the system properly, one needs to develop a model which contains more information than merely the number of jobs at the stations in the system.

We now describe one of the standard models that has been studied in the literature for analyzing transfer lines (see Chapter 6 in Buzacott and Shantikumar, 1993).

Transfer Line Model: Suppose that S is a transfer line (as defined at the beginning of this section) and, for all $t \geq 0$, let $K(t) = (K_1(t), \ldots, K_{n-1}(t))$, where $K_i(t)$ is the number of jobs that have been served by the server at station i but have not yet completed service at station $i+1$ at time t, for $i = 1, \ldots, n-1$.

To determine the state of a station completely, we need to know both the number of jobs at the station and the state of the server of the station (i.e, blocked or not). It can be shown that when the service times of all servers are exponentially distributed, then $K(t)$ completely and uniquely determines the state of the system at time t. Note that the state of the last station (i.e., the number of jobs and the state of the server of station n) at time t can be completely determined by the value of $K_{n-1}(t)$ and the state of any station i, $i = 1, \ldots, n-1$, at time t can be uniquely determined using the state of station $i+1$ at time t and $K_i(t)$. Therefore, we can inductively determine the states of all stations at time t using the vector $K(t)$.

Suppose that the service times of station i are independent and exponentially distributed and that μ_i denotes the rate of the service times at station i, for $i = 1, \ldots, n$. Then, the stochastic process $\{K(t)\}$ is clearly a continuous time Markov chain. The following algorithm explains how M sample paths of the Markov chain $\{K(t)\}$ can be generated simultaneously using a common sequence of potential events. More details are given in Hoseyni-Nasab (1998). In Algorithm 3.1, T_i^s and T denote the time of the next potential service completion at station i and the elapsed simulation time, respectively. Furthermore, we assume that we can generate independent sequences $\{U_i^s(k)\}$, $i = 1, \ldots, n$, of independent and uniformly distributed random variables on the interval $[0, 1]$. The sequence $\{U_i^s(k)\}$ will be used to generate the sequence of potential service completion times at station i, for $i = 1, \ldots, n$. Finally, for $i = 1, \ldots, n$, let n_i^s be the

number of uniform random numbers from the sequence $\{U_i^s(k)\}$ that have been used to generate observations of the potential service times at station i.

Algorithm 3.1 (Simulation of Markovian Transfer Lines)

Step 0: *Initialization.*
For $m = 1, \ldots, M$, select a feasible starting state $(K_1^m, \ldots, K_{n-1}^m)$ for the system S. For $i = 1, \ldots, n$, generate $U_i^s(1)$ and let $T_i^s = -\log(U_i^s(1))/\mu_i$. Let $T = 0$ and $n_i^s = 1$, for $i = 1, \ldots, n$. Go to Step 1.

Step 1: *Determination of the next event.*
Let $T^* = \min_{i=1,\ldots,n}\{T_i^s\}$. If $T^* = T_i^s$, where $i \in \{1, \ldots, n\}$, then let $e^* = i$. If $1 < e^* < n$, then go to Step 2. If $e^* = 1$, then go to Step 3. If $e^* = n$, then go to Step 4.

Step 2: *Service completion at station e^* with $1 < e^* < n$.*
Let $T = T^*$. For $m = 1, \ldots, M$, if $K_{e^*-1}^m > 0$ and station e^* is not blocked, then let $K_{e^*}^m = K_{e^*}^m + 1$ and $K_{e^*-1}^m = K_{e^*-1}^m - 1$. Let $n_{e^*}^s = n_{e^*}^s + 1$ and $T_{e^*}^s = T^* - \log(U_{e^*}^s(n_{e^*}^s))/\mu_{e^*}$. Go to Step 1.

Step 3: *Service completion at station $e^* = 1$.*
Let $T = T^*$. For $m = 1, \ldots, M$, if station e^* is not blocked, then let $K_{e^*}^m = K_{e^*}^m + 1$. Let $n_{e^*}^s = n_{e^*}^s + 1$ and $T_{e^*}^s = T^* - \log(U_{e^*}^s(n_{e^*}^s))/\mu_{e^*}$. Go to Step 1.

