Calhoun: The NPS Institutional Archive

DSpace Repository

Faculty and Researchers | Faculty and Researchers' Publications

2011

Agent Based Simulation Output Analysis

Schruben, Lee; Singham, Dashi

L. Schruben, D. Singham, "Agent based simulation output analysis," Proceedings of the 2011 Winter Simulation Conference, 10 p.
https://hdl.handle.net/10945/38582

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

Proceedings of the 2011 Winter Simulation Conference
S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.

AGENT BASED SIMULATION OUTPUT ANALYSIS

Lee Schruben

University of California, Berkeley
4141 Etcheverry Hall
Berkeley, CA 94720, USA

Dashi Singham

Naval Postgraduate School
1 University Circle
Monterey, CA 93943, USA

ABSTRACT

In most realistic simulations there are multiple outputs of interest and the overall performance of the system can only be estimated in terms of these multiple outputs. We propose a method that uses agent-based modeling to determine a truncation point to remove significant initialization bias. Mapping the output of multiple replications into agent paths that traverse the sample space helps determine when a near steady state has been reached. By viewing these paths in reversed time, qualitative and quantitative methods can be used to determine when the multivariate output is leaving its near-steady state regime as the paths coalesce back towards their common initialization state. The methodology is more efficient and general than typical approaches for finding a truncation point for scalar outputs of individual replicates. Artificial bootstrap-like re-sampling of simulation runs is proposed for expensive simulations to estimate system performance sensitivity.

1 INTRODUCTION

Flocking was introduced in Schruben and Singham (2010) as a method for generating simulation input from trace data. The initial motivation was to create a way to replicate simulations having the stochastic input properties of real system data. Simulation is used in critical decisions in almost all aspects of modern society, and data-driven simulations are often the only credible means of running models of large, complex systems.

The approach proposed integrates concepts from discrete event simulation, time series, copulas, and agent-based modeling; however, it is fundamentally different from current methods. The procedure involves simulating the flight paths of agent "boids" who flock around a path determined by the original data. Sampling the coordinates of each agent produces simulated time series with similar behavior to the real system. The procedure can mimic dependent, high-dimensional, and non-stationary time series with broad potential statistical applications beyond simulation input modeling. Simulation flocking can also be applied to the analysis of multivariate simulation output data. In this paper, we explore how flocking ideas can be used to analyze simulation output data, either to determine initialization bias in multivariate output, or to aid in bootstrapping for sensitivity analysis of output data.

Current simulation initialization methodologies almost exclusively deal with single variable output; an exception is in Schruben (1981), where batched T^2 statistics are used to identify complex initial transient behavior. As seen in the examples that follow, initialization bias in realistic simulations is due to complex interactions. Most academic papers use simple stationary time series examples (AR(1) models) or scalar output statistics like total system queue size to study initialization, starting in an empty and idle (or other convenient, also biasing state). Realistic production and service queueing network simulation runs are also started with some convenient loading (also commonly empty and idle). However, they also have fully-stocked supply chains and brand-new resources, etc. Simulation warm-up not only fills empty shelves and queues, but mixes resource down times. In this paper we propose using agent-based concepts

to look at multidimensional simulation output. This radically new approach is promising, as illustrated in the following examples, but requires more research.

2 CHALLENGING CURRENT NOTIONS OF SIMULATION INITIALIZATION

Three simple to understand systems with complex multivariate output behavior are examined in this section: one is a simple example whose behavior violates all current notions of simulation model initialization bias and does not have what is commonly thought of as a steady-state, one appears to have a steady-state, but complex initial behavior, and one is a textbook network example where one of the four outputs appears (over long periods of time) not to have a steady state, but apparently does. These simulation models are available free from sigmawiki.com

2.1 Virtual Queue

The following simple two-queue network illustrates initialization behavior very different from that assumed in most academic research. This model, suggested by Prof. Loo Hay Lee, is shown in Figure 1.

