Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

PARALLEL SIMULATION BY TIME SEGMENTATION: METHODOLOGY AND APPLICATIONS

Mehdi Hoseyni-Nasab

Department of Industrial Engineering
University of Wisconsin–Madison
1513 University Avenue
Madison, WI 53706, U.S.A.

Sigrún Andradóttir

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

ABSTRACT

In this paper we present a time parallel simulation approach and discuss conditions under which this approach is applicable. Our approach involves distributing the available processors among segments of the time horizon of the simulation. We show that under certain conditions, sample paths of the system generated by a common sequence of potential events will couple (i.e., become identical) with probability one. This property will be exploited to efficiently combine the information collected on different segments of the sample path and generate a complete valid sample path. We show that the expected coupling time of the system (i.e., the amount of time required for all the sample paths of the system to couple) is essential to the efficiency of the approach. We apply our parallel simulation approach to a class of Markovian queueing networks and investigate the efficiency of the method by providing bounds and estimates for the expected coupling times.

1 INTRODUCTION

The need for efficient simulation of discrete event systems has created a rapidly growing demand for parallel simulation methods that can effectively distribute the computational load of a simulation experiment among multiple processing units. Several parallel simulation methods have been studied and implemented in recent years. Depending on the strategy these methods choose to make concurrent use of the available processors, most parallel simulation methods fall into one of the following categories: *distributed simulation* methods, *multiple replications* methods, and *time parallel simulation* methods.

In distributed simulation methods the system under study is partitioned into subsystems and the available processors are appropriately distributed among the subsystems. Processors simulate the subsystems and communicate with each other to guarantee that the generated sample paths are valid (see Fujimoto (1993) for more details on distributed simulation).

In multiple replications methods, every processor generates sample paths of the system independently of the other processors. Then, the results obtained by different processors are averaged to provide estimators for the performance measures of interest (see Heidelberger (1988) for more details on the multiple replications approach).

A variety of time parallel simulation approaches have been suggested in recent years for parallel simulation of different classes of discrete event systems. Greenberg, Lubachevsky, and Mitrani (1990) and Chen and Serfozo (1995) take advantage of representations of the dynamics of the systems under study in terms of recurrence relations to achieve parallelism. Heidelberger and Stone (1990) propose a trace partitioning approach to perform parallel trace driven simulation of a cache, and Lin and Lazowska (1991) suggest a parallel simulation method applicable to a class of partially regenerative systems.

Andradóttir and Ott (1995) have recently proposed a time parallel simulation approach, namely the time segmentation approach, that can be applied to a variety of queueing networks. They have applied this method to Markovian queueing systems that consist of either loss stations or communication blocking stations. They have investigated the efficiency of the approach by providing bounds and estimates for the expected coupling times of the systems under study. The results of Andradóttir and Ott (1995) have been extended and strengthened by Hoseyni-Nasab and Andradóttir (1996a,b). They have shown that the time segmentation method is effectively applicable to different classes of queueing networks.

In this paper we describe how the time segmentation method can be applied to simulate discrete event systems whose sample paths satisfy certain constructibility conditions on parallel processors. We also present analytical and numerical results indicating the validity, applicability, and efficiency of the ap-

proach in the context of parallel simulation of a family of Markovian queueing networks containing loss and communication blocking stations.

The outline of the paper is as follows: In Section 2 we present the time segmentation parallel simulation method. In Section 3 we show that the time segmentation method is applicable to a class of Markovian queueing networks and investigate the efficiency of the approach by providing bounds for the expected coupling times. In Section 4 we further study the efficiency of the approach by estimating the expected coupling times for a number of queueing systems and providing numerical results directed towards understanding the dependence of the expected coupling times on the parameters of the system. We conclude the paper by a few remarks in Section 5.

2 THE PARALLEL SIMULATION APPROACH

In this section we describe the time segmentation method for parallel simulation of discrete event systems and discuss conditions under which this approach is applicable.

