BETTER SELECTION OF THE BEST

Huaiyu Harry Ma
Thomas R. Willemain

Department of Decision Sciences and Engineering Systems
Rensselaer Polytechnic Institute
Troy, NY 12180-3590, U.S.A.

ABSTRACT

We present a new method of selecting the best of several competing system designs on the basis of expected steady-state performance. The method uses a new form of time-series bootstrap in a sequential hypothesis testing framework. We illustrate the efficiency of the new method by comparing its results against the current state of the art in a series of simulation experiments. The new method achieves equal or greater probability of correct selection of the best system with substantial savings in the number of simulation observations required for the decision.

1 INTRODUCTION

Our focus is the selection portion of the well-known simulation ranking and selection problem (Law and Kelton 2000, Kim and Nelson 2003). One potentially valuable application of discrete event simulation is rapid identification of the best among a set of competing system designs. Rapid identification is important when choosing among alternative ways to respond to disruptions in the operation of mission-critical systems. It is also important when sorting through the large number of alternatives generated by combinatorial design methods.

In our opinion, the state of the art is represented by the KN++ sequential selection method described in Goldsman et al. (2002); hereafter it is referred to as GKMN. KN++ is well developed, well justified, and efficient compared to its predecessors. It seemed to us that the best chance for further progress on this problem lay in investigating radically different approaches. The beginning of this new approach followed from the realization that methods that discard or overwhelm the initial transient (warmup period) waste useful data due to a mistaken assumption that only steady state data have relevance in ranking and selection problems. The seminal work of Sheth-Voss and colleagues (Sheth-Voss, Haddock, and Willemain 1996; Sheth-Voss Willemain, and Haddock forthcoming) inspired the new approach.

We describe a new approach that bends a few rules but produces excellent results. The method uses a few short simulation runs that begin with the alternative systems empty and idle and does not delete the transient data. The method then augments these data with artificial data created by a new variant of time series bootstrapping, bootstrapping with mirroring (BWM). This new bootstrap is an offshoot of the threshold bootstrap (Park and Willemain 1999, Park et al. 2001) with three interesting twists: it ignores the fact that the data are nonstationary, it creates "mirrored" data that is reflected around the zero axis, and it pools the results of multiple time series and their mirrored counterparts.

We compared the performance of this new transient-based method to the KN++ results by duplicating the experiments in GKMN. The performance comparisons were based on the same criteria: expected number of simulation observations required to arrive at a decision, and probability of correct selection (PCS) of the best of several alternative system designs. The new method achieved equal or better PCS with substantially fewer simulation observations.

2 STATE OF THE ART

The KN++ method (Goldsman et al. 2002) is a screening and selection procedure. With high probability, the screening procedure will choose a subset which contains the best alternatives, and the selection procedure will pick the best. The underlying approach is that of batch means (with or without overlapping): Produce one long simulation run, break it into batches to achieve approximately iid normal data, then perform inference using an indifference zone approach (Kim and Nelson 2001). An indifference zone δ, which describes the smallest absolute difference in expected performance that is considered important to detect, is set by the experimenter. KN++ guarantees, with confidence level greater than or equal to $1 - \alpha$, that the system ultimately selected has the best true mean performance when the mean performance of the best is at least δ better than the second best. When there are inferior systems whose means are within δ of the true best, then the procedure guarantees to find one of these "good" systems with the same probability.

KN++ uses a long initial series of observations to get a preliminary estimate of the means and variances of the expected performance measure, such as delay in queue. It then decides whether there is a clear winner among the competing designs. If so, the method stops with a recommendation. If not, KN++ obtains more simulation observations, updates the estimates of means and variances, and reconsiders the question. Such a sequential pair-wise comparison based method has the potential to eliminate alternatives at each stage so that the total simulation cost is reduced. By allowing the batch size to update each iteration and introducing better variance estimators, KN++ outperformed the previous methods tested in GKMB.

3 NEW METHODOLOGY

To explain the new method, consider the simple case of comparing two systems. To fix ideas, assume that the two systems are queues, and the performance measure of interest is the steady state mean delay in queue. We begin by simulating each system through n_0 customers, starting both systems empty and idle. From these observations, we compute the differences in delay for corresponding pairs of customers. Finally, we form the cumulative mean of the differences, which represents our first datum. We repeat this procedure M times to generate M values of the cumulative mean delay (M typically takes values in the range of 1 to 5).

