2010

Simulating Multivariate Time Series Using Flocking

Schruben, Lee W.

Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

SIMULATING MULTIVARIATE TIME SERIES USING FLOCKING

Lee W. Schruben
Dashi I. Singham

University of California, Berkeley
Department of Industrial Engineering & Operations Research
4141 Etcheverry Hall
Berkeley, CA 94720 USA

ABSTRACT

Notions from agent based modeling (ABM) can be used to simulate multivariate time series. An example is given using the ABM concept of flocking, which models the behaviors of birds (called boids) in a flock. A multivariate time series is mapped into the coordinates of a bounded orthotope. This represents the flight path of a boid. Other boids are generated that flock around this data boid. The coordinates of these new boids are mapped back to simulate replicates of the original time series. The flock size determines the number of replicates. The similarity of the replicates to the original time series can be controlled by flocking parameters to reflect the strength of the belief that the future will mimic the past. It is potentially possible to replicate general non-stationary, dependent, high-dimensional time series in this manner.

1 INTRODUCTION

Accurate simulations require complicated multivariate data series as input when modeling stochastic processes that are not stationary, independent, and identically distributed. This problem must be addressed when simulating many important systems.

The most straightforward solution is simply to drive the simulation with real data, in what is called a trace driven simulation. That data is used in a trace driven simulation to estimate the performance of a new system. Trace data has the advantages of containing information on trends and dependence that might be hard to model. While there is little argument that real input data is not realistic, this approach has some serious drawbacks. These include the fact that statistical analysis of the results of a trace-driven simulation experiment is not possible since there is only one observation of each scenario considered. Sensitivity analysis of trace-driven simulation models is hard to conduct and interpret. Most importantly, the validity of a trace-driven simulation experiment depends on the assumption that the future will be exactly like the past – rare events that did not occur yet never will, and rare events that did occur always will (muddling what one means by rare).

A solution to the problem of obtaining replications for statistical analysis is to bootstrap the time series (Härdle, Horowitz, and Kreiss 2003). A popular method is the block bootstrap; that is, divide a multivariate time series into (possibly overlapping) time blocks, sample the blocks randomly with replacement, and lay them end to end. There are several difficulties with using bootstrapped time series and this is an active area of statistical and econometric research.

An approach to both the problems of replication and sensitivity analysis is to use a parametric statistical model for the multivariate input time series. The parameters for these input processes may be estimated from data collected as part of a simulation study, or simply hypothesized as a baseline to be used in sensitivity analysis. There has been a great deal of research in developing quantitative models for simulating dependent input streams. Cario and Nelson (1996) developed a method for generating autoregressive series with a specified marginal distribution and autocorrelation from a Gaussian dependent series. Biller and Nelson (2003) introduce a method for generating vector autoregressive series with specified marginal distributions and autocorrelation matrix. Copulas can also be used to model dependent data, especially dependence in the tails of the distributions (Biller 2009). Many of the modeling techniques described above work by simulating input according to a pre-specified model. The parameters of the marginal distribution of the data may be estimated, but are taken

as given for the rest of the experiment. Once an autoregressive model has been selected, the goal is to generate input according to that model, not according to the data. This avoids overfitting the simulation results to the data. However, in many cases the data may not appear to be from a parametric statistical model. It may have a non-standard distribution, be non-stationary, or have a dependence structure that is not easily modeled. In these cases, a trace driven simulation, despite its many limitations, may be more appropriate. The user may expect that future input closely resembles the past. But we may want to test how sensitive the simulation output is to that particular set of past data.

In this paper, we propose looking at a completely different general approach to generating multivariate time series data. This will allow both replication and sensitivity analysis without explicitly defining a parametric statistical model. The data may non-stationary, dependent, and multivariate. The basic idea is to use agent based modeling concepts, such as flocking, to generate replications that will behave similarly to a multivariate time series, depending on the strength of the belief that the past represents the future.

Flocking is a technique developed to simulate the flight of a flock of birds. Reynolds (1987) introduced the concept of flocking in computer science modeling and qualitatively described the rules for the movement of the birds (which he called boids"). We propose generating a flock of boids that are attracted to an alpha boid whose flight is determined by the trace data. The paths of each of these other boids are then used to create replications in the simulation model. These replications may be qualitatively similar to the trace data. Using multiple replications can improve simulation results by avoiding the uncertainty associated with one input stream. Having random input that resembles the trace data can help determine how sensitive the system performance estimate is to the input process.

