Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

SIMULATING M/G/1 QUEUES WITH HEAVY-TAILED SERVICE

John C. Sees, Jr.

6001 Goethals Road, Suite 102
Center for Army Analysis
Fort Belvoir, VA 22060-5230, U.S.A.

John F. Shortle

Systems Engineering & Operations Research Department
George Mason University
Fairfax, VA 22030, U.S.A.

ABSTRACT

We examine the performance and accuracy of simulating M/G/1 queues when the service time is Pareto distributed with shape parameter, alpha, between one and three. Two applications of this problem are in insurance risk and telecommunications. When 2 < alpha <= 3, the theoretical distribution of the sample averages of the queue waiting times is a stable distribution. When alpha <= 2, the mean waiting time does not exist. We provide a modified quantile simulation method, which is able to solve harder problems than existing methods; in addition, it requires less memory, and allows the user to emphasize accuracy or execution time. We also give numerical examples for other heavy-tailed distributions, such as the lognormal.

1 INTRODUCTION

This paper discusses the performance of M/G/1 (G~Pareto) queues over a range of Pareto shape parameter values (α-values) and provides insight on overcoming the difficulties of simulating these queues. We provide a modified quantile simulation method, which is able to solve harder problems than existing methods; in addition, it requires less memory, and allows the user to emphasize accuracy or execution time.

For our problem, there are two main difficulties: simulating the queue waiting time distribution and calculating the accuracy of our results. Simulating queueing systems with Pareto service times is hard. The value of α significantly affects the rate of convergence to steady state. This leads to simulations requiring many observations and long run times. Although large sample sizes and long execution times are typical of many interesting problems reported in the literature, simply simulating the system longer may not work. This is because for low α-values, simulation outcome statistics converge very slowly.

Another difficulty is that the distribution of the sample mean queue waiting times is not the same for all α-values. Depending on the parameter value, sample mean queue waiting times converge to different distributions or do not converge at all. For $3 \le \alpha$, the distribution of these sample means converges to a normal distribution. In these cases, the Central Limit Theorem and traditional confidence interval calculations apply. For $2 < \alpha < 3$, the distribution of the sample averages of queue waiting times converges to a stable distribution. Here, traditional confidence intervals do not apply. For $\alpha \le 2$, the mean queue waiting time does not exist. We modify an existing quantile estimation method finding solutions across a broader range of parameters, requiring less memory, and giving users flexibility to make accuracy and execution time trade-offs.

Our M/P/1 queues have a single-server and the interarrival times and service times are independent with interarrival times exponentially distributed with mean $1/\lambda$ and service times Pareto distributed with mean $1/\mu$. We assume the capacity of the queue is infinite and the queueing discipline is first-come first-served. Harris (1968) showed that the Pareto distribution could be derived as a gamma mixture of exponentials leading to the distribution,

$$\Pr(X \le x) = 1 - \frac{\theta^{\alpha}}{(\theta + x)^{\alpha}}, \quad (x \ge 0 \ \alpha > 0)$$

where α is the shape parameter that measures the tail-thickness of the distribution and θ is a shift parameter. The density is given by,

$$f(x) = \frac{\alpha}{(\theta + x)^{\alpha+1}}, \ (x \ge 0 \ \alpha > 0).$$

Although there are several forms of this distribution, we use a one-term Pareto defined over the nonnegative real numbers with the shift parameter value $\theta = 1$.

A distribution has a power-tail if the tail of the distribution decays geometrically in the limit. That is,

$$\overline{F}(x) \sim cx^{-a} \qquad (1)$$

where c is a constant and $a(x) \sim b(x)$ means $\lim_{x\to\infty} a(x)/b(x) = 1$. The Pareto distribution has a power-tail $a = \alpha$.

Cohen (1982) provides additional information by proving an important relationship between the moments of the service-time distribution and the moments of the queue waiting-time distribution for an M/G/1 queue. He shows that the distribution of the queue waiting times has one less moment than the service-time distribution. For Pareto service-time distributions with $2 < \alpha \le 3$, both the mean and variance of the service-time distribution exist, but only the mean of the queue waiting-time distribution exists. Of particular importance to our research is that for Pareto service-time distributions with $\alpha \le 2$, the mean queue waiting time does not exist.