Suppose S is a discrete event system and suppose that we would like to generate long sample paths of a simulation model of S. Let $N(t)$ be a function of time t that uniquely determines the state of the system at time t. Now suppose we would like to use P processors concurrently to generate a sample path of S. Let T be the length of the required sample path, partition the interval $[0, T]$ into P subintervals $[t_0, t_1), [t_1, t_2), ..., [t_{P-1}, t_P]$, and assign one processor to each subinterval. Furthermore, suppose it is possible to generate valid sample paths of the system in parallel using a common sequence of potential events (i.e., given a sequence of events, a scheduled event in the sequence will be executed in a sample path if the event is feasible for the sample path). We start the simulation of the sample path on the first subinterval from the true initial state of the sample path. In other subintervals, the corresponding processors start simulating two sample paths, sample paths l and u (call them the bounding sample paths), using a common sequence of potential events. Let $N_l(t)$ and $N_u(t)$ denote the state of the system at time t in sample paths l and u, respectively. Suppose sample paths l and u can be chosen and the simulation can be conducted in such ways that by the time sample paths l and u couple (i.e., the first time T_c when $N_l(T_c) = N_u(T_c)$), any other sample path that has been constructed with the same sequence of potential events as l and u will have coupled with sample paths l and u. Finally, suppose that if two sample paths that have been constructed with the same sequence of potential events couple, then they will stay coupled for the remainder of the simulation period. For $i = 2, ..., P$, if there exists T_c^i such that $T_c^i = \inf\{t \in [t_{i-1}, t_i) : N_l(t) = N_u(t)\}$ (T_c^i is the coupling time of sample paths l and u on the interval $[t_{i-1}, t_i)$), then the above assumptions clearly imply that $N_l(t) = N(t) = N_u(t)$, for all $t \in [T_c^i, t_i)$. Therefore the information collected from either one of the bounding sample paths on the interval $[T_c^i, t_i)$ is valid data for the true sample path as well. Also, note that if sample paths l and u in the interval $[t_{i-2}, t_{i-1})$ are coupled prior to the end of the subinterval, then the real initial state of the sample path on the interval $[t_{i-1}, t_i)$ is the same as the terminal state of the coupled sample paths on the interval $[t_{i-2}, t_{i-1})$. Therefore, we can redo the simulation of the sample path on the interval $[t_{i-1}, T_c^i)$ with the true initial state (using the same sequence of potential events as the bounding sample paths of the interval) and collect the required data. If sample paths l and u in interval $[t_{i-2}, t_{i-1})$ are not coupled prior to time t_{i-1}, then we repeat what we did to sample paths l and u by initiating two sample paths in the interval $[t_{i-1}, t_i)$ starting at the initial states $N_l(t_{i-1})$ and $N_u(t_{i-1})$, which are the terminal states of sample paths l and u on the interval $[t_{i-2}, t_{i-1})$, respectively. By going through the above process (possibly repeatedly) and putting together the data collected on all the subintervals, we will finally generate a complete observation of the true sample path on the interval $[0, T]$. Note that the simulation is conducted in such a way that the sequence of potential events used for generating a complete observation of a true sample path is the same as the sequence obtained by putting together the sequences of potential events on the consecutive subintervals of the simulation.

As we mentioned before, the magnitudes of the coupling times are essential to the efficiency of the method. Note that if the bounding sample paths of a subinterval couple, then the part of the sample path generated after the coupling time on that subinterval is absolutely valid. Therefore, in general, the shorter the coupling times of the bounding paths the more efficiently we can construct the true sample path. In the best case scenario (i.e., when the bounding sample paths of all subintervals couple on the first run of the simulation), the total simulation time will be of order $\frac{2T}{P}$ or less. In the worst case scenario (i.e., when no bounding sample paths ever couple), the total simulation time will be of order T; that is the same order of magnitude as the simulation time of a single processor simulation of the system.

In the next section we will show that the time segmentation approach is applicable to a large class of queueing networks and investigate the efficiency of the method when applied to these systems.