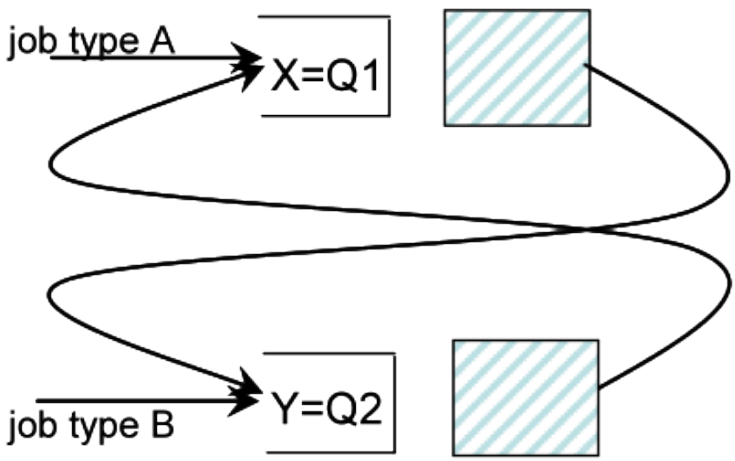

Figure 1: A two system queueing network.

This simple example exhibits complex dependencies among its state variables that is important when considering simulation initialization bias. Suppose we have n samples of the two queue lengths over time. The scatter plot of Q1 versus Q2 covers the two dimensional space in the left plot of in Figure 2. The same points can be joined sequentially, and this path is plotted in the right plot of Figure 2. This "connects the dots" in the left plot of Figure 2. The right plot is much more informative than the left one. The very unusual dynamic interactions between these two time series is further revealed when motion is added to the path in Figure 2, as in a video of a particle's movement in a unit cube. The two time series that generated these figures are in plotted in Figure 3.

The path in Figure 2 shows the inverse relationship between the two queues, and the magnitude of their peaks over time. One can look at the path in a variety of ways. In this research we employ the analog of snapshots of a bird's flight path within a cube. Such behavior has long been modeled by a class of agents called boids (Reynolds 1987). Many types of boids have been created to animate social behavior called "flocking" and are used in computer animations and movies (for example, in the movie Jurassic Park) as well as to look for emergent social behavior in groups. Flocking has also been used to visualize time series in order to find natural groupings among the data (Moere 2004). Here, boids are used to analyze output data.

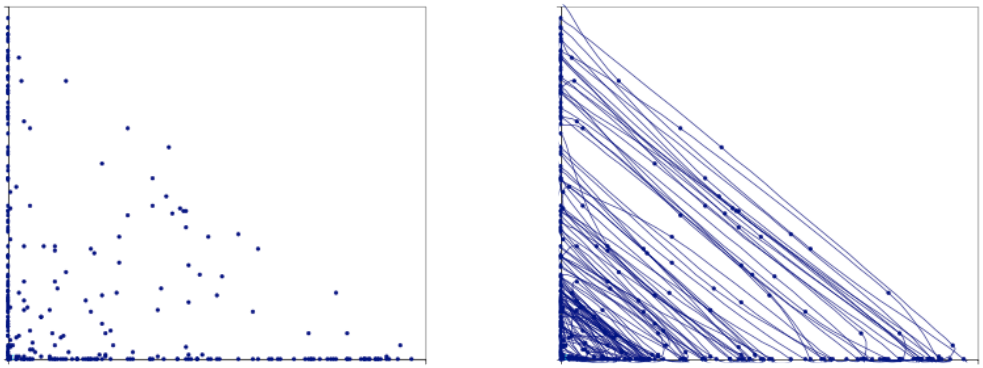

Figure 2: Scatter plot of a sample of Q1 and Q2 (left). Path for time series (Q1,Q2) (right).

