Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

PARALLEL REPLICATED SIMULATION OF MARKOV CHAINS: IMPLEMENTATION AND VARIANCE REDUCTION

Simon Streltsov

Department of Manufacturing Engineering
Boston University
Boston, MA 02215

Pirooz Vakili

Department of Manufacturing Engineering
Boston University
Boston, MA 02215

ABSTRACT

We discuss a way of simulating M replications of a uniformizable Markov chain simultaneously and in parallel (the so-called parallel replication approach). Simulation is performed to estimate the expectation of some cumulative reward over a finite deterministic time horizon. Distributed implementation on a number of processors and parallel SIMD implementation on massively parallel computers are described.
We investigate the adaptation of Fishman's rotation sampling approach (Fishman 1983a and 1983b) to our setting in order to reduce the variance of the estimators. It is pointed out that the algorithms can be used in conjunction with the Standard Clock simulation of uniformized chains at distinct parameters (Vakili 1992) to increase the effectiveness of the Standard Clock simulation.

1 INTRODUCTION

Parallel/distributed approaches for discrete-event simulation are categorized as *single simulation run* and *multiple runs* methods; the latter corresponding either to multiple statistical replications of one system, the parallel replications approach (Glynn and Heidelberger 1991), or to simulation runs of one system at multiple parameter settings (Vakili 1992). Often the approaches in these two categories are designed to address different computational issues of discrete-event simulation. Those in the first category address the simulation of a large discrete-event system (for a review see Fujimoto (1990)) where the size and complexity of the system renders its simulation on sequential machines inefficient or infeasible. Those in the second category address the computational issues related to cases where a large number of statistical replications of a system is required to achieve acceptable levels of accuracy, or when the system is to be simulated at a large number of parameter settings in order to, for example, find an optimal setting.

In this paper we describe a way of implementing M replications of a system in parallel. Our model of parallel simulation is very similar to that of the so-called Standard Clock simulation of a system at multiple parameter settings (Vakili 1992). In fact using both approaches in conjunction with each other, where, for example, a system is simulated at N parameter settings and for each setting M replications are run, is highly desirable. Parallel simulation via Standard Clock allows for dynamic and simultaneous comparisons at multiple parameter settings. Simulating multiple replications at each setting simultaneously makes the comparisons significantly more effective.

We limit ourselves to uniformized Markov chains. Generalizations of this setting is possible (see Vakili 1992). Our basic model is similar to that used in (Heidelberger and Nicol 1992, 1993a and 1993b). We assume that the objective of the simulation is to estimate the expected value of some cumulative "reward/cost" over a finite deterministic horizon. Random horizons require a separate treatment (see Glynn and Heidelberger (1991), Heidelberger (1988)). We investigate the adaptation of Fishman's rotation sampling approach (Fishman 1983a and 1983b) to our setting in order to reduce the variance of the estimators.

The paper is organized as follows: preliminaries are given in section 2. Section 3 and 4 describe parallel independent and parallel correlated simulation algorithms. Experimental results are presented in section 5. We conclude in section 6.

2 PRELIMINARIES

In this section we present the basic setting of the paper and establish notation.

2.1 Simulation of Markov chains

Let $X = \{X_t, t \geq 0\}$ be a continuous-time Markov chain (CTMC) on a (finite or) countable set S. Let Q_{ij} be the rate of transitions from state s_i to state s_j, and let $q_i = -Q_{ii}$ be the total rate of transitions out of s_i. All through this paper we assume that the chains simulated are *uniformizable*, i.e., Q is bounded ($\sup q_i < \infty$). Given that the chain is in some state, say s_i, it remains in that state for a duration that is exponentially distributed with mean q_i^{-1}. Given a transition out of s_i, the next state of the chain - a discrete random variable - has probability mass Q_{ij}/q_i at state j. One strategy for simulation, therefore, is to generate samples of an exponential residence time and a discrete random variable of the next state for each transition.

2.2 Uniformization

For uniformizable chains, the sojourn times of the chain in states can be *uniformized* by appropriately introducing extra fictitious transitions from states to themselves. The inter-event times can thus be made an i.i.d. sequence of exponential random variables independent of the states of the chain. This often leads to simplifications that can be exploited to design alternative simulation strategies that in one way or another improve upon the above straightforward approach (e.g., see Hordijk et al. (1976), Fox and Glynn (1990), Heidelberger (1992, 1993a and 1993b) and Vakili (1992)). In this paper we assume that the chain is uniformized.