Using flocking as a way of visualizing data has been explored in the literature. Proctor and Winter (1998) introduced the concept of information flocking, which involves relating boids to data in order to visually see how similar elements of a population tend to flock together. Moere (2004) extended this idea to time varying datasets by having boid paths map to stock market prices for different companies.

Here we will map the trace data into a bounded hyperrectangle and use flocking methods to simulate boids that fly in the space. The path formed by the trace data (the alpha boid) is used to determine the general path taken by the boids. Boid paths may cross and do not need to be distinct from each other at all times. One can add noise to the paths in order to simulate different sets of replications for sensitivity analysis. The coordinates of the boid locations during their flight are mapped back to the components of the original multivariate time series for input to a simulation (or for other uses such as forecasting and quality control).

The boid paths generated using this method provide input streams to the simulation experiment that appear to be qualitatively similar to the trace data. We use this method to simulate paths having approximately the same trends, cycles, and dependencies as the original series. Flocking behavior can follow simple rules, or may involve sophisticated mathematical modeling. This idea has the potential to develop into a flexible, easily implementable tool for modeling multivariate time series.

2 AN EXAMPLE

We present an example of this method for simulating multivariate time series from trace data. The first step involves transforming the multivariate series (with n dimensions) into a single flight path for the alpha boid by mapping each series into a separate dimension in the hyperrectangle or orthotope. For hypothetical stationary time series with known marginal distributions, one can simply use cumulative distribution functions for the mappings. These will map the multivariate time series into the $[0,1]^n$ space. This results in approximately uniformly distributed marginals along each dimension of the hypercube.

The mappings of time series components into flight path coordinates should be monotonic and increasing to preserve the relationships in the data. We choose mappings with a bounded domain so that we may regulate the movement of the boids without allowing them to hit the boundary. Rather than keep the boids in real space, for illustration, we map the boids into the unit hypercube so that the distributions of their coordinates are approximately uniformly distributed along each edge. This allows us to generalize the flocking algorithm to many different kinds of multivariate time series. Since all the paths are bounded within $[0,1]^n$, each series is mapped to the same scale, so the flocking algorithm can be applied symmetrically to determine the movement of the boid in each of the n directions.

Suppose we have a multivariate time series $Y_{i,j}, i = 1,\ldots,m$ and $j = 1,\ldots,n$, where each j of the n series has m observations. Let $Y_j = (Y_{1,j},\ldots,Y_{m,j})$ correspond to the j^{th} series. We map the series Y_j to $Z_j = (Z_{1,j},\ldots,Z_{m,j})$ (in [0,1]) using the mapping f_j, which can be any cumulative distribution function. These experiments involved three possible functions f_j:

1. Empirical cumulative distribution function of Y_j.
2. Exponential cumulative distribution function using the sample mean of Y_j as the mean parameter.
3. Normal cumulative distribution function using the sample mean and variance of Y_j as the parameters.

The empirical cdf allows us to map each point into [0,1] without worrying about estimating any parameters, and the resulting mapping will be approximately uniform over [0,1]. The exponential and normal distribution mappings allow for the possibility of simulating values over an infinite space, whereas the empirical cdf restricts values to the range given by the data. The exponential mapping can also be used to scale down extremely large observations if the data has outliers. The normal cdf mapping will provide approximately uniformly distributed values of Z_j if Y_j appears normally distributed. For our examples in this paper, we use the normal cdf mapping.

The multivariate series $Y_{i,j}$ is mapped by paths f_j into a path in $[0,1]^n$. The values from the series at time i $(Y_{i,1},\ldots,Y_{i,n})$ map into the coordinates of the alpha boid at time i, $(Z_{i,1},\ldots,Z_{i,n})$. We use as an example the vector autoregression (VAR) model with lag p. Series in VAR(p) models depend on p previous values in their series and the other series. A VAR(1) model has the following form (Shumway and Stoffer 2000):