3 APPLICABILITY AND EFFICIENCY OF THE APPROACH

The parallel simulation approach described in Section 2 is applicable to a variety of discrete event systems. In this section we show that the time segmentation approach can be applied to simulate a family of queueing networks with loss and communication blocking stations. We will also investigate the efficiency of the approach by studying the magnitude of the expected coupling times of the systems under study. Before stating our results we need to define the system under study and introduce some notation.

Suppose S is a network of n queueing stations. For $i = 1, ..., n$, let s_i and B_i denote the number of servers and the buffer capacity at station i, respectively, and let $C_i = B_i + s_i$. Suppose the service times at all servers of station i are independent and exponentially distributed with rate μ_i, $i = 1, ..., n$, and let jobs arrive from the outside to stations $1, ..., n$ according to independent Poisson processes with rates $\lambda_1, ..., \lambda_n$, respectively. For $i = 1, ..., n$, suppose p_{ij} is the probability that a job will attempt to join the queue at station j, immediately after a service completion at station i, $j = 1, ..., n$, and let $p_{i,n+1}$ denote the probability that a job will depart from the system upon a service completion at station i. Furthermore, suppose every station $i \in \{1, ..., n\}$ operates under one of the following two mechanisms:

1. Station i is a *loss* station if arriving jobs (from other stations in the system or from the outside) that find a full buffer at station i leave the system immediately.

2. Station i is a *communication blocking* station if arriving jobs from the outside that find a full buffer at station i leave the system immediately, and arriving jobs from other stations in the system that find a full buffer at station i get rerouted (independently of the past), after undergoing another service completion in the station of their most recent service completion.

The definition of communication blocking given above is different from the standard definition used in the literature. In the standard communication blocking mechanism, jobs will not start receiving service at a station unless there is room for them in the station that they are visiting next. However, it is not clear how the standard communication blocking mechanism works when we have queueing stations with multiple servers, multiple sources of arrivals, or multiple destinations for the departing customers. When the service times are exponentially distributed, the two definitions of communication blocking are equivalent if every station has only one server, one source of arrivals, and one destination for the departing customers.

For all $t \geq 0$, let $N(t) = (N_1(t), ..., N_n(t))$, where $N_i(t)$ is the number of customers at station i at time t, $i = 1, ..., n$. It is clear that $\{N(t)\}$ is a continuous-time Markov chain. Our goal is to show that the time segmentation approach can be used to simulate long sample paths of the Markov chain $\{N(t)\}$ efficiently.

For $i = 1, ..., n$, let T_i^a, T_i^s and T denote the time of the next arrival at station i, the time of the next potential service completion at station i, and the elapsed simulation time. Suppose we can generate four independent sequences $\{U_i^a(k)\}$, $\{U_i^s(k)\}$, $\{U_i^d(k)\}$, and $\{U_i^r(k)\}$ of independent $U[0,1]$ distributed random variables, for $i = 1, ..., n$. These sequences will be used to generate the sequence of interarrival times at station i, the sequence of potential service completion times at station i, the sequence of numerical marks that will be used to determine whether a potential service completion is an actual service completion at station i, and the sequence of destinations of the jobs that are departing station i, respectively. Finally, for $i = 1, ..., n$, let n_i^a, n_i^s, n_i^d, and n_i^r be the number of uniform random numbers from the sequences $\{U_i^a(k)\}$, $\{U_i^s(k)\}$, $\{U_i^d(k)\}$, and $\{U_i^r(k)\}$ that have been used to generate observations of the corresponding random variables, respectively.

We are now ready to state our algorithm. The empty sum in the following algorithm is defined to be zero. Algorithm 3.1 is a combination of Algorithms 2.1 and 3.1 in Andradóttir and Ott (1995).