More precisely, let $X = \{X_t, t \geq 0\}$ be a CTMC with infinitesimal generator Q bounded by Λ. Let $N^\Lambda = \{N_t, t \geq 0\}$ be a Poisson process with rate Λ and $Y = \{Y_n, n \geq 0\}$ a discrete-time Markov chain (DTMC) with transition probability matrix $P = I + \Lambda^{-1}Q$ (where I is the identity), with N^Λ and Y mutually independent. If X_0 has the distribution of Y_0, then $\{X_t, t \geq 0\}$ and $\{Y_{N_t}, t \geq 0\}$ are equal in law. The Poisson process N^Λ determines the (potential) state transition epochs of the CTMC X, while the state transitions are determined by the DTMC Y.

2.3 The Model

We limit ourselves to those chains that are "well-structured". We assume that all state transitions are caused by a finite number of *events*. It is implicitly assumed that the transitions caused by the same event in all states are in some sense "similar"; for example, in simple migration processes (see Kelly (1979), Ch. 2) the effect of a migration from colony i to colony j is similar in all states: it corresponds to the reduction of the population of colony i by one and the increase of the population of colony j by one. This additional structure - which is often present in simulated systems - is not easily discernible from the transition rate matrix Q. To make things more explicit we consider the following model:

Let S denote, as above, the state space. Let $E = \{e^1, \ldots, e^K\}$ be the set of events, K finite. To each event e there corresponds a state transition rule

$$f_{(e,u)} : S \to S.$$

where u is a uniform r.v. on $(0,1)$ used to generate probabilistic transitions. Note that $f_{(e,u)}$ is defined on all of S. If event e is not active in state s, then $f_{(e,u)}(s) = s$, corresponding to a null transition.

Let $(\tau, \varepsilon, v) = \{(\tau_n, \varepsilon_n, v_n), n \geq 0\}$ be a marked Poisson process, where $\{\tau_n, n \geq 0\}$ is the sequence of arrival instances of N^Λ, $\{\varepsilon_n, n \geq 0\}$ is an i.i.d. sequence of discrete random variables, independent of the Poisson process N^Λ, such that $\varepsilon_n \in E$ and $P(\varepsilon_n = e^i) = p_i$, and $\{v_n, n \geq 0\}$ is an i.i.d. sequence of uniform random variables on $(0,1)$ independent of the sequences τ and ε. Λ is the rate at which the clock ticks, τ_n is the n-th tick of the clock, ε_n is the type of event that occurs at the n-th tick of the clock, and v_n is the uniform r.v. used to generate the n-th state transition.

Given initial state $Y_0 = X_0$, the state of the system evolves as follows:

$$Y_n = f_{(\varepsilon_n, v_n)} \circ f_{(\varepsilon_{n-1}, v_{n-1})} \circ \cdots \circ f_{(\varepsilon_1, v_1)}(X_0)$$

and

$$X_t = \sum_{n=0}^{\infty} Y_n I\{\tau_n \leq t < \tau_{n+1}\}, \text{ for } t \geq 0$$

($I\{.\}$ is the indicator function.)

Let $g : S \to \mathbf{R}$ be a real-valued function defined on the state space of the system, and let T be a finite deterministic stopping time and let

$$L = \int_0^T g(X_t)dt$$

be a random variable representing some cumulative cost/reward over $[0,T]$. Our objective is to estimate $\theta = E[L]$ via simulation.

Remark 2.1 : This setting avoids some of the issues that arise when random stopping times are selected. See, for example, Glynn and Heidelberger (1991), Heidelberger (1988), where random stopping times are considered and it is shown that when simulation

experiments are not carefully designed the estimators may be biased. The case of random stopping times requires special treatment which will not be considered in this paper.