$$Y_{i,*} = \alpha + \Phi Y_{i-1,*} + \varepsilon_i \tag{1}$$

where $Y_{i,*}$ is the vector of the values of the series at time i, Φ is a matrix of the autoregressive coefficients, α is the vector of the n constant terms, and ε_i is a vector of the random error terms. For a given set of data, the estimates $\hat{\alpha}$ and $\hat{\Phi}$ can be found using a least-squares method. The example we use to demonstrate the flocking technique is from Zivot and Wang (2006) and is the following VAR(1) model with dimension $n = 2$:

$$\begin{pmatrix} y_{t,1} \\ y_{t,2} \end{pmatrix} = \begin{pmatrix} -0.7 \\ 1.3 \end{pmatrix} + \begin{pmatrix} 0.7 & 0.2 \\ 0.2 & 0.7 \end{pmatrix} \begin{pmatrix} y_{t-1,1} \\ y_{t-1,2} \end{pmatrix} + \begin{pmatrix} \varepsilon_{t,1} \\ \varepsilon_{t,2} \end{pmatrix} \tag{2}$$

where the error terms ε are normally distributed with mean zero, variance one, and covariance one-half. This model is covariance stationary because the eigenvalues of the coefficient matrix are less than one. As noted earlier, the multivariate time series does not need to be stationary, nor do we need to know the true marginals or dependency structures in the data. Figure 1 shows the plot of data simulated from this VAR(1) model, and the corresponding path in the space $[0,1]^2$ using the normal cdf mapping.

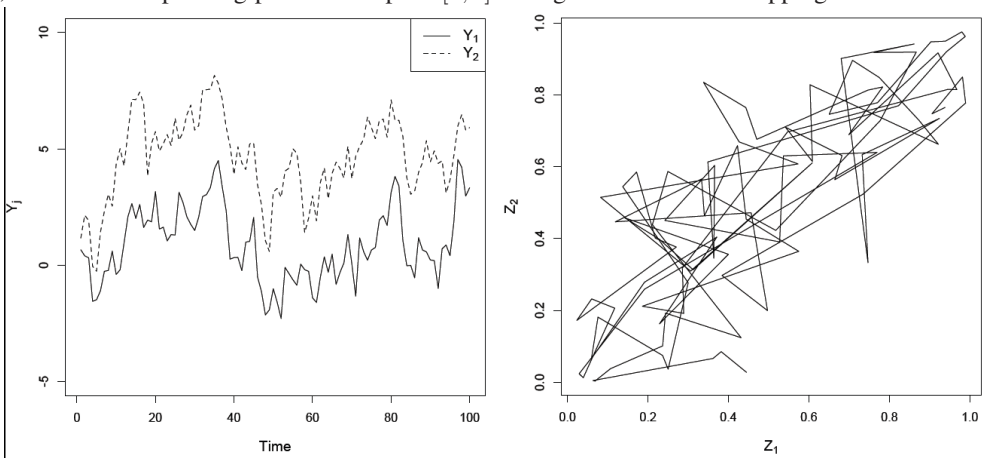

Figure 1: Sample time series simulated from a two-dimensional VAR(1) model (left). Mapped multivariate time series to a path in $[0,1]^2$ (right).

Once we have plotted the data as a boid traveling in the unit hypercube, we label that boid the alpha boid. Then we simulate other boids that are attracted to the alpha boid so that they may retain behaviors similar to those of the original data when their trajectory coordinates are mapped back to the individual time series. In the original boid models, there were three rules for determining the motion of individual boids that resulted in the emergent behavior of flocking, as described by Reynolds (1987):

1. Flock centering - boids try to stay close to the center of the flock.
2. Collision avoidance - the boids attempt to avoid colliding with each other.
3. Velocity matching - they all maintain approximately the same velocity as the lead boid.

Here we are not concerned with producing emergent behavior, but in using flocking for data simulation. For this application, we are not worried about implementing Rule 2 as we are not concerned if the boids cross paths. It is acceptable for different replicates to contain the same values of some or more components at the same time points. As for Rule 1, we will structure the flock centering to happen *around* the alpha boid. This allows the other boids to remain somewhat together around the alpha boid to retain its component dependency behavior. The third rule requires that boids move at approximately the same velocity. For our model, the path consists of m data points; it is not a continuous path. Though most flocking algorithms result in boids that appear to travel in a continuous path, the algorithms are actually updating the boid locations at discrete time points. Here, we generate the boid paths sequentially by updating their locations at times $i = 1, \ldots, m$ based on the corresponding locations of the alpha boid – the original data time indices. We update the locations of the boids according to a scale to measure attraction with the alpha boid, and include a random term used to simulate different replications. Section 3 details our algorithm. Figure 2 shows two sample boid paths, one with a low variance random term, and one with a higher variance random term.