The tail asymptotics of the M/G/1 queue are well known. If G is heavy-tailed, then (e.g., Sigman 1999)

$$\Pr(W > x) \sim \frac{\rho}{(1-\rho)} G_e(x),$$

where $G_e(x)$ is the equilibrium distribution of G. For example, if G is a Pareto distribution, then it follows that:

$$\Pr(W > x) \sim \frac{\lambda}{(1-\rho)(\alpha-1)} \frac{1}{x^{\alpha-1}}. \qquad (2)$$

A well-known random variable that describes the convergence of sample averages to their expectation is,

$$Z_n = \sqrt{n}(A_n - \phi), \qquad (3)$$

where $A_n = \frac{1}{n}\sum_{i=1}^{n} X_i$.

From the Central Limit Theorem, $\lim_{n\to\infty} Z_n$ is normally distributed N $(0, \sigma^2)$.

If the variance is infinite, then we cannot use this result. Feller, 1971, defined the general case of (3) as

$$Z_n = n^{\kappa}(A_n - \phi), \qquad (4)$$

where $\kappa = 1\text{-}1/\eta$, η is the tail index of the distribution, and ϕ is the true mean of the distribution. We use η instead of the more traditional α in this definition to avoid confusion with the Pareto shape parameter α. These two parameters are related with $\eta = \alpha - 1$, for $2 < \alpha \le 3$. When the X_is follow a power law (equation (1), with $a = \eta$) and $1 < \eta \le 2$, then $\lim_{n\to\infty} Z_n$ is a stable distribution $S1(\eta, \beta, \gamma, \delta)$.

Chen and Kelton (1999) seek to improve the large storage and processing costs associated with fixed sample-sized, quantile estimation. Their method has an iteration phase and a replication phase and requires multiple sampling and sorting. However unlike fixed sample-sized, quantile estimation, Chen and Kelton's (1999) algorithm only keeps a small number of the sample values while counting all observations. It therefore has fewer observations to sort. The method brackets the quantile estimate by a pair of bounds and at each iteration increases the sample size. By taking an initial sample large enough to capture the desired quantile and controlling the constriction of the bounds, their method obtains the shape and values of the distribution near the desired quantile. They then use replication to find the final quantile estimate and confidence interval. We improve their algorithm to solve additional problems, while requiring less memory, and allowing the user to emphasize accuracy or execution time.

When sampling outcomes from any system, their independence becomes important. A characteristic of queueing problems is that successive sampled queue waiting times are not independent. A well-known recursion, Lindley's formula (Gross and Harris 1998), relates the queue waiting time (W_q) of the n^{th} customer to the $n+1^{st}$ customer

$$W_{q(n+1)} = \text{Max}(0, W_{q(n)} + S_n - T_n), \qquad (5)$$

where S_n is the service-time of the n^{th} customer and T_n is the interarrival time between the n^{th} and the $n+1^{st}$ customer. Clearly these waiting times are not independent unless each arriving customer finds the queue empty. Our results address this dependence.

2 A MODIFIED METHOD TO SIMULATE QUANTILES

Quantile estimation depends on point estimators obtained by order statistics. Let $X_1, X_2, \cdots X_n$ be sampled random variables from a continuous distribution $F(x)$ or density function $f(x)$. Let x_p denote the p^{th} quantile having the property, $F(x_p) = \Pr(X \le x_p) = p$ $(0 < p < 1)$.

If $Y_1, Y_2, \cdots Y_n$ are the order statistics corresponding to the X_i's from n observations, then a point estimator for x_p based on the order statistics is the sample p^{th} quantile, $\hat{x}_p = y_{\lceil np \rceil}$, where $\lceil z \rceil$ denotes the integer ceiling (round-up) of the real number z, and $y_{\lceil np \rceil}$ is the np^{th} smallest of $Y_1, Y_2, \cdots Y_n$.

Statistically, the queue waiting time process of simulations is nonstationary and autocorrelated. These characteristics usually prevent using traditional statistics. However, we can use order statistics to overcome the lack of independence if the random variables are derived from a ϕ-mixing process. A ϕ-mixing process is a stochastic process where a distant future event is approximately independent of its present and past events (Billingsley 1999).

Quantile estimates from ϕ-mixing processes are asymptotically unbiased and may be averaged. Sen (1972)

showed that quantile estimates based on order statistics have a limiting normal distribution if three conditions are satisfied:

1. The sampled process $\{X_i\}$ is ϕ-mixing.
2. The cumulative distribution function $F(x)$ is absolutely continuous.
3. The density function $f(x)$ is finite, positive, and absolutely continuous for all $x = F^{-1}(t)$ and $0 < t < 1$.