Example 2.1 Consider a tandem queueing network with K servers and buffer sizes $\{b_l, l = 1, \ldots, K\}$. Customers arrive to the system according to a Poisson process with rate λ_0, and service times at server l are assumed to be exponentially distributed with rate λ_l.
Let $Y_n(l)$ be the number of customers at queue l immediately after the n-th transition. The set of the events consist of $K+1$ events $\{e^l, l = 0, \ldots, K\}$ where e^0 corresponds to the arrival event and e^l to the departure from server $l, l = 1, \ldots, K$. Event e^0 is active at all times; event e^l is active, if $Y_n(l) > 0$ and $Y_n(l+1) < b(l+1)$. Let $g_n = \sum_{l=1}^{K} Y_n(l)$, be the total number of customers in the system.

To simulate a uniformized chain a uniform r.v. u_n is generated at the n-th tick of the clock. Then $\varepsilon_n = e^l$ if $\sum_{i=0}^{l-1} \lambda_i \leq u_n < \sum_{i=0}^{l} \lambda_i$. The transition rule f_{e^l} is applied to determine the next state of the system.

2.4 Discrete-time conversion

Uniformization is often used to convert a continuous-time problem to a discrete-time equivalent. In theoretical investigations such a discrete-time conversion simplifies the analysis (see Keilson 1979) and in simulation it can lead to variance reduction (see Hordijk et al. 1976); the latter sometimes involve the extra cost of generating some null transitions. The cost is already incurred in our context; hence from now on we consider the discrete-time version of the problem and assume that the discrete-time chain $Y = \{Y_n; n \geq 0\}$ is simulated. More precisely, let

$$N_T = \text{Max}\{n; \tau_n \leq T\}$$

N_T is Poisson distributed with rate ΛT. Let $L' = E[L|Y, N_T]$, then

$$L' = \sum_{n=0}^{N_T} g(Y_n) \frac{T}{N_T + 1}$$

and $E[L'] = \theta$. Hence to estimate θ, it is sufficient to simulate N_T and Y. For a general discussion of this approach see (Fox and Glynn 1990).

3 PARALLEL (INDEPENDENT) REPLICATED SIMULATION

To simplify the presentation we assume that all transitions $f_{(e,u)}$ are deterministic, i.e., once event ε_n is generated, there is no need to generate v_n to determine the next state. The extension to the probabilistic case is straightforward. We assume that we are to simulate M replications of the system to estimate θ.

3.1 Model of parallel simulation

Let $\varepsilon^j = \{\varepsilon_n^j; n \geq 1\}$ and $Y^j = \{Y_n^j; n \geq 0\}$ be, respectively, the i.i.d. sequence of events that drive the j-th replication and the sequence of states visited by the j-th replication $(j = 1, \ldots M)$. In this section we assume that ε^j's are independent for $j = 1, \ldots M$.

Similar to the parallel simulation of a parametric family of chains (see Glasserman and Vakili 1992), we consider the following model for the parallel simulation of parallel replications.

Let $\underline{e} \in E^M$. Define $F_{\underline{e}} : \prod_1^M S \to \prod_1^M S$ componentwise, i.e.,

$$F_{\underline{e}}(y^1, \ldots, y^M) = (f_{e_1}(y^1), \ldots, f_{e_M}(y^M))$$

The process $\underline{Y} = \{\underline{Y}_n; t \geq 0\}$, defined on $\prod_1^M S$, is given by

$$\underline{Y}_n = F_{\underline{\varepsilon}_n} \circ F_{\underline{\varepsilon}_{(n-1)}} \circ \ldots \circ F_{\underline{\varepsilon}_1}(\underline{Y}_0)$$

where $\underline{\varepsilon}_n = (\varepsilon_n^1, \ldots, \varepsilon_n^M)$.

For example, when M replications of a tandem queueing network are simulated, at each tick, M independent uniform r.v. – u_n^j for replication j – are generated; event ε_n^j is determined using u_n^j;

Remark 3.1 A note of clarification is in order here. This construction introduces a coupling across M replications in continuous time. We assume that the same Poisson process N^Λ is shared between replications and

$$X_t^j = \sum_{n=0}^{\infty} Y_n^j I\{\tau_n \leq t < \tau_{n+1}\}, \text{ for } t \geq 0$$

for $j = 1, \ldots M$. Moreover, to estimate θ, we will use the same N_T for all replications. Nonetheless, it can be easily verified that the processes X^j (and Y^j) have the right marginal probability laws $(j = 1, \ldots M)$.