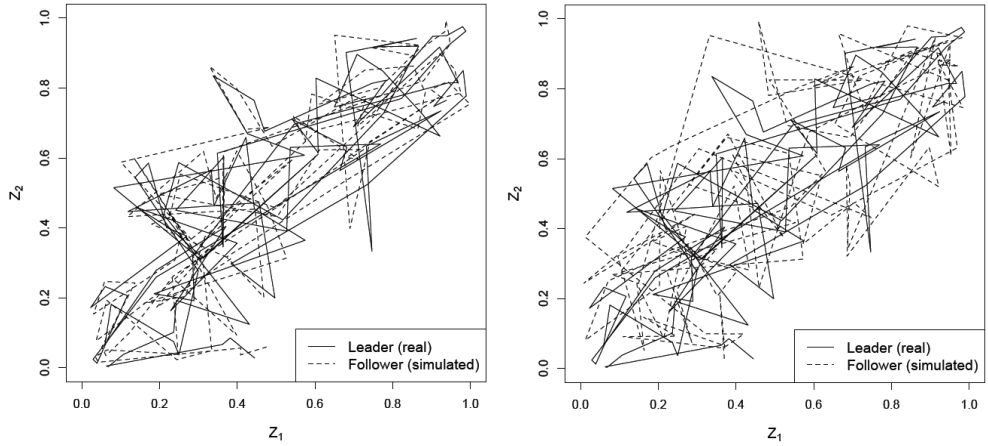

Figure 2: Simulated boids that follows the alpha boid, with low variance random terms (left) and higher variance terms (right).

It is possible to simulate the flock of boids around the original path of the data and skip mapping the data into the uniform hypercube. However, characteristics of the path of the alpha boid in the transformed space may provide some information on the type of dependence in the series. For example, if the time series are independent from each other, then the empirical cdf mapping will result in the path appearing to uniformly cover the hypercube. If there is positive dependence between the series, then the path will travel with a higher density in the region along the positive diagonal (as in Figure 2). The dependence between time points can be observed by seeing how the path moves between points. A jagged path with large distances between observations implies greater independence between time points, say in series with low autocorrelations. A smoother path with smaller distances between points implies that there is high positive autocorrelation, causing the alpha boid to move less between observations. We hope to capture the dependence qualitatively by letting the flock of boids be attracted to the alpha boid so that their coordinate dependency behaviors are similar. (The pitfalls of measuring dependency only by correlations are avoided entirely.)

Once we have simulated the boid paths, we map them back to the space of the original data using the inverse mapping f_j^{-1}. Each coordinate of the point $(Z_{i,1}, \ldots, Z_{i,j}, \ldots, Z_{i,n})$ is mapped by its appropriate function f_j^{-1} back to its corresponding value in the real world. This results in n separate series that can be plotted in real space. Figure 3 shows the plot of the two original series together with a simulated series generated from another flocking boid path. For clarity, Figure 3 shows only one boid path along with the path for the alpha boid; the flock size (number of replications) can be any size.