Step 4: *Service completion at station $e^* = n$.*
Let $T = T^*$. For $m = 1, \ldots, M$, if $K_{e^*-1}^m > 0$, then let $K_{e^*-1}^m = K_{e^*-1}^m - 1$. Let $n_{e^*}^s = n_{e^*}^s + 1$ and $T_{e^*}^s = T^* - \log(U_{e^*}^s(n_{e^*}^s))/\mu_{e^*}$. Go to Step 1.

Algorithm 3.1 explains how multiple sample paths of the transfer line S can be generated simultaneously using a common sequence of potential events. In order to apply the time segmentation approach to our transfer line model (using Algorithm 3.1), we need to identify the bounding sample paths. One difficulty with applying the time segmentation method to the above model is that there is not a unique sample path that would initially bound all other sample paths from above. For example, consider two sample paths K^1 and K^2 that start from the initial states $K^1(0) = (B_2 + 1, B_3 + 2, B_4, B_5 + 2, B_6, \ldots, B_*^1)$ and $K^2(0) = (B_2 + 2, B_3, B_4 + 2, B_5, \ldots, B_*^2)$, respectively, where $B_*^j = B_n + 1$ or $B_n + 2$, if $K_{n-2}^j = B_{n-1} + 2$ or B_{n-1}, respectively, for $j = 1, 2$. It is clear that neither one of these sample paths bounds the other one from above. However, the initial states of the sample paths K^1 and K^2 together bound all the other initial states from above. This suggests that we can use multiple upper bounding sample paths in our time segmentation simulation. We will investigate the effects of using multiple upper bounding sample paths in the simulation results in Section 4.

Another difficulty with applying the time segmentation method to our transfer line model is lack of monotonicity. More specifically, if we are simulating sample paths A and B using Algorithm 3.1 and sample path A is initially below sample path B, then we are not guaranteed that sample path A will stay below sample path B throughout the simulation process (i.e., sample path A may eventually cross sample path B). This implies that if we initially bound all sample paths between a number of bounding sample paths, then the coupling of the bounding sample paths may not necessarily guarantee the coupling of the other sample paths. The following proposition exactly determines the situations where two sample paths could cross each other without getting coupled. This result is proved in Hoseyni-Nasab (1998).

Proposition 3.1 *Suppose that S is a transfer line with n stations and independent and exponentially distributed service times at each station. Let A and B be two sample paths of the system and for all $t \geq 0$, let $(a_1(t), \ldots, a_{n-1}(t))$ and $(b_1(t), \ldots, b_{n-1}(t))$ denote the states of the system at time t in sample paths A and B, respectively. Let sample paths A and B be simulated simultaneously by Algorithm 3.1 and suppose $a_i(0) \leq b_i(0)$, for $i = 1, \ldots, n-1$. Furthermore suppose that the first scheduled event is a service completion at station j that is scheduled to occur at time t_0. Then we have $a_i(t_0) \leq b_i(t_0)$, for $i = 1, \ldots, n-1$ if and only if the following condition is* **not** *satisfied (i.e., if and only if at least one of the logical statements of the following condition does not hold):*

Crossing Condition: *$j < n$, $a_j(0) = b_j(0) = B_{j+1} + 1$, station j in sample path B is blocked at time $t = 0$ and station j in sample path A is not blocked at time $t = 0$.*

Proposition 3.1 describes the only situation in which sample paths of the transfer line S can cross each other. Since a number of extreme conditions need to be satisfied simultaneously in order for crossing to take place, we expect that sample paths generated in parallel by Algorithm 3.1 will rarely cross in practice (the numerical results given in Section 4 support this conclusion). Moreover, we can modify the time segmentation approach in such a way that crossing is nonexistent for our transfer line model. A thorough study of this issue is a subject of our current research.