Algorithm 3.1

Step 0: *Initialization.*
For $m = 1, ..., M$, select initial states N_i^m such that $0 \leq N_i^m \leq C_i$, $i = 1, ..., n$. For $i = 1, ..., n$, generate $U_i^a(1)$ and $U_i^s(1)$, and let $T_i^s = -\frac{\log(U_i^s(1))}{s_i \mu_i}$ and $T_i^a = -\frac{\log(U_i^a(1))}{\lambda_i}$. Let $T = 0$, $n_i^a = 1$, $n_i^s = 1$, $n_i^d = 0$ and $n_i^r = 0$, $i = 1, ..., n$. Go to Step 1.

Step 1: *Identifying the next event.*
Let $T^* = \min_{i=1,...,n} \{T_i^a, T_i^s\}$. If $T^* = T_i^a$, where $i \in \{1, ..., n\}$, then let $e^* = i$ and go to Step 2. Otherwise, if $T^* = T_i^s$, where $i \in \{1, ..., n\}$, then let $e^* = i$ and if station i is a loss or communication blocking station, go to Step 3 or 4, respectively.

Step 2: *Arrival at station e^*.*
Let $T = T^*$. For $m = 1, ..., M$, if $N_{e^*}^m < C_{e^*}$, then let $N_{e^*}^m = N_{e^*}^m + 1$. Let $n_{e^*}^a = n_{e^*}^a + 1$ and $T_{e^*}^a = T^* - \frac{\log(U_{e^*}^a(n_{e^*}^a))}{\lambda_{e^*}}$. Go to Step 1.

Step 3: *Departure from a loss station e^*.*
Let $T = T^*$ and $n^d_{e^*\bullet} = n^d_{e^*\bullet} + 1$. If $U^d_{e^*\bullet}(n^d_{e^*\bullet}) \leq \max_{m=1,...,M}\{N^m_{e^*\bullet}\}/s_{e^*}$, then let $n^r_{e^*\bullet} = n^r_{e^*\bullet} + 1$ and let $j \in \{1,...,n+1\}$ be such that $\sum_{k=1}^{j-1} p_{e^*,k} < U^r_{e^*\bullet}(n^r_{e^*\bullet}) \leq \sum_{k=1}^{j} p_{e^*,k}$ (if $U^r_{e^*\bullet}(n^r_{e^*\bullet}) = 0$, then let $j = 1$). Otherwise, let $j = 0$.

- *Departure from the system.*
  If $j = n+1$, then for $m = 1,...,M$, if $U^d_{e^*\bullet}(n^d_{e^*\bullet}) \leq N^d_{e^*\bullet}/s_{e^*}$, then let $N^m_{e^*\bullet} = N^m_{e^*\bullet} - 1$.
- *Departure at e^* and arrival at j.*
  If $1 \leq j \leq n$, then for $m = 1,...,M$, if $U^d_{e^*\bullet}(n^d_{e^*\bullet}) \leq N^m_{e^*\bullet}/s_{e^*}$, then let $N^m_{e^*\bullet} = N^m_{e^*\bullet} - 1$. If also $N^m_j < C_j$, then let $N^m_j = N^m_j + 1$.

Let $n^s_{e^*\bullet} = n^s_{e^*\bullet} + 1$ and $T^s_{e^*\bullet} = T^* - \frac{\log(U^s_{e^*\bullet}(n^s_{e^*\bullet}))}{s_{e^*}\mu_{e^*}}$. Go to Step 1.

Step 4: *Departure from a communication blocking station e^*.*
Same as Step 3, except for the following:

- *Departure at e^* and arrival at j.*
  If $1 \leq j \leq n$, then for $m = 1,...,M$, if $U^d_{e^*\bullet}(n^d_{e^*\bullet}) \leq N^m_{e^*\bullet}/s_{e^*}$ and $N^m_j < C_j$, then let $N^m_{e^*\bullet} = N^m_{e^*\bullet} - 1$ and $N^m_j = N^m_j + 1$.