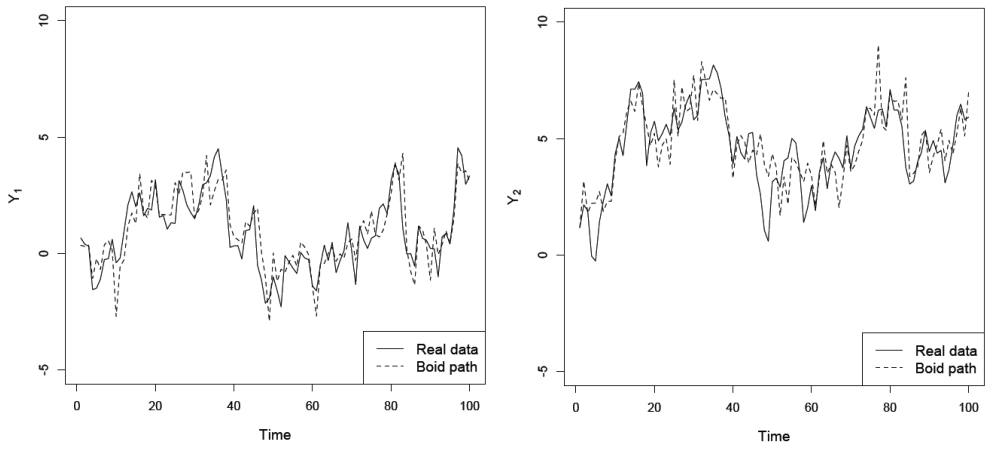

Figure 3: Path taken by one boid mapped back into real space in two dimensions, compared with the series Y_1 and Y_2.

3 NUMERICAL RESULTS

The motivation here is not to generate random samples from known parametric time series, many direct methods exist for that. In a simulation context, it may be more important to do sensitivity analysis to departures from past behaviors. It is important for sensitivity analysis to examine how a simulated system might perform when the input data is very different from (or very similar to) its past data for sensitivity analysis. We performed some preliminary experiments to measure how well in our example the boids statistically replicated the dependent time series. Again we consider the simple VAR(1) model used in (2). We simulate observations from this process and use those values as 'real data' to construct the flight path for the alpha boid. We then calculate the estimates of the VAR parameters $\hat{\Phi}$ and $\hat{\alpha}$ from the alpha boid to be a baseline for comparison with the parameter estimates of its other flock mates. Here, the alpha boid is the time series generated in Figure 1. Next, we simulated boids that were attracted to the alpha boid with varying levels of intensity. The simple flocking algorithm we used allows us to vary two parameters: the level of attraction of the flocking boids to the alpha boid, and the variance of the random noise in their flight paths. This algorithm is a modification of standard flocking algorithms, an example of which is provided in Parker (2007).

Let $x_{i,j}$ correspond to the location of the simulated follower boids at time i, in dimension j, where $i = 1,\ldots,m$ and $j = 1,\ldots,n$. Recall that $Z_{i,j}$ is the mapped position of the alpha boid in the $[0,1]^2$ space. Let λ be a parameter defining the level of attraction between the follower boids and the alpha boid. It can take values between zero and one. Then we simulate the movement of the follower boids recursively according to the following:

$$x_{i,j} = x_{i-1,j} + \lambda(Z_{i,j} - x_{i-1,j}) + \varepsilon_{i,j} \quad \forall i,j. \tag{3}$$

The boid moves towards the alpha boid with attraction level λ and has some error in its movement signified by the ε term, which for illustration here has a normal distribution with mean zero and variance σ^2. We varied the values of λ and σ^2 in our flocking algorithm to see how the boid paths varied.

If $\lambda = 1$ and $\sigma^2 = 0$, the follower boids exactly replicate the path of the alpha boid. For different values of λ and σ, we simulated 100 follower boids. We calculated the estimates of Φ and α associated with each path, and used the 100 replications to obtain interquartile ranges for the mean parameter estimates $\hat{\Phi}$ and $\hat{\alpha}$. Since assumptions of normality and independence did not appear to hold between replications (and between parameters), we looked at the interquartile ranges rather than the standard confidence intervals. We present results from two examples in Table 1.

It appears that when the noise term is low and the attraction is less than one, the parameter estimates may be biased in one direction, whereas if the attraction and noise are high, the estimates may be biased in the other direction. Decreasing λ and σ^2 tends to result in higher apparent autocorrelation between $Y_{i,j}$ and $Y_{i-1,j}$. The results in Table 1 are only two examples of several we ran, so any observations at this point are merely conjectures. Qualitatively, the algorithm works as we expected. When λ is small, the boids flock less towards

Table 1: Interquartile ranges of the estimates of the VAR(1) parameters of the follower boids.