3.2 Parallel implementation

The above construction can be easily implemented in a variety of computational environments. The construction has two basic steps: (i) generation of $\underline{\varepsilon}_n = (\varepsilon_n^1, \ldots, \varepsilon_n^M)$, and (ii) evaluation of $F_{\underline{\varepsilon}_n}(\underline{Y}_{(n-1)})$.

In a parallel/distributed implementation assume each (or a group of) processor(s) simulates one replication. First a sample of N_T is generated and reported to all processors. Subsequently, each processor

evaluates its associated ϵ_n^j and $f_{\epsilon_n^j}(Y_{(n-1)}^j)$. Little, if any, coordination is required across processors during the simulation (however, such a coordination can be imposed so that the processors execute their codes in lock-step).

In a SIMD implementation, a sample of N_T is generated at the front end. Again assume that each (or a group of) processor(s) simulates one replication. Processors evaluate their associated ϵ_n^j and $f_{\epsilon_n^j}(Y_{(n-1)}^j)$ in parallel and in a SIMD fashion (the front-end directs them to find ϵ_n^j . Once all processors have evaluated ϵ_n^j, they are directed by the front-end to evaluate $f_{\epsilon_n^j}(Y_{(n-1)}^j)$.

Remark 3.2 A comparison of the model of parallel simulation of parametric families of chains (see Glasserman and Vakili 1992) and the above model and implementation strategy reveals that the two can be easily implemented in conjunction with each other. In other words multiple replications of a number of chains that have distinct parameters can be simulated simultaneously and in parallel.

Remark 3.3 Note that a significant part of the simulation in the above model is to sample from ϵ_n^j. ϵ_n^j's are identically distributed across j and n. Samples of the discrete r.v. ϵ_n^j can be generated in $O(1)$ (independent of K) using the Alias method (see Bratley 1987). The setup for this method is done only once for all n and all j.

4 PARALLEL (CORRELATED) REPLICATED SIMULATION

Our objective here is to reduce the variance of the estimator for θ by introducing correlation across replications. The approach is based on the use of rotation sampling, proposed in (Fishman and Huang 1983) and is inspired by the particular application of rotation sampling to the simulation of Markov chains as reported in (Fishman 1983a and 1983b).

We introduce correlation across ϵ_n^j for $j = 1, \ldots, M$. Let

$$I^i = \sum_{j=1}^{M} I\{\epsilon_n^j = e^i\} \qquad \text{for} \qquad i = 1, \ldots K$$

In words, I^i is the number of replications at the n-th transition that observe event e^i. In parallel independent replications $I\{\epsilon_n^j = e^i\}$'s $(j = 1, \ldots, M)$ are M i.i.d. Bernoulli random variables with parameter p_i; hence $I^i = \sum_{j=1}^{M} I\{\epsilon_n^j = e^i\}$ is a binomial random variable with parameters M and p_i. Let $I^{i\prime}$ be the number of correlated replications observing event e^i. When rotation sampling is introduced $I^{i\prime} = \lfloor p_i M \rfloor + Z$ where Z is a Bernoulli r.v. ($\lfloor a \rfloor$ = integer part of a). In this case the sampling seems to minimize number of samples that agree, a property that, informally, appears desirable for variance reduction (see (Devroye 1990) for the case where maximal coupling, i.e., maximizing the number of agreeing samples is desirable).

We first give a brief introduction to rotation sampling.

4.1 Rotation Sampling

Let $u_1, u_2, \ldots, u_M$ be M uniform r.v.s on $(0, 1)$ and let $g_1(u_1), g_2(u_2), \ldots, g_M(u_M)$ be M corresponding response functions. A relevant problem is to impose appropriate dependencies among $u_1, u_2, \ldots, u_M$ - while retaining the uniform marginal distributions - to minimize the following variance

$$\mathrm{Var}\left(\frac{1}{M}\sum_{j=1}^{M} g_j(u_j)\right).$$

It is shown (see Fishman and Huang 1983) that if we limit ourselves to the set Ω of one-to-one mappings ω of $(0, 1)$ onto itself where except for a finite number of points $d\omega/dx = 1$, we can approximate the infimum of the above variance as closely as we wish by inducing dependencies via setting

$$u_j = \omega_j(u) \qquad \omega_j \in \Omega$$

In rotation sampling ω's are limited to a subset of Ω, namely, the class of one-parameter rotation transformations

$$\begin{aligned} \omega_j(u) &= u \oplus \delta_j \\ &= u + \delta_j \qquad \text{for } 0 \le u < 1 - \delta_j \\ &= u + \delta_j - 1 \quad \text{for } 1 - \delta_j \le u < 1 \end{aligned}$$

