ABSTRACT

This paper describes a procedure for assigning simulation trials to a set of parallel processors for the purpose of conducting a simulation study, involving a complex simulation model, in near real time. Unlike distributed simulation, where a complex simulation model is decomposed and its parts run in a parallel environment, the parallel replications approach discussed here involves running simulation replications to completion for the entire model. The unique element here is that the workload involved in running the simulation study is too time consuming to execute on a single workstation, so that the simulation analyst must utilize computer resources available through the web. New statistical methodology is needed for running a complex simulation study in a web-based, parallel replications environment.

INTRODUCTION

Biles and Kleijnen [1999] described a Java-based approach for allocating a set of simulation trials to $P$ parallel processors available through the world-wide web, where there were $R$ replications of the simulation at each of $S$ sets of input conditions. The methodology proposed there assumed that the simulation model had $N$ input variables $X_i$, $i = 1, \ldots, N$ and $M$ system responses $Y_j$, $j = 1, \ldots, M$, and that the objective of the simulation effort was to carry out the predictive phase (see Kelton, Sadowski and Sadowski [1998]) of a simulation study using experimental design, response surface methodology, or an optimization approach. The key component of the methodology described by Biles and Kleijnen [1999] was a program called the Simulation Manager that resided on a central processor under the control of the simulation analyst. The Simulation Manager allocated the $K$ simulation trials to the $P$ processors, sent a file to each processor specifying the input parameters necessary for that processor to carry out its assigned workload, and received back a file at the completion of that assignment that gave the summary statistics for the simulation activity.

Biles and Kleijnen [1999] assumed that the Simulation Manager would assign a simulation workload only to idle processors according to a priority scheme in which the faster processors among the $P$ slave processors received a proportionally higher workload than the slower processors. But later on, Biles, Marr, Storey and Kleijnen [2000] showed that a simulation task assigned to the $p^{th}$ processor was executed even if that processor
was in use by its owner during the simulation effort, albeit at a slower pace. For example, suppose that the \( p^{th} \) processor was in the process of executing a simulation task assigned by the *Simulation Manager* when the owner of that processor undertook a word processing task. The owner is unaware of the simulation activity underway, even though s/he has previously consented to make that processor available for simulation work by (a) leaving the processor switched on and (b) having the simulation application active on the “desktop.” Biles, Marr, Storey and Kleijnen [2000] showed that the simulation activity continued to progress, but at a slower rate than if the processor were idle. Hence, it is difficult to predict just how long a simulation assignment will take on a given processor, although the *Simulation Manager* maintains current records on the availability and projected usage of each of the \( P \) processors available to it.

Now this paper reports further developments in the web-based simulation research reported in the 1999 and 2000 papers. It focuses on how the *Simulation Manager* assigns workload to the \( P \) available processors and how it carries out the analysis of voluminous statistical results generated in the simulation study. Thus, this paper provides more details about the statistical methodology embedded in the *Simulation Manager*.

**STATISTICAL METHODOLOGY USED BY THE SIMULATION MANAGER**

The *Simulation Manager*’s role in coordinating a web-based simulation study is initiated by its sending the \( p^{th} \) processor a file containing the following data input:

1. The set of input conditions \( X_i, i = 1,\ldots,M \) at each of \( k_j, j = 1,\ldots,K \) assigned simulation tasks.
2. The set of random number seeds for each of \( U \) random processes included in the simulation model being executed. The random number seeds will be independent, common, or antithetic -- depending on the needs of the simulation study.
3. The initial number of replications \( n_r \) for the simulation.
4. The time length \( t_l \) of each replication.
5. The warm-up period \( t_w \) for each replication.
6. The prescribed half-width \( w_j \) of the 100(1-\( \alpha \)) % confidence interval for each of the \( M \) system responses \( Y_j, j = 1,\ldots,M \).