Algorithm 3.1 generates multiple sample paths of the Markov chain $\{N(t)\}$ simultaneously. To generate the potential service times at station i for all the sample paths, we generate exponentially distributed service times with rate $s_i\mu_i$. Since the number of busy servers at station i may be less than s_i, we determine whether a potential service completion is an actual service completion depending on the number of busy servers in station i of each sample path through *thinning by rejection*. Our approach is related to the methods proposed by Cassandras and Strickland (1989) and Vakili (1991) for using a common sequence of potential events in parallel simulation of multiple discrete event systems.

Suppose we are generating two sample paths of the system, sample paths A and B, using Algorithm 3.1. Let $N^A_i(t)$ and $N^B_i(t)$ denote the number of jobs at station i at time t in sample paths A and B, respectively, for $i = 1,...,n$. Let $T^{A,B}_c$ be the coupling time of sample paths A and B, that is $T^{A,B}_c = \inf\{t \geq 0 : N^A_i(t) = N^B_i(t), i = 1,...,n\}$. Let sample path 1 be the sample path that initiates at the state where all servers are idle and sample path 2 be the sample path that initiates at the state where all buffers are full. Finally, let $T^{1,2}_c$ denote the coupling time of sample paths 1 and 2.

The following two propositions show that sample paths of the queueing system S generated by Algorithm 3.1 satisfy the conditions that are required for the time segmentation approach to be applicable. The proofs of Propositions 3.1 and 3.2 are given in Hoseyni-Nasab and Andradóttir (1996a).

Proposition 3.1 (Monotonicity Property) *Suppose S is a queueing system of loss and communication blocking stations. Let A and B be two sample paths of S generated concurrently by Algorithm 3.1 and suppose $N^A_i(0) \leq N^B_i(0)$, for $i = 1,...,n$. Then $N^A_i(t) \leq N^B_i(t)$ for $i = 1,...,n$ and all $t \geq 0$.*

Proposition 3.2 *In S, if $\{N(t)\}$ is an irreducible Markov chain, then $E\{T^{1,2}_c\} < \infty$. Also, for any two sample paths A and B we have $T^{A,B}_c \leq T^{1,2}_c$ (and hence $E\{T^{A,B}_c\} < \infty$).*

Proposition 3.1 indicates that sample path B that is initially above sample path A will never go below sample path A. This clearly implies that once two sample paths couple, they will stay coupled afterwards. Proposition 3.2 indicates that all sample paths generated by Algorithm 3.1 will couple with probability one and no later than the coupling time of sample paths 1 and 2. Moreover, the expected coupling time of all sample paths is finite. Considering the discussion in Section 2, Propositions 3.1 and 3.2 guarantee the validity of the approach. It is clear that sample paths 1 and 2 serve as the bounding paths on every subinterval of the simulation and the simulation of a long sample path of the queueing system S can be carried out by assigning the available processors to subintervals of the time horizon of the simulation and following the procedure described in Section 2.

As we discussed in Section 2, the magnitude of the expected coupling time of the system is essential to the efficiency of the time segmentation approach. In particular, it is important to understand how the expected coupling time of the system changes with respect to changes in the parameters of the system. The following proposition is one of our results that indicates that, under certain conditions, the expected coupling time of the queueing network S grows no faster than linearly with respect to the number of stations in the system. The proof of Proposition 3.3 can be found in Hoseyni-Nasab and Andradóttir (1996a).

Proposition 3.3 *For the queueing network S, suppose $C_i \leq C < \infty$, for $i = 1,...,n$. Moreover, suppose that the Markov chain $\{N(t)\}$ is irreducible and that there exists a real number p such that $0 < p < 1$, and that for every station i at least one of the following two conditions is satisfied (for all $n \geq 1$):*

1. *$\lambda_i \geq p$ and $\frac{\lambda_i}{\lambda_i + s_i\mu_i + \sum_{j \neq i} p_{ji}s_j\mu_j} \geq p$; or*

2. $\mu_i \geq p$ and $\frac{\mu_i p_{i,n+1}}{\lambda_i + s_i\mu_i + \sum_{j \neq i} p_{ji} s_j \mu_j} \geq p$.