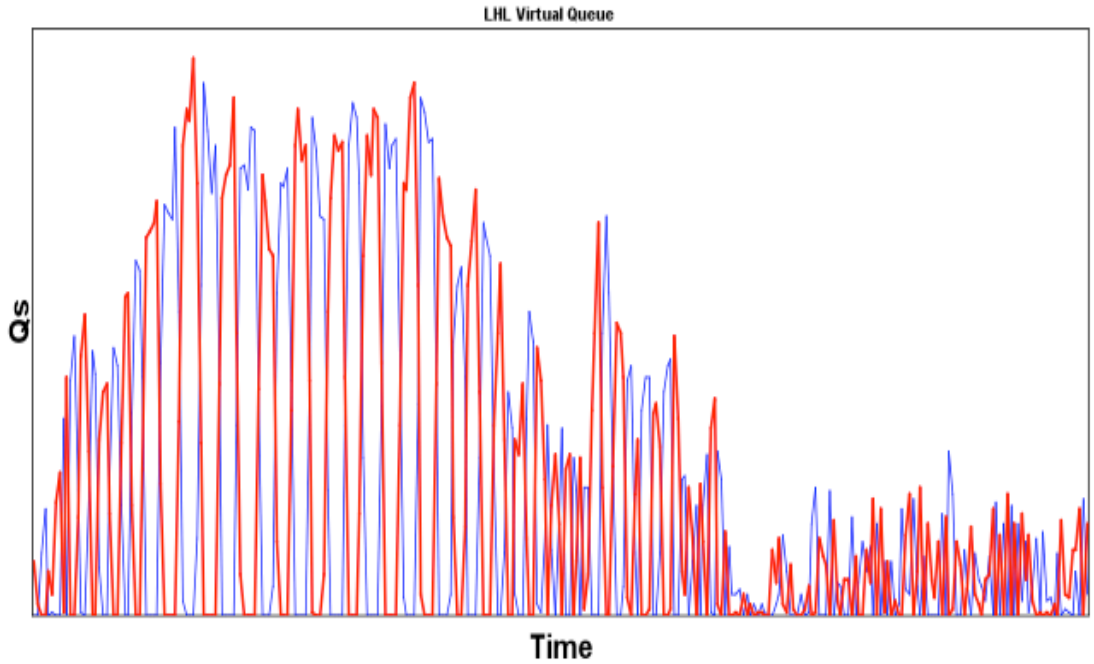

Figure 3: The two time series of Q1 and Q2 used to generate Figure 2.

Re-sampling is done by generating a flock of boids that follow a "leading" boid whose flight path is given by the coordinates of a multivariate time series. Modeling a prescribed path for a leading boid has been called "scripting" in the agent modeling literature. Simulated time series are generated as snapshots of the coordinates of individual members of the flock. The degree that a member of the flock is attracted to the leading boid reflects how much each simulated future will behave like the past. The flock size can be increased to generate as many replications as desired, without assuming stationarity or independence. Sensitivity analysis can also be performed, because flocking algorithms can be designed that control how much the follower boids deviate from the path of the leading boid.

2.2 Two Interacting Resources

The next model we consider is a simple tool bank with M machines and N workers. This example is a simplified version of the model in Schruben (1981), that still illustrates an unconventional initial transient and has what might be viewed as a conventional steady-state in the scalar output of total work in the process queue. The system being modeled is a set of parts presses with workers who do degating (trimming the flange) of finished parts. When the parts queue is too long a "start" light signals the workers, who as soon as they are free, degate the parts in the queue until they reach a "free" limit. Afterwards, they can take a break if there are no parts above their "start" limit. The objectives of the project were to determine M and W as well as set the degating limits so that a labor clause limiting average shift work time is not violated. (The current system has W=M and most of the time the workers were idle.) A schematic of the system is in Figure 4.

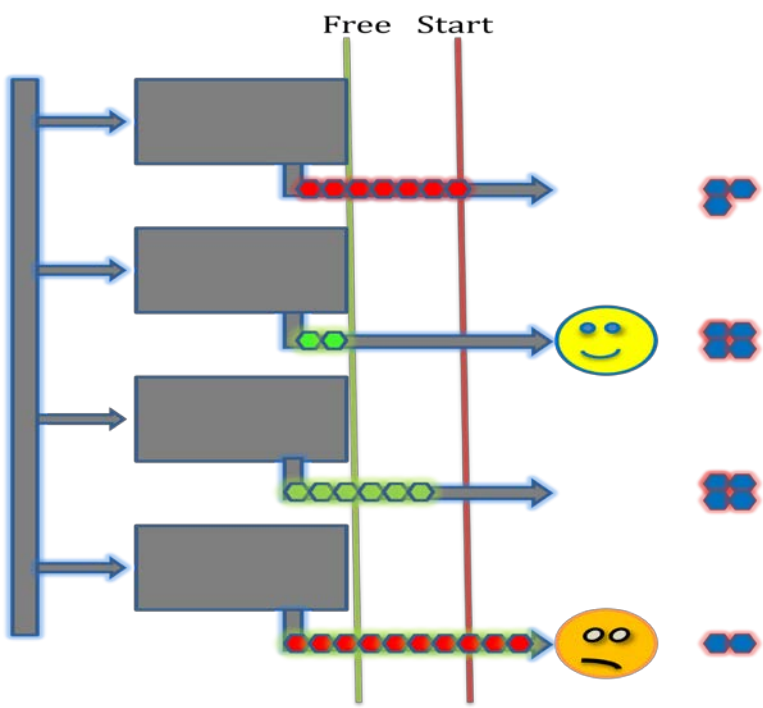

Figure 4: A tool bank model with M=4 machines and W=2 workers.