It is not hard to see that successive queue waiting times from a M/P/1 queue is a ϕ-mixing process for utilizations less than one ($\rho < 1$). From Gross and Harris (1998), the probability that the system is idle equals 1-ρ. Therefore, with probability one, at some time the queue will empty. When the queue is empty, the queue waiting time is zero, and from Lindley's equation (5), the preceding waiting times are independent of the following waiting times. When the probability mass at zero is removed, the M/P/1 queue's continuous arrival-time and service-time densities lead to the distribution of positive, real queue waiting times meeting conditions two and three from Sen.

Our quantile estimation method determines the required simulation run length using an iterative process and is a modification of a method developed by Chen and Kelton (CK)(1999). Figure 1 illustrates the initial sampling and the iteration flow of the method.

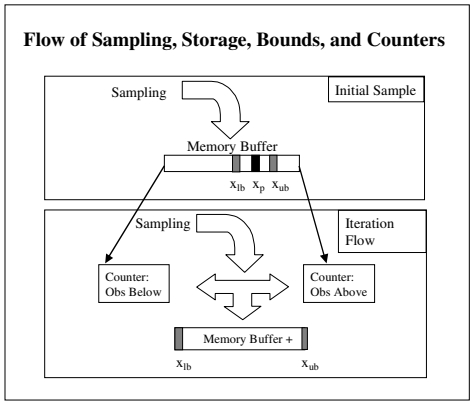

Figure 1: Initial Sampling and Iterations for the CK Method

The algorithm begins with an initial sample of random variables sufficiently large to include the true but unknown desired quantile. For clarity, we designate this array of values as the *memory buffer*. From this sample, the method calculates an intermediate quantile estimate $\hat{x}_p$ and then encloses the unknown true p^{th} quantile by calculating upper and lower bounds (x_{lb} and x_{ub}, respectively). The bounds are located by position relative to $\hat{x}_p$. Before beginning the iteration phase, the number of observations below the lower bound position is added to a counter (*obs_below*) and the number of observations above the upper bound position is added to another counter (*obs_above*).

The method then performs several iterations (Figure 1, Iteration Flow), where only observations between x_{lb} and x_{ub} are stored. If the sampled random variables are not between x_{lb} and x_{ub}, the method increments upper or lower counters. For each iteration, the method increases the memory buffer, resamples to fill the buffer, increments counters, and establishes new bounds. The bounds calculated from the previous iteration become the largest and smallest values in the memory buffer for the current iteration. The program uses these bounds to screen sampled random variables for inclusion in the memory buffer, and as the method progresses, these bounds contract about the true but unknown quantile.

After some empirical stopping criteria are met, the method saves upper and lower bounds, the size of the memory buffer and sets the counters to zero. It then samples until the memory buffer is filled, keeping track of the number of samples that lie outside the bounds. This process (the replication phase) is repeated up to seven times to obtain confidence intervals.

We investigated several modifications to the CK method. We call our method the Modified-Chen-Kelton (MCK) method. The intent of our modifications was to increase the problem space over which quantile estimates are possible, to generate unbiased quantile estimates, and to maintain or improve the processing time and accuracy.

Our modifications construct the upper and lower bounds at different rates, relax the stopping criteria in the iteration phase, and eliminate a possible quantile estimator bias in the replication phase.

First, we wanted to constrict the lower and upper bounds as quickly as possible around the true but unknown quantile value. This would decrease the method's time in the iteration phase. M/P/1 queue waiting times have a preponderance of small observations. We can use this information to constrict the lower bound faster than the upper bound. By applying different weights to find the two bounds, we can independently converge on the true quantile value from below and above and end the iteration phase sooner than the CK method.

Obtaining close bounds in a short period of time is in conflict with one of the heuristic stopping criteria of the CK method. Therefore, our method eliminates the stopping criteria requiring no monotonic increase or decrease in the quantile estimate based on the previous four iterations. The reason we do this is that sometimes our method has already obtained candidate bounds before the fourth iteration. Performing additional iterations to perform this check unnecessarily uses computation time in the iteration phase. Our results show that the processing time used to achieve this stopping criteria is better spent in the replication phase, where each replication contributes to reducing the width of the confidence interval.