Let T_c denote the coupling time of sample paths 1 and 2. Then

$$E\{T_c\} \leq nM,$$

where M is a quantity depending on C and p.

In addition to the dependence of the expected coupling times on the number of stations in the system, we have studied the dependence of the expected coupling times on the buffer sizes, traffic rates, routing probabilities, and the types of the stations (loss vs. communication blocking) in the system. Our results indicate that the time segmentation approach can be efficiently used to simulate a large variety of queueing systems. More details on this issue can be found in Andradóttir and Ott (1995) and Hoseyni-Nasab and Andradóttir (1996a,b).

4 NUMERICAL RESULTS

In this section we present our numerical results aiming at better understanding the dependence of the expected coupling times of queueing networks on the parameters of the systems under study.

Our first set of numerical results has been obtained by simulating a network of n tandem communication blocking queueing stations with $s_i = 1$, $B_i = 10$, $\mu_i = 1$, for $i = 1, ..., n$, $\lambda_1 = 1$, $\lambda_i = 0$, for $i = 2, ..., n$, $p_{n,n+1} = 1$, $p_{i,n+1} = 0$, for $i = 1, ..., n-1$, and $n \in \{1, 2, 5, 10, 20, 100\}$. Note that for $n > 2$ this queueing system does not satisfy the conditions of Proposition 3.3. All clock times are independent and exponentially distributed. In each simulation two bounding sample paths are generated using Algorithm 3.1, one starting with all servers idle and one starting with all buffers full. The confidence intervals for the expected coupling times are obtained by generating 100 independent replications of the two sample paths. The results of our experiments are presented in Table 4.1.

The results in Table 4.1 suggest that for this class of queueing networks, the expected coupling times do not grow linearly with respect to the number of stations in the system. The growth rate of the expected coupling times appears to be beyond linear but no more than quadratic. This indicates that the conditions of Proposition 3.3 are at least to some extent necessary. To further investigate this issue we conducted simulations of two different variations of the queueing system of our first set of simulations. We slightly modified the parameters of the system in such a way that the modified systems satisfy the conditions of Proposition 3.3. Tables 4.2 and 4.3 show the results of simulations of two sets of queueing networks

Table 1: Dependence of the Expected Coupling Times of Tandem Networks of Communication Blocking Stations on the Number of Queueing Stations in the System when $\lambda_1 = 1, \lambda_i = 0$, for $i > 1$, $p_{n,n+1} = 1$, and $p_{i,n+1} = 0$, for $i < n$

Number of Stations, n	Coupling Times (95% Confidence Interval)
1	34.91 ($\pm$ 2.92)
2	66.09 ($\pm$ 5.60)
5	174.23 ($\pm$ 12.97)
10	435.92 ($\pm$ 26.64)
20	1,286.84 ($\pm$ 92.71)
100	12,061.88 ($\pm$ 772.24)

Table 2: Dependence of the Expected Coupling Times of Tandem Networks of Communication Blocking Stations on the Number of Queueing Stations in the System when $\lambda_1 = 1$, $\lambda_i = 0.1$, for $i > 1$, $p_{n,n+1} = 1$, and $p_{i,n+1} = 0$, for $i < n$

Number of Stations, n	Coupling Times (95% Confidence Interval)
1	34.91 ($\pm$ 2.92)
2	69.44 ($\pm$ 5.17)
5	106.13 ($\pm$ 6.40)
10	132.29 ($\pm$ 4.89)
20	159.18 ($\pm$ 6.20)
100	181.39 ($\pm$ 3.65)

that are different from the previous example in that $\lambda_i = 0.1$, for $i > 1$ (in Table 4.2), and $p_{i,n+1} = 0.1$, for $i < n$ (in Table 4.3), respectively. It is clear that even for large n these networks satisfy the conditions of Proposition 3.3. Tables 4.2 and 4.3 indicate that for the modified systems the expected coupling times grow sublinearly with respect to n. A thorough study of the conditions that affect the behavior of the expected coupling times is a subject of our current research.