In order to see the effect of the initialization bias on the output, we can plot the values of the four different queues against each other. For a single replication of the simulation, we can create one boid that flies in a four-dimensional space, with each coordinate corresponding to the value of one of the queues at a particular point in time. Since we can only plot in two-dimensions for this paper, we present a matrix of plots for each of the possible two-queue combinations.

Figure 5 shows the presence of the initialization bias in the four queues. Schruben (1981) presents a method for dealing with initialization bias in multivariate output. Here, we show how a boid can be used to tie together jointly dependent series to determine visually what might be a good truncation point, as opposed to considering each series separately. We can see clearly how the four queues start at zero and move together into region where they appear to exhibit stationary behavior. Viewing multiple boids in reverse time and determining when they leave the stationary region and return to their starting conditions can help determine an appropriate truncation point.

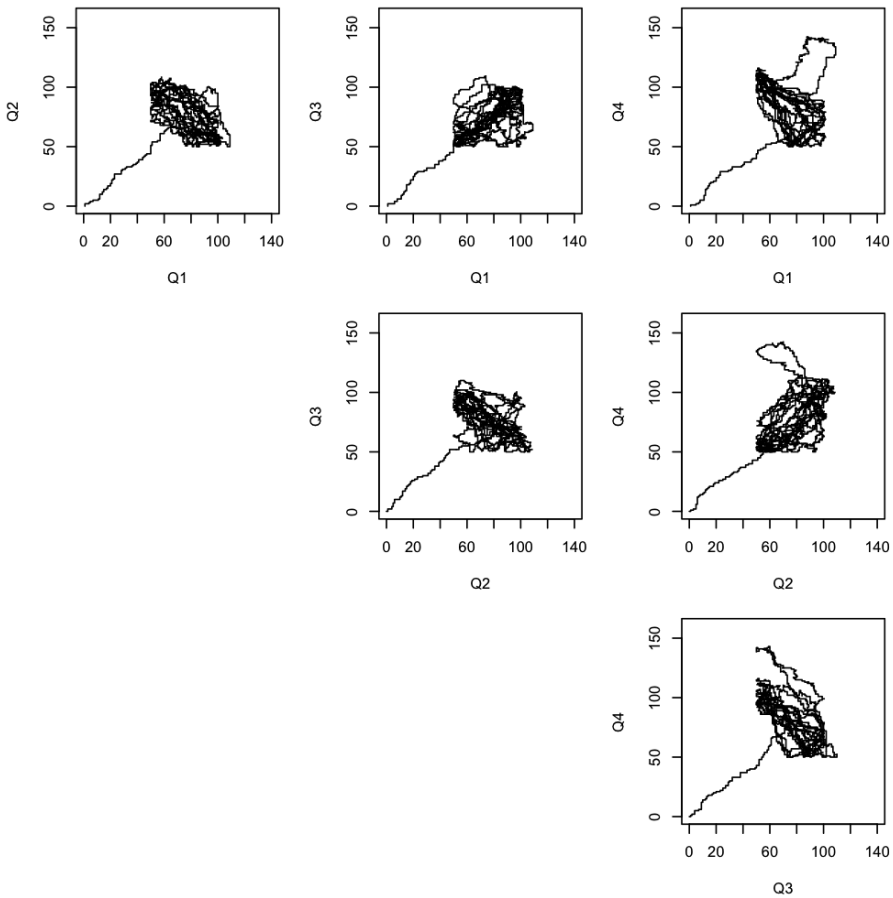

Figure 5: Plot of each queue against every other queue to visualize their joint behavior.

2.3 Textbook Production Network

This example is from the popular simulation textbook by Averill Law (Law 2007). It is a network of four multiple-server production stations. The output of the four queues is shown in Figure 6 for four independently seeded replications. It is difficult to determine a truncation point that is appropriate for all of the series. The red line in each plot corresponds to the first of the four queues, which appears to have the most extreme behavior.