Parameter	Model	Lead Boid Estimate	$\lambda = 0.90, \sigma = .01$	$\lambda = 1.00, \sigma = 0.05$
ϕ_{11}	0.70	0.63	[0.64, 0.67]	[0.50, 0.57]
ϕ_{12}	0.20	0.21	[0.20, 0.23]	[0.27, 0.32]
ϕ_{21}	0.20	0.18	[0.17, 0.21]	[0.25, 0.34]
ϕ_{22}	0.70	0.71	[0.68, 0.73]	[0.52, 0.59]
α_1	-0.70	-0.61	[-0.73, -0.62]	[-1.08, -0.84]
α_2	1.30	1.25	[1.18, 1.38]	[1.74, 2.00]

the alpha boid, so the apparent dependence of replications on history becomes higher and the paths remains closer to the simulated data values. If $\lambda = 1$, then each boid follows the alpha boid exactly, so λ behaves qualitatively like a global correlation (one can argue that our concepts of correlation are at best qualitative anyway). But increasing the noise reduces the dependence on the previous path. Despite the possible bias mentioned, the best fit VAR(1) model appears to have parameter estimates that are reasonably close to the estimates for the alpha boid time series. (Of course, like any such pseudo-experiments, the more replications the worse the results would look, even if the methodology were exact!)

4 CONCLUSIONS

We present what we believe is a different approach to generating replications of multivariate dependent time series. Single run trace driven simulations do not permit statistical output analysis or sensitivity analysis. Fitting a parametric model to generate replications allows for better sensitivity analysis, but a good model is not always available, particularly when the data is obviously not stationary. Bootstrapping time series has its own set of unresolved issues. Here we propose using ABM concepts such as flocking boids to simulate complex data. By incorporating error in the boids and varying the level of attraction to the alpha boid, we can generate new time series paths that appear to have dependency behaviors similar to those of the original data. This method has the potential to be a simple way of generating replications from data that is not easily modeled parametrically.

At the time of this writing, the most we can conclude is that this approach appears to have promise in being able to replicate approximately some very complicated behaviors in high-dimensional data series. Of course, our example is only one of many possible implementations of this idea and further research might produce a robust new way to model the behavior of multiple time series.

Future research includes exploring different flocking models, developing a global measure of the confidence we have that the future will follow the past, and refining the attraction parameter in the example. Repulsion can be incorporated into the models instead of attraction. We may want to find ways to maintain the dependency behaviors without mimicking the original time series too closely, as in Figure 3. We also plan to see how this method works for data that has different forms of dependence, or is non-stationary. Finally, we plan to improve the metrics to evaluate the performance of flocking algorithms to help define a range of applications, which now seems unlimited.

REFERENCES

Biller, B. 2009. Copula-based multivariate input models for stochastic simulation. *Operations research* 57 (4): 878–892.

Biller, B., and B. Nelson. 2003. Modeling and generating multivariate time-series input processes using a vector autoregressive technique. *ACM Transactions on Modeling and Computer Simulation* 13 (3): 237.

Cario, M., and B. Nelson. 1996. Autoregressive to anything: Time-series input processes for simulation. *Operations Research Letters* 19 (2): 51–58.

Härdle, W., J. Horowitz, and J. Kreiss. 2003. Bootstrap methods for time series. *International Statistical Review/Revue Internationale de Statistique* 71 (2): 435–459.

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.

Parker, C. 2007, September. Boids pseudocode. Available via <http://www.vergenet.net/ conrad/boids/pseudocode.html> [accessed June 7, 2010].

Proctor, G., and C. Winter. 1998. Information flocking: Data visualisation in virtual worlds using emergent behaviours. In *Virtual Worlds*, 168–176. Springer.

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.
Shumway, R., and D. Stoffer. 2000. *Time series analysis and its applications*. Springer Verlag.
Zivot, E., and J. Wang. 2006. *Modeling financial time series with S-PLUS*. Springer Verlag.

AUTHOR BIOGRAPHIES

LEE W. SCHRUBEN is in the Department of Industrial Engineering and Operations Research at the University of California, Berkeley. Prior to joining the Berkeley faculty, he held the Schulz Professorship in Engineering at Cornell. He received his Ph.D. from Yale and is a Fellow of the Institute for Operations Research and Management Science. Professor Schruben's research interests are in simulation modeling and analysis. His email address is <lees@berkeley.edu>.

DASHI I. SINGHAM is a Ph.D. candidate in Industrial Engineering & Operations Research at the University of California, Berkeley. She has a masters in Statistics from Berkeley, and holds a bachelors degree in Operations Research & Financial Engineering from Princeton University. Dashi's research interests include simulation modeling and analysis, and applied statistics. Her email address for these proceedings is <dsingham@gmail.com>.