If we had deleted the transient phase of each simulation replication, we would now have a pure case of independent replications, and we could perform a one-sample t-test on the null hypothesis that the mean difference is zero. By using the cumulative mean of the difference series, we allow the central limit effect to shape the distribution of the data toward the normality required for the t-test. However, in our method we forego the deletion of the transient phase, recognizing that the sign of the difference is what we need to know, not the magnitude, and that the sign is robust against the transient.

To get high power from such a t-test, we would need many independent replications. However, we wish to minimize the computational cost of running the simulation software, since our interest is in comparing large, complex systems. Accordingly, we use the M sequences of delay differences as the basis for creating B bootstrap samples. These B values provide a better estimate of the standard deviation of the cumulative mean difference for use in the t-test.

The bootstrap used at this stage is a new variant of the threshold bootstrap (TB). The TB works by dividing a stationary time series into "chunks" and then concatenating chunks chosen by sampling with replacement from the set of chunks (Park and Willemain 1999, Park et al. 2001). Chunks are composed of "cycles"; cycles are composed of consecutive high and low "runs"; runs are sequences of data values on the same side of a "threshold". In our preliminary work, we have used a threshold of zero and a chunk size of one cycle. (We think that time series length is more usefully measured in units of cycles than individual observations, and plan to develop guidelines for using the new method in those terms. Counting length in cycles automatically incorporates the effect of serial correlation on the effective sample size of the data.)

We have made several modifications to adapt the TB to the selection problem. First, while we recognize that data from the transient phase are not stationary, we simply ignore that fact and use all the data. Second, to enforce the null hypothesis of zero mean difference required in the selection procedure, and also to better capture intra-sample variation, we create from each difference series its "mirrored" version, which is simply the series reflected about the origin. Third, we form the set of chunks from which we resample by pooling the chunks from all M series of differences and M their mirror images. Pooling and mirroring allow us to better capture the inter-sample variation evident in highly autocorrelated time series. Then we finish by creating artificial time series of the same length as the real difference series, resampling from the pool of chunks derived from the 2M series and concatenating the sampled chunks until the bootstrap series has the required length. Repeating this process B times produces the final pseudo-data needed for a significance test. We regard the distribution of the B values as the null distribution of the raw data, which are the cumulative mean differences. A value of B around 100 or more seems to suffice.

At this point, the analysis proceeds along classical lines. We revert back to the actual simulation results, which consist of M values of the cumulative mean difference. The grand mean of these values is the test statistic. Its standard error is estimated by the standard deviation of the B values of the cumulative means of the bootstrap series divided by $\sqrt{M}$. Dividing the grand mean by this estimated standard error gives the sample value of the test statistic. Comparing the observed test statistic to a t-distribution with B-1 degrees of freedom yields a two-tail p-value for the observed simulation results.

The final stage of the method is to demand confirmation of apparently significant results. If the resulting p-value is below a threshold P, we reach a preliminary conclusion that the mean difference is not zero. However, we require confirmation before making a final conclusion, so we append a "data gulp" of g simulation observations to each of the M independent replications and re-analyze the problem. If we get C consecutive rejections from the same side of the null distribution, we determine that there is a winner and nominate the system with the lower grand mean for its M cumulative means. If, instead, the p-value falls above the threshold P or the rejection is from the other side of the null distribution, we append another data gulp, reset the count of consecutive rejections to zero, and repeat the analysis.

When there are more than two alternative systems involved, the new method compares all possible pairs of systems. If a clear winner is detected between two systems, the inferior one is eliminated for further consideration. The sequential procedure stops when there is only one system left.

Figure 1 represents the new approach in flowchart form.

4 SIMULATION EXPERIMENTS

To facilitate comparison, we duplicated the setup of the GKMN experiments. These used three types of systems: M/M/1 queues, AR(1) and MA(1) processes. The utilization levels of the queues and the extent of autocorrelation in the ARMA processes were exactly the same as in GKMN: 60% or 90% utilization for the best queues and lag-1 autocorrelations of 0.9 and 0.497 for the AR(1) and MA(1) processes, respectively. (Note that GKMN did not adjust these alternatives to have equal difficulty, so comparison across system types is not appropriate.) The comparisons for each system involved varying numbers of alternative systems. The alternatives were arranged in one of two configurations. In the MDM configuration, the expected performance of the alternatives was arranged in a sequence with a constant offset between the mean performance of the alternatives. In the SC configuration, which was applied only for the M/M/1 experiments, the best alternative had expected performance offset by the same amount from all the others.