In order to show the movement of a boid that represents the four queues in this model, we show the path of the first queue (in red above) against the fourth queue for three different lengths of time in Figure 7 using the data in the first replication of Figure 6. Figure 7 shows the boid starting in the lower left corner of the space (near values of zero for both queues), and as time progresses both queues increase in value, with the first queue moving even farther from the zero. The third plot of Figure 7 can be compared with the middle plot, to show how the behavior of the system during a really long run differs from that observed in a medium length run.

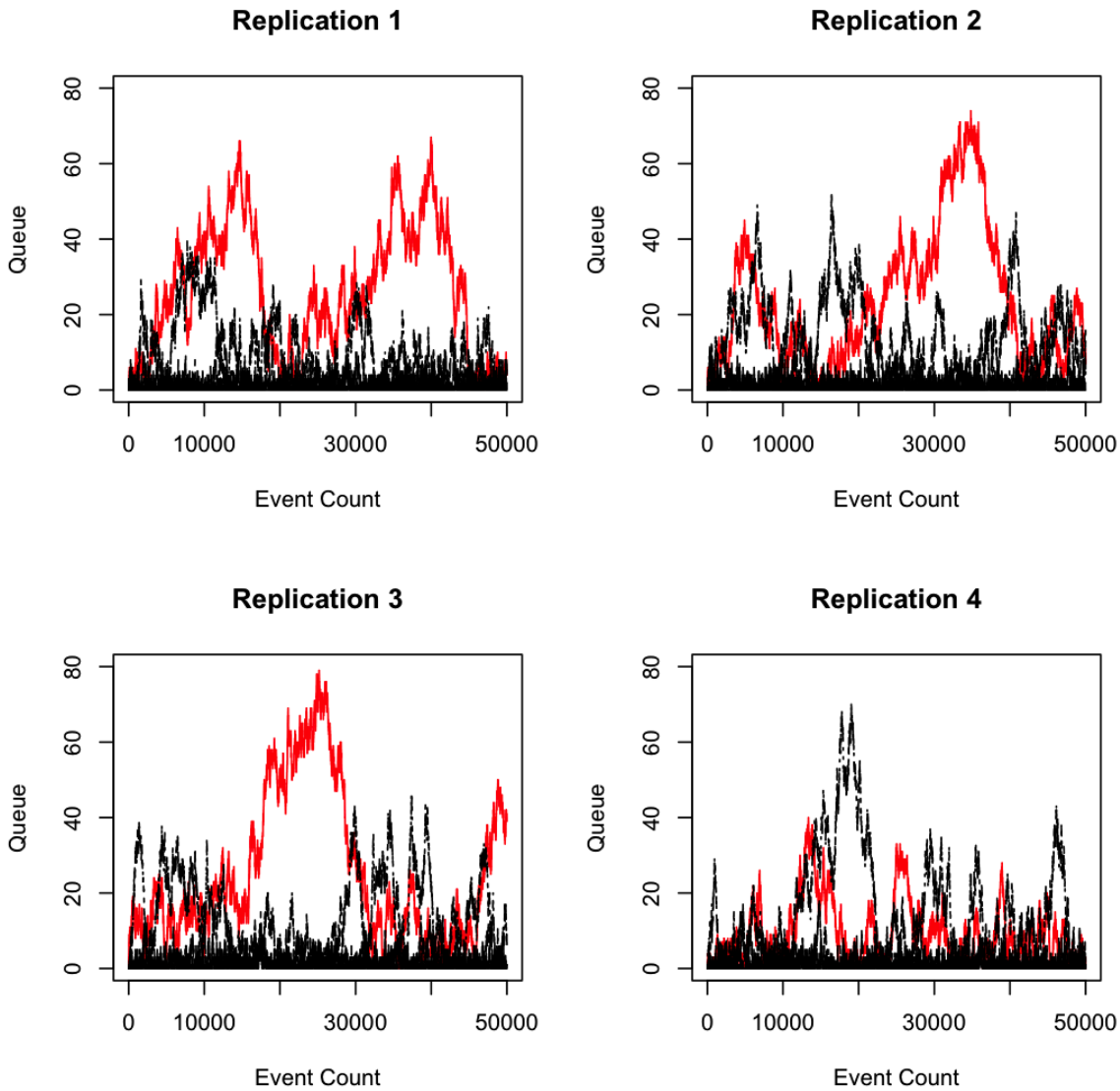

Figure 6: Output for the four queues from four independent replications of NetworkT.mod from sigmawiki.com.