$(0 \le \delta_j \le 1,\ j = 1, \ldots, M)$.

It is shown that for $g_j = g, j = 1, \ldots M$, and under appropriate conditions, the sampling corresponding to $\delta_j^* = \frac{j}{n}$ for $j = 1, \ldots, M$ is optimal among all rotation samplings. In this case the variance of the average is of the order of $O(1/M^2)$ as compared to $O(1/M)$ for independent sampling (see Fishman and Huang 1983) .

4.2 Model of parallel correlated simulation

Let $\phi : (0, 1) \to E$ be the function used to generate samples of ϵ_n (this may represent an inverse transform or an application of Alias method.) Let

$\{u_n; n \geq 1\}$ be a sequence of i.i.d. uniform r.v.s on $(0,1)$. Then we define $\epsilon_n^{j\prime}$, the n-th event at replication j, by

$$\epsilon_n^{j\prime} = \phi([u_n + \frac{\pi_n(j)}{M}]\text{Mod}(1)),$$

where $\pi_n(j)$ is a permutation of $\{1, 2, \ldots, M\}$.
Then $\underline{Y}' = \{\underline{Y}'_n; n \geq 0\}$, defined on $\prod_1^M S$, is given by

$$\underline{Y}'_n = F_{\underline{\epsilon}'_n} \circ F_{\underline{\epsilon}'_{(n-1)}} \circ \ldots \circ F_{\underline{\epsilon}'_1}(\underline{Y}'_0)$$

where $\underline{\epsilon}'_n = (\epsilon_n^{1\prime}, \ldots, \epsilon_n^{M\prime})$.
Renumbering systems after each step with the permutation π_n allows us to control dependencies between certain pairs or groups of replications over time. Note that the permutation π_n can be any function of $Y_1, \ldots Y_{n-1}$ without distorting marginal distributions of $\epsilon_n^{j\prime}$. The proper choice of π_n is a topic of current research.

We investigated several choices of $\pi_n(j)$. The choice $\pi_n(j) = j$ for all j seems undesirable, because then $Var(g(u_i, u_j))$ is a function of $\mid i - j \mid$. We can achieve independence of $Var(g(u_i, u_j))$ from indices by generating random permutation $\pi_n(j)$ (Method 1). If some functional $G(Y_n^j)$, reflecting an ordering of current states of each replication can be defined for the simulated system, then we can try to improve the algorithm by increasing rotations between systems with "close" or equal values of $G(Y_n^j)$. For example, for 8 systems, numbered such that $G(Y_n^j) > G(Y_n^i)$ for $j > i$, $\pi = \{1, 5, 3, 7, 2, 6, 4, 8\}$ (Method 2).

Example 4.1 For each replication j of the tandem queueing network $u_n^{j\prime} = [u_n + \frac{\pi_n(j)}{M}]Mod(1)$ is used to determine $\epsilon_n^{j\prime}$. For method 1, $\pi(j)$ is a random permutation of $1, \ldots, M$.
For method 2 we need to define some measure of "closeness" between systems. We define it to be the weighted average queue lengths

$$G = \frac{1}{K}\sum_{l=1}^{K} \alpha_n^l Y_n^j(l),$$

where α_n is a parameter.
For the $M/M/1$ queue $G = Y_n^j$ provides a natural measure of "closeness".

4.3 Parallel implementation

Except for the generation of $\underline{\epsilon}'_n$, the implementation is identical to that of parallel independent replications.

In parallel/distributed implementation on a number of processors and in SIMD implementations either a central mechanism generates the uniform random numbers used for event determination (i.e., u_n's) or via appropriate selection of seeds at processors one insures that the same sample of u_n is generated at all processors.