The GKMN experiments were conducted under laboratory conditions. That is, certain parameters, such as the length of the initial simulation run, were established at values optimized using knowledge that would not be available in practice (e.g., the length of the initial run n_0). Other pa-

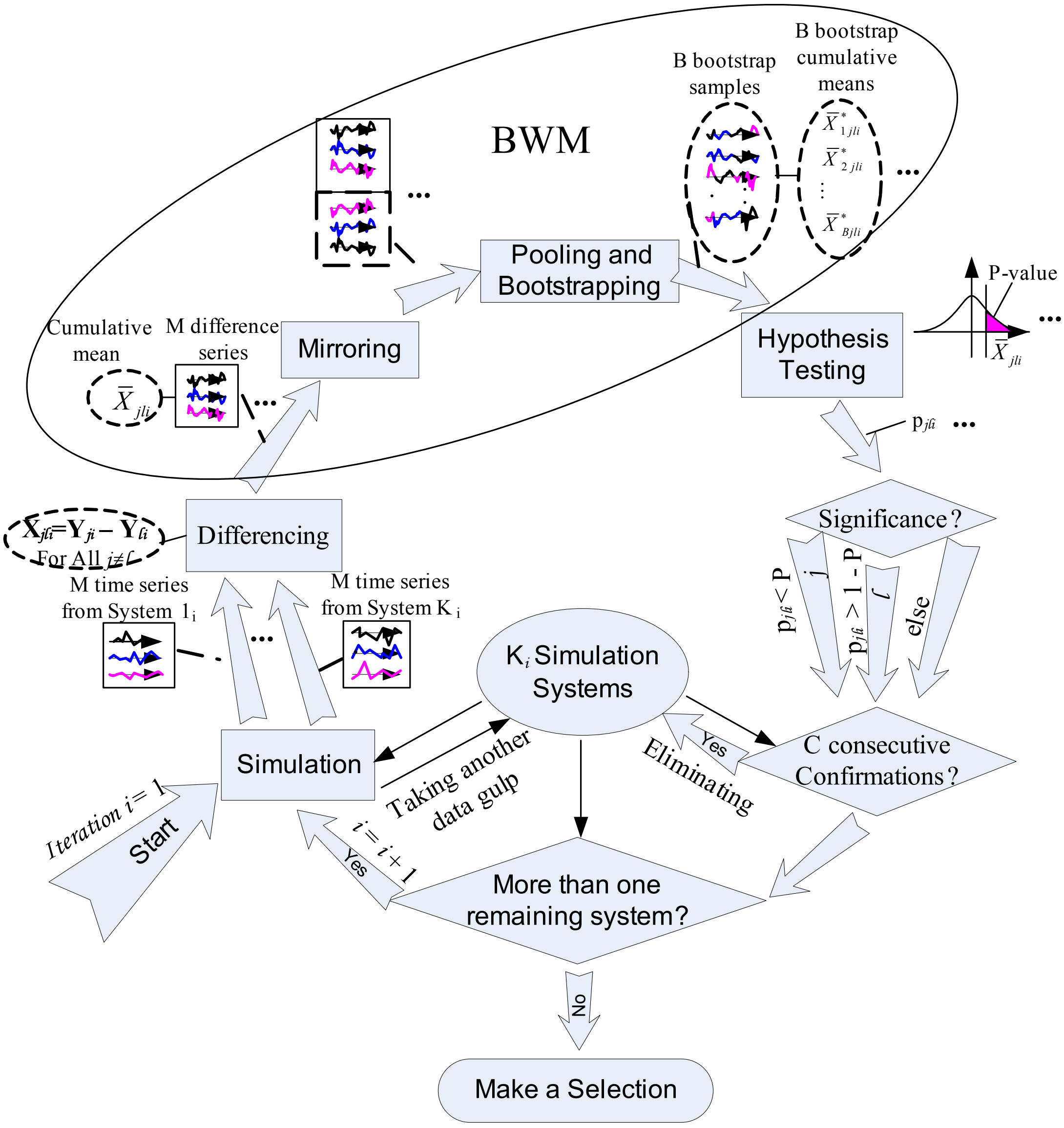

Figure 1: The Flow Chart of the New Procedure

rameter values, such as the choices of variance estimator and batch size, were varied as experimental factors, resulting in tables with multiple results. In the following section, we select for citation those results that were best by either PCS or number of observations, which always derived from different parameter choices.