In the CK method's replication phase, the quantile estimate is set to the nearest bound value if it is outside of the array. This biases the estimate. In our modification, we linearly extrapolate from the memory buffer data and bound positions to obtain a quantile estimate when this circumstance arises.

Both methods use a precision criterion (designated *EPS > 0*). The methods use this factor during the iteration phase to check convergence of the distribution and during the replication phase as a decision criteria to make additional replications. The role of the *EPS* is to affect the setting of the confidence interval. Setting *EPS* too low will cause numerous iterations. This can lead to the algorithm terminating due to insufficient memory. Setting *EPS* too high results in large confidence intervals. Our experiments have shown us that we obtain confidence intervals that are an order of magnitude lower than the *EPS* setting, and that for a fixed desired confidence interval half-width, as α decreases and/or the desired quantile increases, the EPS value should increase. For our example problem (α = 1.5 and we desire an estimate of the 90th quantile), if we want a confidence interval half-width of approximately one, then we set EPS to approximately ten.

Figure 2 shows the lower and upper bounds and the quantile estimates of both methods as the algorithms progress. For this illustration, we set the upper weight to 0.51 and the lower weight to 0.24 for comparison with the CK method. The MCK method meets the stopping criteria after four iterations and the final estimate is based on five replications. The 90th quantile estimates ($\hat{x}_{90}$), 90% confidence interval (CI) half-widths, and execution times (X-time) for the MCK and CK methods are $\hat{x}_{90}$ = 1085.1, CI = 1.07, X-time = 4.9 hours; and $\hat{x}_{90}$ =1085.4, CI = 1.31, X-time = 5.17 hours respectively. Thus, for this example, our modifications provide a tighter CI half-width in less time.

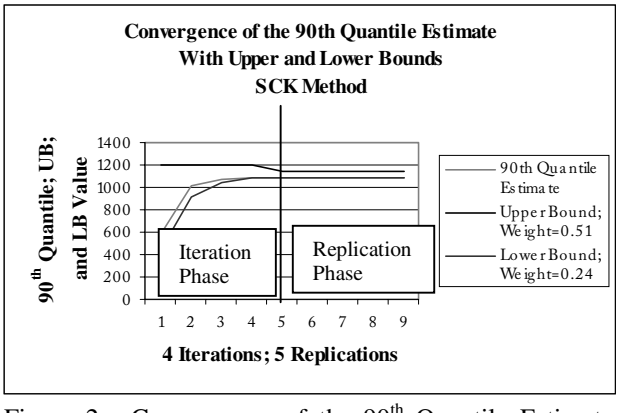

Figure 2: Convergence of the 90th Quantile Estimate with Upper and Lower Bounds (MCK Method); α = 1.5; Initial Bounds Percentage = +/- 0.1**p*

3 NUMERICAL RESULTS

We executed numerous simulations and collected data on each method's memory storage requirements, 90% CI half-widths, average sample sizes, and execution times. Our method requires less memory and generally has a narrower CI for a given sample size. Figures 3-4 plot our results. Figure 3 shows the significant advantage of the MCK method's lower memory buffer requirements for all cases considered. Since the MCK method uses less memory, we can solve harder problems and/or estimate multiple quantiles in one simulation run. Later, we show that for low α–values or high quantile estimates, the CK method fails, where MCK does not.

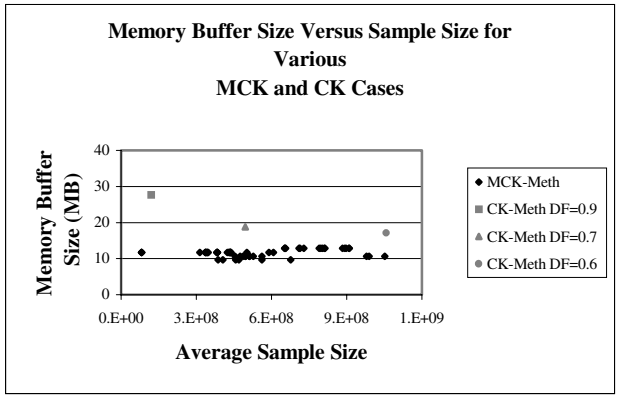

Figure 3: Memory Buffer Size Versus Sample Size for Various MCK and CK Cases

Figure 4 illustrates the MCK method's narrower 90% confidence interval half-widths. The preponderance of these values are better than those obtained by the CK method. This result is true even though the CK method biases its quantile estimates by restricting them to values within the memory buffer. Considering our performance criteria of memory storage, execution time and CI half-width, our method provides a distinct improvement.