5 CONCLUDING REMARKS

In this paper we presented a method for parallel simulation of discrete event systems, namely the time segmentation method, that is applicable to several classes of queueing networks. We presented our approach in a general setting for discrete event systems whose sample paths satisfy certain constructibility conditions. We showed that the efficiency of the approach is closely related to the expected coupling time

Table 3: Dependence of the Expected Coupling Times of Tandem Networks of Communication Blocking Stations on the Number of Queueing Stations in the System when $\lambda_1 = 1, \lambda_i = 0$, for $i > 1$, $p_{n,n+1} = 1$ and $p_{i,n+1} = 0.1$, for $i < n$

Number of Stations, n	Coupling Times (95% Confidence Interval)
1	34.91 ($\pm$ 2.92)
2	59.18 ($\pm$ 5.04)
5	114.86 ($\pm$ 6.24)
10	150.75 ($\pm$ 6.08)
20	184.42 ($\pm$ 5.53)
100	284.79 ($\pm$ 4.22)

of the sample paths of the system under study. We applied the time segmentation method to a family of Markovian queueing networks containing loss and communication blocking stations and investigated the validity and efficiency of the method through a variety of numerical and analytical results.

Our current research is directed towards understanding the behavior of the expected coupling times with respect to changes in the parameters of the systems under study, investigating further applications of the time segmentation approach, and studying the performance of the time segmentation approach in comparison with other parallel simulation methods.

ACKNOWLEDGMENT

This research was supported in part by the National Science Foundation under grants DDM-9210679 and DMI-9523111.

REFERENCES

Andradóttir, S., and T. 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.

Cassandras, C. G., and S. G. Strickland. 1989. Sample Path Properties of Timed Discrete Event Systems. *Proceedings of the IEEE* 77(1):59–71.

Chen, L., and R. Serfozo. 1995. Parallel Simulation by Multi-Instruction, Longest-Path Algorithms. Preprint.

Fujimoto, R. M. 1993. Parallel and Distributed Discrete Event Simulation: Algorithms and Applications. *Proceedings of the 1993 Winter Simulation Conference*, eds. G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, 106–114.

Greenberg, A., B. Lubachevsky and I. Mitrani. 1990. Unboundedly Parallel Simulation Via Recurrence Relations. *Proceedings of the Conference on Measurement and Modeling of Computer Systems*, 1–12.

Heidelberger, P. 1988. Discrete Event Simulation and Parallel Processing: Statistical Properties. *SIAM Journal on Statistical Computing* 9(6):1114–1132.

Heidelberger, P., and H. Stone. 1990. Parallel Trace Driven Cache Simulation by Time Partitioning. *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci, R. P. Sadowski, and R. E. Nance, 734–737.

Hoseyni-Nasab, M., and S. Andradóttir. 1996a. Efficiency of Time Segmentation Parallel Simulation of Finite Markovian Queueing Networks. Working Paper.

Hoseyni-Nasab, M., and S. Andradóttir. 1996b. Parallel Simulation of Transfer Lines by Time Segmentation. Working Paper.

Lin, Y., and D. Lazowska. 1991. A Time Division Algorithm for Parallel Simulation. *ACM Transactions on Modeling and Computer Simulation* 1 (1):73–83.

Vakili, P. 1991. Using a Standard Clock Technique for Efficient Simulation. *Operations Research Letters* 10(8):445–452.

AUTHOR BIOGRAPHIES

MEHDI HOSEYNI-NASAB is a Ph.D. student in the Department of Industrial Engineering of the University of Wisconsin–Madison and a guest student in the School of Industrial and Systems Engineering of the Georgia Institute of Technology. His research interests include parallel simulation and applied probability. He is a member of INFORMS and IIE.

SIGRÚN ANDRADÓTTIR is an Associate Professor in the School of Industrial and Systems Engineering at the Georgia Institute of Technology, on leave from the University of Wisconsin – Madison where she has held a faculty position since 1990. 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.