Our experiments used the same test systems and configurations as GKMN. One difference, which we believe is not significant, is that we used the NAG C random number generator instead of the one in GKMN. We treated the parameters of the new method as experimental factors: the number of short independent replications (*M*), the number of consecutive rejections required before declaring a winner (*C*), and the p-value required to make a decision (*P*). We determined the length of the initial simulation runs n_0, the data gulp size, *g*, and the number of bootstrap replications, *B*, using only casual empirical analysis without attempting to optimize these parameters. For each combination of factor levels, we created *R* = 1000 independent replications.

5 RESULTS

The results of the BWM experiments are presented in Tables 1-4, which correspond to the order of results in GKMN. The tables have a common format. The BWM results are reported for various combinations of the BWM parameters *P* = p-value allowing rejection of the hypothesis test of identical system performance, *M* = number of independent simulation replications, and *C* = number of consecutive rejections of the null hypothesis before a decision is final. Two results are reported for each combination of parameter values: the sample average number of simulation observations required before a selection decision was made, and the sample proportion of trials in which the best system was properly identified (PCS). Each pair of results is based on *R* = 1000 independent replications of the experiment. The estimates are expressed in the form of 95% confidence intervals for the true mean and proportion, respectively. In each table, some of the cells are shaded; these shaded cells contain results that dominate both of the two best results for KN++, which are shown below the BWM results in each table.

Table 1 shows results for identifying the best of ten AR(1) processes with means offset in steps (MDM configuration). The best KN++ results achieved a 94.6% PCS after an average of 15,200 simulation observations and a 98.9% PCS after an average of 34,900 observations. BWM dominated KN++ for many parameter combinations. For example, with *P* = 0.05, *M* = 1, and *C* = 3, BWM achieved a 99.5% PCS after an average of 7,600 observations.

Table 2 shows results for identifying the best of ten MA(1) process in the MDM configuration. Here too BWM dominated KN++. For instance, whereas KN++ achieved a 98.8% PCS after an average of 1,700 observations, BWM reached a 99.7% PCS after 560 observations.

Table 1: Average Number of Observations ($\times 10^4$) and Estimated Probability of Correct Selection when K = 10 AR(1) Processes with ϕ = 0.9 in MDM Configuration

BWM

±95%CI's

P	*M*	*C*		
		1	2	3
0.05	1	0.46±0.03 94.6%±1.4%	0.65±0.03 98.1%±0.8%	0.76±0.04 99.5%±0.4%
	2	0.58±0.03 98.4%±0.8%	0.82±0.04 99.8%±0.3%	1.01±0.04 99.9%±0.2%
	4	0.72±0.04 99.4%±0.5%	1.04±0.04 99.8%±0.3%	1.29±0.04 100%±0%

g=500 n_0=500 *B*=100 *R*=1000

KN++ (GKMN Tables 3 and 4)

Summary (Var estimator A)	
Best #Obs	1.52 94.6%
Best PCS	3.49 98.9%

Table 2: Average Number of Observations ($\times 10^4$) and Estimated Probability of Correct Selection when K = 10 MA(1) Processes with θ = 0.9 in MDM Configuration

BWM

±95%CI's

P	*M*	*C*		
		1	2	3
0.05	1	0.037±0.002 98.9%±0.6%	0.056±0.002 99.7%±0.3%	0.073±0.003 100%±0%
	2	0.049±0.002 99.8%±0.3%	0.075±0.002 100%±0%	0.097±0.003 100%±0%
	4	0.065±0.002 100%±0%	0.108±0.002 100%±0%	0.152±0.003 100%±0%

g=100 n_0=100 *B*=100 *R*=1000

KN++ (GKMN Tables 5 and 6)

Summary (Var estimator A)	
Best #Obs	0.17 98.8%
Best PCS	0.30 99.5%

Table 3 shows results for the more difficult problem of selecting the best of five heavily loaded M/M/1 queues with offset means (MDM configuration). The computational savings with BWM were substantial. KN++ needed an average of 619,000 observations to attain a 96.7% PCS, but BWM needed only 167,000 observations on average to achieve the same PCS.