4 NUMERICAL RESULTS

In this section we present a selection of numerical results aiming at studying the effectiveness of the time

Table 1: Effects of Multiple Upper Bounding Sample Paths in Time Segmentation Simulation of Markovian Transfer Lines when $n = 5$

Buffer Size, B	Coupling Times of Upper Bounding Sample Paths (95% C.I.)	Coupling Times of All Bounding Sample Paths (95% C.I.)
5	12.79 ($\pm$ 3.19)	60.00 ($\pm$ 3.87)
10	21.85 ($\pm$ 6.37)	154.08 ($\pm$ 12.11)
20	30.97 ($\pm$ 11.09)	506.63 ($\pm$ 38.40)

Table 2: Effects of Multiple Upper Bounding Sample Paths in Time Segmentation Simulation of Markovian Transfer Lines when $n = 10$

Buffer Size, B	Coupling Times of Upper Bounding Sample Paths (95% C.I.)	Coupling Times of All Bounding Sample Paths (95% C.I.)
5	28.12 ($\pm$ 6.48)	181.17 ($\pm$ 14.33)
10	46.01 ($\pm$ 11.69)	466.14 ($\pm$ 38.34)
20	58.20 ($\pm$ 17.27)	1,405.55 ($\pm$ 102.36)

segmentation method in efficient simulation of transfer lines. More specifically, we present results that investigate the effects of using multiple upper bounding sample paths. Furthermore, we study the efficiency of the approach by providing estimates for the expected coupling times of the systems under study.

In all of our experiments we have simulated a transfer line with n queueing stations and a common buffer capacity B at each station i, for $i = 2, \ldots, n$. Furthermore, in all of our experiments we use the service rate $\mu_i = 1$, for $i = 1, \ldots, n$. The confidence intervals (c.i.) are obtained by simulating 100 independent replications of the sample paths of the systems under study. In all simulations we have used Algorithm 3.1 for generating multiple sample paths of the systems under study using a common sequence of potential events.

Our first experiment aims at better understanding the effects of using multiple upper bounding sample paths on the simulation results. The simulations are conducted for $n \in \{5, 10\}$ and $B \in \{5, 10, 20\}$. We have used Algorithm 3.1 with $M = 5$ to simulate sample paths 1, 2, 3, 4, and 5, starting from the initial states $(0, \ldots, 0)$, $(B-1, \ldots, B-1)$, $(B, \ldots, B)$, $(B+1, B+2, B, B+2, B, \ldots, B_*^1)$ and $(B+2, B, B+2, B, \ldots, B_*^2)$, respectively, where $B_*^j = B+1$ or $B+2$, if $K_{n-2}^j = B+2$ or B, respectively, for $j = 1, 2$. Clearly sample path 1 is the lower bounding sample path and sample paths 4 and 5 initially bound all other sample paths from above. The results are presented in Tables 1 and 2.

In Tables 1 and 2, the second column presents confidence intervals for the expected coupling time of the upper bounding sample paths and the third column presents confidence intervals for the expected coupling time of all bounding sample paths. In all cases, the middle sample paths (i.e., sample paths 2 and 3) have coupled with the bounding sample paths prior to the coupling of all the bounding sample paths. Considering the crossing condition, we selected the initial states of the middle sample paths near the upper boundary of the state space in order to increase the likelihood of crossings in the simulation process. The results indicate that the upper bounding sample paths couple early in the simulation process. Furthermore, the coupling of the bounding sample paths appears to guarantee the coupling of the middle sample paths with high probability. This suggests that we can use the time segmentation approach with these bounding sample paths and expect to obtain simulation results that are very close approximations for the performance measures of interest.