5 EXPERIMENTAL RESULTS

In this section we report simulation results for an $M/M/1$ queue ($\rho = 0.2, 0.5$ and 0.9) and two tandem queueing networks with blocking. Average queue length

$$\frac{1}{T}\int_0^T X(t)dt$$

is chosen as the total cost. T denotes the time horizon.
For the $M/M/1$ queue we ran independent simulations, simulations using Fishman's rotation method (described in Fishman (1983b)) and using methods 1 and 2, described above. For method 2 we used $G = Y_n^j$.

Table 1: $M/M/1$ queue

ρ	T_{sim}	k	$\frac{var_{ind}}{var_F}$	$\frac{var_{ind}}{var_1}$	$\frac{var_{ind}}{var_2}$
0.2	1000	8	1.26	2.03	2.52
0.2	1000	16	2.04	2.00	3.04
0.2	1000	32	2.53	3.11	4.92
0.2	1000	64	2.74	2.66	7.22
0.2	1000	128	4.45	2.57	7.54
0.2	1000	256	4.28	2.45	12.72
0.2	1000	512	4.83	2.88	15.53
0.5	10000	8	1.13	1.48	2.68
0.5	10000	16	1.41	1.99	5.02
0.5	10000	32	1.99	2.34	5.63
0.5	10000	64	3.68	2.81	11.18
0.5	10000	128	4.38	2.69	14.37
0.9	10000	8	1.44	4.02	5.98
0.9	10000	16	2.59	4.44	10.18
0.9	10000	32	5.39	5.52	24.97
0.9	10000	64	9.08	5.84	33.89
0.9	10000	128	16.07	4.46	37.34
0.9	100000	8	1.25	2.08	3.89
0.9	100000	16	1.13	2.40	6.03
0.9	100000	32	1.84	3.99	9.71
0.9	100000	64	1.79	2.49	15.32
0.9	100000	128	3.04	9.41	82.31

For each system we report the ratio of the variance of the estimator of the total cost of independent simulations to the variance of rotation estimator for different values of the rotation parameter k.

The goal of these experiments is to investigate the degree of variance reduction achieved by the proposed methods.

The results (Table 1) show that for $M/M/1$ queue methods 1 and 2 generally achieve variance reduction not less than the method of (Fishman 1983b). We also observe that Method 2 has a bigger potential, especially for large values of k.

We also simulated tandem queueing network with blocking. The simulated systems have 10 servers with equal service rates and a given ratio of the arrival rate to the service rate ($\rho = 0.5$). We consider 2 systems with different buffer sizes. To apply method 2 to the tandem queueing network with K servers, we use $G = \frac{1}{L}\sum_{l=1}^{K} \alpha^l Y_n^j(l)$. We report results for method 1 and several modifications of method 2 with different values of the parameter $\alpha = 0, 1$, and 1.1 (Tables 2 and 3).

Table 2: Tandem queue with 10 buffers, $\rho = 0.5$ Buffer sizes { 3 3 3 3 3 3 3 3 3 3 }.

T_{sim}	k	$\frac{var_{ind}}{var_1}$	$\frac{var_{ind}}{var_{\alpha=0}}$	$\frac{var_{ind}}{var_{\alpha=1}}$	$\frac{var_{ind}}{var_{\alpha=1.1}}$
10000	8	1.39	1.35	1.42	1.49
10000	16	2.00	1.82	1.78	1.75
10000	32	1.96	2.00	2.21	2.01
10000	64	2.75	2.19	3.20	3.38
10000	128	2.92	2.73	1.84	2.38
10000	256	3.22	2.14	2.87	3.71
10000	512	3.95	2.19	1.94	3.38

Table 3: Tandem queueing network with 10 buffers, $\rho = 0.5$ Buffer sizes { 10 4 4 4 4 4 4 4 4 4 }

T_{sim}	k	$\frac{var_{ind}}{var_1}$	$\frac{var_{ind}}{var_{\alpha=0}}$	$\frac{var_{ind}}{var_{\alpha=1}}$	$\frac{var_{ind}}{var_{\alpha=1.1}}$
10000	8	1.29	1.27	1.22	1.29
10000	16	2.12	1.73	1.65	1.83
10000	32	2.77	2.96	3.05	3.03
10000	64	4.12	3.97	3.97	4.06
10000	128	4.25	5.85	4.23	4.10
10000	256	5.74	2.18	2.86	3.02