Table 3: Average Number of Observations ($\times 10^5$) and Estimated Probability of Correct Selection when K = 5 M/M/1 Queues with $\rho \geq$ 0.9 in MDM Configuration

BWM

±95%CI's

P	*M*	*C*		
		1	2	3
0.025	1	0.52±0.04 65.6%±2.9%	0.98±0.04 84.9%±2.2%	1.16±0.04 92.8%±1.6%
	2	0.96±0.07 79.6%±2.5%	1.50±0.08 92.7%±1.6%	1.67±0.08 96.6%±1.1%
	4	1.45±0.10 89.0%±1.9%	2.02±0.11 96.8%±1.1%	2.23±0.12 99.1%±0.6%
	6	1.64±0.11 91.6%±1.7%	2.05±0.12 97.3%±1.0%	2.35±0.12 99.1%±0.6%
	8	1.70±0.11 94.1%±1.5%	2.30±0.12 99.1%±0.6%	2.51±0.13 99.8%±0.3%
	10	1.66±0.11 95.3%±1.3%	2.30±0.12 98.8%±0.7%	2.76±0.14 99.8%±0.3%
0.05	1	0.31±0.04 56.4%±3.1%	0.75±0.05 76.2%±2.6%	0.99±0.06 87.6%±2.0%
	2	0.49±0.04 66.4%±2.9%	0.85±0.04 87.4%±2.1%	1.00±0.04 93.0%±1.6%
	4	0.90±0.08 78.7%±2.5%	1.33±0.09 92.3%±1.7%	1.74±0.11 97.0%±1.1%
	6	0.93±0.07 86.5%±2.1%	1.49±0.09 96.3%±1.2%	1.75±0.09 98.3%±0.8%
	8	1.05±0.08 86.9%±2.1%	1.62±0.10 97.2%±1.0%	1.88±0.10 99.1%±0.6%
	10	1.14±0.09 89.5%±1.9%	1.73±0.10 98.2%±0.8%	1.95±0.10 99.2%±0.6%

g=1000 n_0=1000 *B*=100 *R*=1000

KN++ (GKMN Tables 7 and 8)

Summary (Var estimator A)	
Best #Obs	2.49 90.9%
Best PCS	6.19 96.7%

Table 4 shows results for identifying the best of ten heavily loaded M/M/1 queues where one is offset from the other nine (SC configuration). KN++ required an average of 107,500 observations to reach a PCS of 95.2%, whereas BWM achieved the same PCS with only 20,200 observations on average

Table 4: Average Number of Observations ($\times 10^4$) and Estimated Probability of Correct Selection when K = 10 M/M/1 Queues with $\rho \geq$ 0.6 in SC Configuration

BWM

±95%CI's

P	*M*	*C*		
		1	2	3
0.05	1	2.13±0.15 85.3%±2.2%	2.30±0.11 96.0%±1.2%	2.51±0.09 98.4%±0.8%
	2	2.29±0.14 90.6%±1.8%	2.35±0.08 99.1%±0.6%	2.73±0.08 99.9%±0.2%
	4	2.39±0.15 98.1%±0.8%	2.89±0.09 99.9%±0.2%	3.41±0.08 100%±0%
	6	2.60±0.18 98.7%±0.7%	3.19±0.08 100%±0%	3.92±0.08 100%±0%
	8	2.57±0.11 99.6%±0.4%	3.56±0.08 100%±0%	4.47±0.09 100%±0%
	10	2.64±0.08 100%±0%	3.92±0.09 100%±0%	4.94±0.09 100%±0%
0.1	1	1.30±0.08 70.7%±2.8%	1.71±0.08 87.8%±2.0%	1.89±0.07 94.9%±1.4%
	2	1.73±0.13 84.2%±2.3%	2.02±0.10 96.2%±1.2%	2.20±0.07 98.9%±0.6%
	4	2.05±0.18 94.4%±1.4%	2.45±0.11 98.9%±0.6%	2.83±0.08 99.9%±0.2%
	6	2.28±0.21 96.6%±1.1%	2.69±0.08 100%±0%	3.43±0.09 99.9%±0.2%
	8	2.12±0.13 97.7%±0.9%	3.05±0.07 100%±0%	3.82±0.07 100%±0%
	10	2.29±0.17 99.2%±0.6%	3.29±0.07 100%±0%	4.42±0.07 100%±0%

g=1000 n_0=1000 *B*=100 *R*=1000

KN++ (GKMN Table 10)

Summary (B_3 column)	
Best #Obs	6.96 89.0%
Best PCS	10.75 95.2%

6 CONCLUSIONS

The BWM method compared favorably to the KN++ method, requiring fewer simulation observations for the same or better PCS. Specifically, BWM required only about 18-30% (AR: 0.76/3.49; MA: 0.056/0.30; M/M/1: 1.67/6.19 and 2.02/10.75) as many observations in all test cases. These gains were made without extensive tuning of the BWM parameters.