Table 3: Dependence of the Expected Coupling Times of Transfer Lines on the Buffer Sizes and the Number of Stations in the System

Number of Stations, n	Buffer Size, B	Coupling Times (95% C.I.)
5	5	60.00 ($\pm$ 3.87)
5	10	154.08 ($\pm$ 12.11)
5	20	506.63 ($\pm$ 38.40)
5	40	1,711.80 ($\pm$ 137.19)
10	5	181.17 ($\pm$ 14.33)
10	10	466.14 ($\pm$ 38.34)
10	20	1,405.55 ($\pm$ 102.36)
10	40	4,999.01 ($\pm$ 345.22)
20	5	491.94 ($\pm$ 33.45)
20	10	1,313.44 ($\pm$ 82.20)
20	20	3,903.43 ($\pm$ 266.25)
20	40	12,560.84 ($\pm$ 1,016.57)

Our second set of simulation results investigates the efficiency of the time segmentation approach by studying the behavior of the expected coupling times. The simulations are conducted for $n \in \{5, 10, 20\}$ and $B \in \{5, 10, 20, 40\}$. We have used Algorithm 3.1 with $M = 3$ to simulate bounding sample paths 1, 4, and 5 (note that our previous results suggest that all sample paths that are simulated with a common sequence of potential events are likely to couple no later than the coupling time of these three bounding sample paths). The results are presented in Table 3.

As we see in Table 3, the expected coupling times of the system do not appear to grow linearly with respect to the number of stations or the buffer capacities of the system. However, the expected coupling times appear to grow no faster than quadratically with respect to both n and B. Depending on the length of the sample path that needs to be generated, these numerical results can be used as a guideline to determine what choices of the length of the segments of the time segmentation simulation have the property that the coupling of the bounding sample paths prior to the end of the segments is very probable. Further studies of the dependence of the coupling times of transfer lines on the parameters of the system is a subject of our future research.

ACKNOWLEDGMENT

This research was supported in part by the National Science Foundation under grant DMI–9523111 and in part by DIMACS.

REFERENCES

Andradóttir, S., and T. J. Ott. 1995. Time segmentation parallel simulation of networks of queues with loss or communication blocking. *ACM Transactions on Modeling and Computer Simulation* 5(4):269–305.

Buzacott, J., and J. Shantikumar. 1993. *Stochastic Models of Manufacturing Systems*. Prentice Hall.

Fujimoto, R., and D. Nicol. 1994. Parallel simulation today. *Annals of Operations Research* 53:249–285.

Hoseyni-Nasab, M. 1998. *Parallel simulation of queueing networks by time segmentation*. Ph.D. Dissertation, Department of Industrial Engineering, University of Wisconsin–Madison.

Hoseyni-Nasab, M., and S. Andradóttir. 1996. Parallel simulation by time segmentation: Methodology and applications. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. M. Morrice, D. B. Brunner, and J. J. Swain, 376–381. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Hoseyni-Nasab, M., and S. Andradóttir. 1997. Efficiency of time segmentation parallel simulation of queueing networks as a function of the size of the network. In *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 181–186. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

AUTHOR BIOGRAPHIES

MEHDI HOSEYNI-NASAB is a Postdoctoral Fellow at DIMACS, Rutgers University. His research interests include parallel simulation, applied probability, and analysis of communication systems.

SIGRÚN ANDRADÓTTIR is an Associate Professor in the School of Industrial and Systems Engineering at the Georgia Institute of Technology. Previously, she was an Assistant Professor and later an Associate Professor in the Departments of Industrial Engineering, Mathematics, and Computer Sciences at the University of Wisconsin – Madison. She received a B.S. in Mathematics from the University of Iceland in 1986, an M.S. in Statistics from Stanford University in 1989, and a Ph.D. in Operations Research from Stanford University in 1990. Her research interests include stochastic optimization, simulation, and stochastic processes. She presently serves as Associate Editor for *IIE Transactions*, *Stochastic Models*, and the *ACM Transactions on Modeling and Computer Simulation*. She is a member of INFORMS and served as Editor of the *Proceedings of the 1997 Winter Simulation Conference*.