6 SUMMARY

We described a way of simulating M replications of a uniformizable Markov chain simultaneously and in parallel. Distributed and massively parallel SIMD implementations on a number of processors and on massively parallel computers were described. We described ways of adapting Fishman's rotation sampling approach (Fishman 1983a and 1983b) to our setting to reduce the variance of the estimators. Understanding the tradeoffs and efficiency of these algorithms is a future project. It is worth investigating the degree of variance reduction and conditions that guarantee it.

REFERENCES

Bratley, P.B., B.L. Fox, and L. E. Schrage. 1987. *A Guide to Simulation.* 2d ed., New York: Springer-Verlag.

Devroye, L. 1990. Coupled Samples in Simulation. *Oper. Res.*, **38**, 115-126.

Fishman, G.S. 1983a. Accelerated Accuracy in the Simulation of Markov Chains, *Oper. Res.*, **31**, 466-487.

Fishman, G.S. 1983b. Accelerated Convergence in the Simulation of Countably Infinite State Markov Chains, *Oper. Res.*, **31**, 1074-1089.

Fishman, G.S., and B.D. Huang. 1983. Antithetic Variates Revisited, *Communications of the ACM,* v.26, n.11.

Fox, B.L. 1990. Generating Markov-Chain Transitions Quickly: I, *ORSA Journal on Computing*, Vol. 2, **2**, 126-135.

Fox, B.L., and P.W. Glynn. 1990. Discrete time conversion for simulating finite-horizon markov processes, *SIAM Journal on Appl. Math.*, 50(5), pp.1457–1473.

Fujimoto, R.M. 1990. Parallel Discrete Event Simulation, *Communication of ACM*, Vol 33, 10, pp. 30-53.

Glasserman, P., and P. Vakili. 1992. Correlation of Uniformized Markov Chains Simulated in Parallel, *Proc. of 1992 Winter Simulation Conference,* 475-482.

Glynn, P.W., and P. Heidelberger. 1991. Analysis of Parallel Replicated Simulations Under a Completion Time Constraint, *ACM Transactions on Modeling and Computer Simulation*, Vol. 1 No.1, pp. 3-23.

Heidelberger, P. 1988. Discrete Event Simulations and Parallel Processing : Statistical Properties, *SIAM J. Sci Stat. Comput.* Vol. 9, pp. 1114-1132.

Heidelberger, P., and D. M. Nicol. 1991. Simultaneous Parallel Simulations of Continuous Time Markov Chains at Multiple Parameter Settings, *Proceedings of 1991 Winter Simulation Conference*, pp. 602-607.

Heidelberger, P., and D. M. Nicol. 1993. Conservative Parallel Simulation of Continuous Time Markov Chains using Uniformization, to appear in *IEEE Transactions on Parallel and Distributed Systems.*

Heidelberger, P., and D. M. Nicol. 1993b. Parallel Simulation of Markovian Queueing Networks Using Adaptive Uniformization, to appear in *Proceedings of 1993 ACM Sigmetrics Conference.*

Hordijk, A., D.L. Iglehart, and R. Schassberger. 1976. Discrete Time Methods for Simulating Continuous Time Markov Chains, *Adv. Appl. Prob.*, **8**, 772-788.

Keilson, J. 1979. Markov Chain Models - Rarity and Exponentiality, *Springer-Verlage.*

Kelly, F.P. 1979. *Reversibility and Stochastic Networks*, John Wiley & Sons.

Vakili, P. 1992. Massively parallel and distributed simulation of a class of discrete event system: a different perspective, *TOMACS, Vol.2,No.3,, pp.214-238.*

AUTHOR BIOGRAPHIES

SIMON STRELTSOV is a Ph.D. Student in the department of Manufacturing Engineering at Boston University. His research interests include global optimization, selection methods and parallel and distributed simulation.

PIROOZ VAKILI is an Assistant Professor in the department of Manufacturing Engineering at Boston University. His research interests include gradient estimation, stochastic optimization, and parallel and distributed simulation.