3 RE-SAMPLING OUTPUT FOR SENSITIVITY ANALYSIS

Many important simulations take considerable time to run, have many input factors, and produce multiple, dependent output time series. For example: if only 32 factors were considered in a 2-level factorial experiment and each run takes 20 minutes, then it would take over a hundred and sixty thousand years to run the experiment. When only a few runs can be made at only a few design points, then output re-sampling becomes important in analysis. Various conventional and Bayesian bootstrapping methods have been used for decades to account for simulation input uncertainty (Barton and Schruben 1993) and for simulation output analysis (Friedman and Friedman 1993) (see Cheng (1995) for an early survey). Most papers consider only scalar time series and use ad-hoc conventional methods like block bootstrapping, where chunks of a time series are sampled with replacement, and perhaps shuffled.

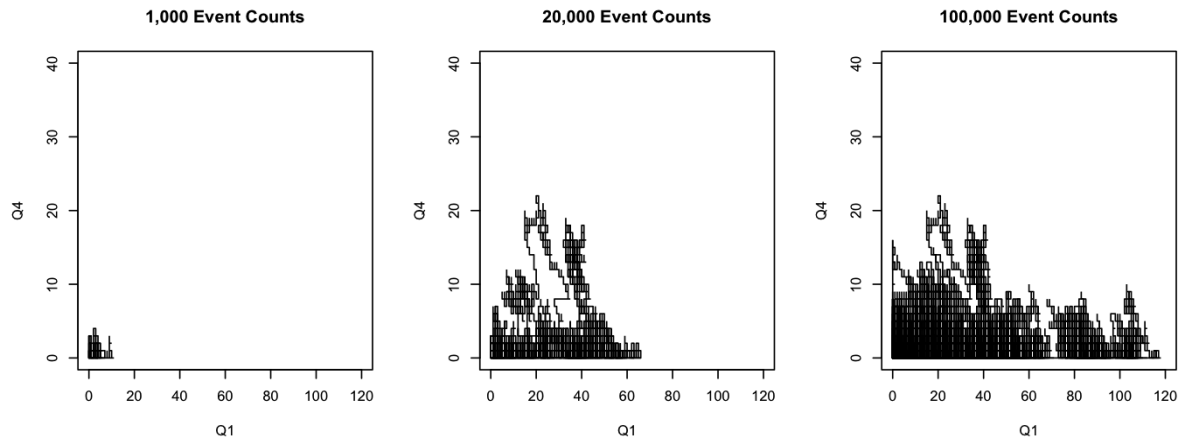

Figure 7: Trajectory of a boid from a replication of NetworkT.mod over a short, medium, and long length of time.

The new multivariate time series re-sampling technique proposed by Schruben and Singham for input modeling using agent flocking can equally be applied to re-sampling multivariate simulation output, with the advantages of easy sensitivity analysis using the concept of *affinity* proposed in that paper (an ordinal scale over [-1,1] of how much the future is believed will behave like, or unlike, the past). Much research on this promising new multivariate times series methodology remains, with huge potential impacts on both simulation input modeling and output analysis.

4 POTENTIAL OUTPUT ANALYSIS TOOLS

Sensitivity analysis is often hard to perform on trace data since there may only be one run available. If we can generate replications of the trace output data that are close but with some variation, the analyst can see how sensitive their system is to variations in the potential output. Another potential application is in forecasting. By creating a model of the boid path, future values in the time series can be forecasted by projecting the boid paths forward a few steps and mapping those points back to data. Currently flocking is being explored with some real biopharmaceutical production data where the goal is to use the time series of pH, temperature, and pressure during fermentation to predict yield (titer). We are currently looking for "flocking" of agents to set up control limits with the intention of using them to decide when to terminate what is likely to be an unproductive batch. An important industrial application of time series where the methodology proposed may be applicable is quality assurance. Quality control sampling in biopharmaceutical production is difficult since the results from a sample are sometimes not known for months. This lag adds another degree of complexity to statistical process control where, for example, a decision has to be made whether to continue or terminate fermentation based on partial information. The hope is that highly productive batches will "flock" past other high-yield batches. The results are encouraging, but are limited by the inability to plot 4-D graphs and the lack of any quantitative research on this approach.