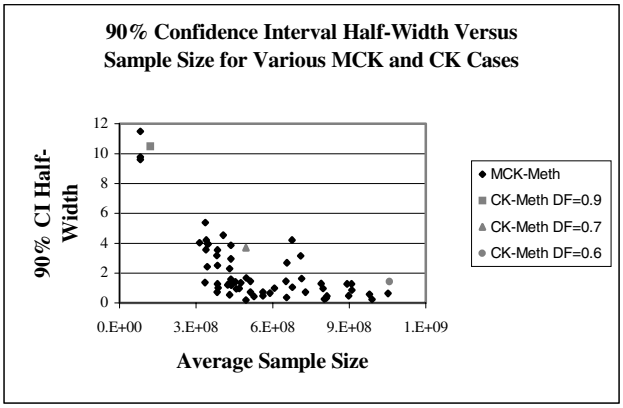

Figure 4: 90% Confidence Interval Half-Width Versus Sample Size for Various MCK and CK Cases

We augment these results with guidelines for weight selection. As the Pareto shape parameter decreases and/or the desired quantile gets larger, increase the upper weight and decrease the lower weight. If the priority is a short execution time, increase the difference between the weights and specify less precision. If increased accuracy is the goal, decrease the difference between the weights and specify more precision.

We have shown that our method has advantages over the CK method for one specific case (α = 1.5, 90th quantile). Now we argue that these advantages hold over a wider class of problems. In particular, the advantages become greater for higher quantiles. In addition, for some problems, the CK method abnormally terminates due to insufficient memory, where the MCK does not.

We designed and conducted a number of experiments to compare the performance between our method and the CK method. We varied the Pareto shape parameter from 1.1 to 1.9 in 0.1 increments and simulated to obtain the 90th, 95th, and 99th quantiles. For these experiments, we kept constant the queue utilization, the precision, the initial bounds, and the initial memory buffer array size. We used our weight-setting rules to set the upper and lower weights for the MCK method. The CK method prescribes no rules for parameter selection, but our experience suggested that we select large values to capture the true but unknown quantile. In order to provide a close comparison, we experimented by incrementally lowering these CK values, but often the quantile estimate was outside of the memory buffer and the simulation terminated.

Table 1 shows the quantile estimates for each case. While the MCK method solves problems throughout the range, the CK method cannot for α–values less than 1.5 for any quantile. This is because the CK method exceeds the available computer memory. The MCK method finds solutions using all α–values for the 90th and 95th quantiles, and for the 99th quantile when the α–value is greater or equal to 1.7. For α–values smaller than 1.7, the MCK method exceeded the available memory for estimating the 99th quantile.

Table 1: CK and MCK Method Solution Performance (α-values: 1.1-1.9; Quantiles: 90th, 95th, and 99th)

Method:	CK	MCK	CK	MCK	CK	MCK
α-value	90th Quantile		95th Quantile		99th Quantile	
1.1	X	352,549	X	929,314	X	X
1.2	X	94,818	X	267,756	X	X
1.3	X	17,760	X	59,468	X	X
1.4	X	3,701	X	11,754	X	X
1.5	1,084	1,084	3,356	3,362	X	X
1.6	422	422	1,174	1,175	X	X
1.7	205	205	502	503	X	2,674
1.8	118	118	261	260	X	1,231
1.9	77	77	153	153	X	649

For α–values greater than or equal to 1.5, we can compare the two methods. Figure 5 shows the MCK method's improvement in memory usage over the CK method. Over these experiments, the MCK method achieved a 38% average fractional reduction in memory usage and a 42% average fractional CI half-width reduction compared to the CK method.

Next, we modified our experiments to increase the chance of both methods obtaining a solution. We did this by decreasing the *memory buffer increment factor* (from 10% to 1%) in the iteration phase (Chen and Kelton used a *memory buffer increment factor* of 10% in their research) and performed the experiments for the most difficult case (the 99th quantile). Both quantile estimation methods arrived at solutions, but the MCK had an average 83% narrower fractional CI half-width and performed on average 56% faster than the CK method. These results were based on four cases, α–values 1.1 to 1.4. Thus, we see significant benefits from using the MCK method for low α–values and quantile estimates far into the tail of M/P/1 queue waiting-time distributions.