The price of this reduction in simulation observations is the overhead of bootstrapping. This overhead is negligible compared to the time required to run a simulation model of a large system. Once the simulation observations are in hand, the bootstrap simply rearranges and resamples from them without requiring new execution of the costly simulation code.

We can relate this work to the list of unresolved issues listed at the end of GKMN. First, the problem of initialization bias is ignored by BWM, apparently with impunity. It would be reassuring to understand the limits of this robustness. The sequential nature of the algorithm means that the extent of dependence on transient observations varies. If the difference between the mean performance of two alternatives is large, most or all of the decision may be based on transient data. If the difference is subtle, more observations will be required, and the transient phase may be a small proportion of the total dataset. Second, the need to obtain sufficient initial data to get a variance estimator with chi-squared distribution is obviated in the BWM method by a combination of multiple replications and the bootstrap. Third, the difficulty of analyzing the use of common random numbers (CRN) remains an interesting theoretical problem. To reproduce the GKMN results, we did not use CRN in our experiments, but there is no reason why this cannot be done.

Much work remains to better understand, optimize and justify the BWM method. But these early results are encouraging, since speedups by factors of four or more promise to be very helpful in practice. Further research is ongoing to develop guidelines for parameter settings, to extend the empirical investigation to more complex systems not studied in GKMN, and to better understand the theoretical properties of the new bootstrap method.

In particular, we need to identify situations where the new approach could be confounded by a reversal in the rankings developed in the transient phase when the systems reach steady state. It is necessary but not sufficient to rule out most kinds of time-varying systems (e.g., consider a comparison between a queueing system that adds servers in mid-day against a system that adds them at the end of the day). To the extent that averages computed from transient results have high variance, this problem may be self-limiting, but further work on this topic is in order.

ACKNOWLEDGMENTS

Seong-Hee Kim generously shared her computer codes for the KN++ method and GKMN experiments. Dave Goldsman offered encouragement at an early stage toward development of new approaches to simulation ranking and selection.

REFERENCES

Goldsman, D., S.-H. Kim, W. S. Marshall and B. L. Nelson. 2002. Ranking and selection for steady-state simulation: Procedures and perspectives. INFORMS Journal on Computing 14: 2-19.

Kim, S.-H. and B. L. Nelson. 2001. A fully sequential procedure for indifference-zone selection in simulation. ACM Transactions on Modeling and Computer Simulation. 11(3): 251-273.

Kim, S.-H. and B. L. Nelson. 2003. Selecting the best system: Theory and methods. Proceedings of the 2003 Winter Simulation Conference. S. Chick, P. J. Sanchez, D. Ferrin and D. J. Morrice, eds. 101-112. New Orleans, Louisiana: Institute of Electrical and Electronics Engineers

Law, A. M. and W.D. Kelton. 2000. Simulation Modeling and Analysis, 3rd ed. McGraw-Hill.

Park, D. and T. R. Willemain. 1999. The threshold bootstrap and threshold jackknife, Computational Statistics and Data Analysis 31: 187-202.

Park, D., Y. B. Kim, K. Shin and T. R. Willemain. 2001. Simulation output analysis using the threshold bootstrap. European Journal of Operational Research 134: 17-28.

Sheth-Voss, P. A., J. Haddock and T. R. Willemain. 1996. Estimating steady state mean from short transient simulations. Proceedings of the 1996 Winter Simulation Conference. J. M. Charnes, D. J. Morrice, D. T. Brunner and J. J. Swain, eds. 222-229. Coronado, California: Institute of Electrical and Electronics Engineers.

Sheth-Voss, P. A., T. R. Willemain and J. Haddock. Forthcoming. Estimating the steady-state mean from short transient simulations. European Journal of Operational Research.

AUTHOR BIOGRAPHIES

HUAIYU HARRY MA is a Ph.D. candidate in the Department of Decision Sciences and Engineering Systems at Rensselaer Polytechnic Institute. He holds the BS and MS degrees in Mechanical Engineering from Tsinghua University.

THOMAS R. WILLEMAIN is Professor of Decision Sciences and Engineering Systems at Rensselaer Polytechnic Institute and co-founder and Senior Vice President of Smart Software, Inc.