Discrete event simulation inputs are typically event times and are continuous non-negative time series, indexed by event count. Avoiding static obstacles (like negative coordinate values) is a common feature of many flocking algorithms used in gaming and animation. The outputs of some simulations, notably queueing-type models, are piecewise constant. Many real queueing simulations are of such large systems (internet server farms, call centers, etc.) that continuous paths are probably reasonable approximations for some outputs. Adequate modeling of other simulation outputs may require that some of the flock coordinates be non-negative integers. Algorithms for exploring a high-dimensional lattice from MCMC literature like in Baumert et. al. (2009) may be useful. The boid steering technique called "wall following" in computer animation to mimic, for example, traveling through a maze, appears appropriate for modeling queue sizes.

Simulation needs to contribute more to designing global industrial, service, and social networks. Such systems include enterprise supply chains, national health care systems, and military communication systems. It can be infeasible to simulate a huge network to assess the impact of changes to a sub-system. (See Gunal and Pidd (2007) for a discussion of the challenges in whole-hospital simulation). A new method for integrating sub-system simulation models is necessary. Current approaches include analytical approximations or statistical meta-models for sub-system simulations. These appear in the literature with key words meta-modeling, model aggregation, and hybrid simulation. It is intriguing to imagine a sub-system simulation embedded in a flocking model of the behavior of the rest of the network. A key extension to flocking algorithms needed here is inclusion of the avoidance of *dynamic* obstacles that represent interactions among sub-systems, like starving or blocking in a supply chain. *Static* obstacle avoidance is incorporated in most current algorithms for animating flock behavior. Inserting and removing obstacles (in general, changing their shape) depending on the states of different sub-systems, or triggered by events in sub-system simulations, appears new.

REFERENCES

Barton, R.R. and L.W. Schruben. 1993. "Uniform and bootstrap resampling of empirical distributions." In *Proceedings of the 1993 Winter Simulation Conference,* edited by G.W. Evans, M. Mollaghasemi, E.C. Russell and W .E. Biles, 503-508. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Baumert, S., A. Ghate, S. Kiatsupaibul, Y. Shen, R.L. Smith, and Z.B. Zabinsky. 2009. "Discrete Hit-and-Run for Sampling Points from Arbitrary Distributions Over Subsets of Integer Hyperrectangles." *Operations Research,* 57(3): 727–739.

Cheng, Russell C. H. 1995. "Bootstrap Methods in Computer Simulation Experiments." In *Proceedings of the 1995 Winter Simulation Conference,* edited by C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman, 171-177. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Friedman, L. W., and H. H. Friedman. 1995. "Analyzing Simulation Output Using the Bootstrap Method." *Simulation*, 64(2): 95-100.

Gunal, M.M., and M. Pidd. 2007. "Interconnected DES models of emergency, outpatient, and inpatient departments of a hospital." In *Proceedings of the 2007 Winter Simulation Conference,* edited by S.G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J.D. Tew, and R.R. Barton, 1461–1466. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Law, A.M. 2007. *Simulation Modeling and Analysis*. 4th ed. McGraw-Hill, New York.

Moere, A. 2004. Time-Varying Data Visualization Using Information Flocking Boids. In *Proceedings of the IEEE Symposium on Information Visualization*, 97–104. IEEE Computer Society.

Reynolds, C. 1987. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques,* 25–34. ACM.

Schruben, L.W. 1981, "Control of Initialization Bias in Multivariate Simulation Response," Invited paper to *Communications of the ACM*, 24(4): 246-252.

Schruben, L.W., and D.I. Singham. 2010. "Simulating Multivariate Time Series Using Flocking." In *Proceedings of the 2010 Winter Simulation Conference,* edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 1048–1054. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

AUTHOR BIOGRAPHIES

LEE SCHRUBEN is a Chancellor's Professor at Berkeley and Honorary (non-paid) Chief Science (non-managing) Officer of Bio-G, Inc. Prior to Berkeley, he was on the Cornell faculty where he held the Schultz Endowed Professorship in Engineering and on the University of Florida Medical School faculty. His Ph.D. is from Yale and his email address is lees@berkeley.edu.

DASHI SINGHAM is a Research Assistant Professor in the Operations Research Department at the Naval Postgraduate School and is working with the SEED Center for Data Farming. She holds a Ph.D. in Industrial Engineering & Operations Research and an M.A. in Statistics from the University of California, Berkeley. She received a B.S.E. in Operations Research & Financial Engineering from Princeton University. Her research areas of interest are simulation analysis, applied probability and statistics, and statistical computing. Her email address is dsingham@nps.edu.