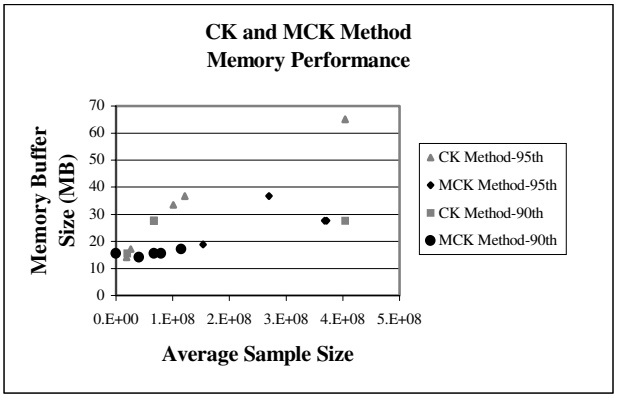

Figure 5. CK and MCK Memory Performance (α-values: 1.5-1.9, 95th Quantile)

Finally, Table 2 illustrates our method applied to the log-normal distribution (logN) and M/M/1 queue waiting times.

Table 2: MCK (CK) Method Performance for Various Distributions

Case	90th Quantile	90th Quantile MCK (CK)	90% CI Half-Width MCK (CK)	Processing Time (Hours)/Buffer Size MCK (CK)
LogN Mean 0.901 Var 17.4	1.845	1.884 (2.115)	0.075 (0.145)	0.18/ 732,050 (0.276/ 974,359)
Pareto α=2.1 ρ=0.8	1.9936	1.99377 (1.99296)	0.0016 (.0017)	0.032/ 805,255 (0.029/ 805,255)
M/M/1 λ=0.75 ρ=0.75	8.06	8.05 (8.07)	0.012 (0.058)	0.496/ 133,100 (0.013/ 161,051)

In summary, the MCK quantile estimation method is an improvement over existing methods. It uses less memory and allows the user to adjust parameters for increased accuracy or shorter execution time. Our rules for weight selection reflect our insight into the method's operation and are easy to apply. The MCK method also performs well for test cases of known distributions, providing confidence for our quantile estimates of the unknown distribution of M/P/1 queue waiting times.

REFERENCES

Billingsley, P. 1999. *Convergence of Probability Measures*. 2nd ed., New York: John Wiley & Sons.

Chen, E. J. and W. D. Kelton. 1999. "Simulation-Based Estimation of Quantiles." *Proceedings of the 1999 WinterSimulation Conference*: 428-434.

Cohen, J. W. 1982. *The Single Server Queue*. Rev. ed. North-Holland, Amsterdam.

Feller, W. 1971. *An Introduction to Probability Theory and its Applications, Volume 2*. 2nd ed., New York: John Wiley & Sons.

Gross, D. and C. M. Harris. 1998. *Fundamentals of Queueing Theory*. 3rd ed., New York: John Wiley & Sons.

Harris, C. M. 1968. "The Pareto Distribution As A Queue Service Discipline." *Operations Research*. 16: 307-313.

Sen, P. K. 1972. "On the Bahadur Representation of Sample Quantiles for Sequences of ϕ-mixing Random Variables." *Journal of Multivariate Analysis*. 2: 77-95.

Sigman, K. 1999. "A primer on heavy-tailed distributions." *Queueing Systems*. 33: 261-275.

AUTHOR BIOGRAPHIES

JOHN C. SEES, JR. is an Army Lieutenant Colonel in the Mobilization and Deployment Division at the Center for Army Analysis, Fort Belvoir, VA. He earned a B.B.A. from Notre Dame in 1982, a M.S. in operations research from the University of Texas in 1992, and his Ph.D. in information technology from George Mason University in 2001. His interests include simulation modeling and analysis with applications in logistics and transportation systems design. His email address is <sees@caa.army.mil>.

JOHN SHORTLE is an assistant professor of Systems Engineering at George Mason University. He received a B.S. in mathematics from Harvey Mudd College in 1992 and a Ph.D. and M.S. in operations research from UC Berkeley in 1996. He worked for three years at U S WEST Advanced Technologies developing stochastic, queueing, and simulation models to optimize networks and operations. In 2000, he won the INFORMS Daniel H. Wagner Prize for excellence in Operations Research Practice. His research interests include simulation and queueing applications in telecommunications and air transportation. His email address is <jshortle@gmu.edu>.