The \( p^{th} \) processor executes the assigned workload and sends a file back to the *Simulation Manager* containing the mean, standard deviation, minimum, maximum, and the 100(1-\( \alpha \)) % confidence interval for each of the \( N \) system responses \( Y_j, j = 1,\ldots,N \). That processor automatically computes the 100(1-\( \alpha \)) % confidence interval for each output response after \( n_r \) replications, determines whether any additional replications are needed to achieve the desired half-width, and executes these additional replications before sending the completed output file back to the *Simulation Manager*. This division of duties minimizes the traffic between the *Simulation Manager* and the \( P \) slave processors, and speeds the execution of the simulation study.
The *Simulation Manager* collects the simulation output from each of the $P$ slave processors and conducts the required analysis for a designed experiment, a response surface analysis, or an optimization algorithm as was set forth by the simulation analyst. The *Simulation Manager* ensures that the response $Y_j, j = 1, \ldots, M$ at point $X_k, k = 1, \ldots, K$ is distinguishable from every other point at the desired level of confidence. Should the responses at a given set of points $X_k, k = \{i, j : \{i, j \in K\}$ not be distinguishable with the desired level of confidence $100(1-\alpha) \%$, then the *Simulation Manager* assigns further work to a selected set of processors and repeats the analysis of the output provided by those processors until the required precision is met.

**EXPERIMENTS OF COMPARISON**

Whether the approach selected by the simulation analyst is an experimental design, regression metamodeling, optimization, or response surface methodology (see Kleijnen [1998]), the fundamental issue is based on analyzing “experiments of comparison”. The approach to assigning workload to each of the $P$ processors, each accessed over the web, is to assign one or more of the specific input vectors $X_k, k = 1, \ldots, K$ to each processor. Having received back the simulation results from each of the $P$ processors, the task of the *Simulation Manager* is to determine the best output $Y_j, j = 1, \ldots, N$.

The approach taken by the *Simulation Manager* in determining the appropriate number of replications at each input point $X_k, k = 1, \ldots, K$ is the “best-of-$k$-systems” approach described by Law and Kelton [2000] and modified by Nelson and Matejcik [1995]. In their methods, a sufficient number of replications are executed so that the simulation analyst can be $100(1-\alpha) \%$ confident that the response $y_k$ at $X_k$ is better ($>$ or $<$) than the same response at another point $X_{k'}$ with $k' \neq k$. In this way, the simulation analyst is $100(1-\alpha) \%$ confident that a selected solution is the correct one.

For instance, if the approach taken in the simulation study is that of Box’s complex search (see M. J. Box [1965]), then the “best-of-$k$-systems” approach is employed at each step in the simulation/optimization approach and deletes the “worst” point among $K$ points in the “constrained simplex” so that an improved replacement point can be generated and simulated (see Biles et al [1996]). After some number of points in the progress of the Box complex search have been simulated, the simulation analyst comes to the conclusion that any further simulation is too costly and accepts the current best point as the solution. So the simulation modeling is accomplished on the $P$ processors accessed through the web, while the mechanisms of Box’s complex search are executed by the *Simulation Manager*.

**SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS**

This paper has briefly described how to manage the statistical tasks associated with executing a simulation study on the world-wide web. For the most part, the $P$
processors accessed through the web – whether these processors are part of a locally available intranet or are part of an internet distributed across the globe – execute simulation replications plus whatever routine output statistical analysis is necessary to carry out the instructions sent to it by the Simulation Manager. The Simulation Manager, on the other hand, must possess software for all of the statistical methodology and optimization techniques necessary to analyze the simulation results sent back to it by the \( P \) processors.

The advantage gained by conducting simulation studies in this way is that executing numerous time-consuming simulation replications can be managed in minutes or hours, compared to the hours or days required to execute the entire workload on a single processor. Given a sufficiently large number \( P \) of slave processors, the task can be completed in near real-time.

REFERENCES


Box, M. J., “A New Method for Constrained Optimization and a Comparison with Other Methods,” Computer Journal (8), pp 42-52 (1965)


**AUTHOR BIOGRAPHIES**

**William E. Biles** is the Edward R. Clark Chair of Computer Aided Engineering in the Department of Industrial Engineering of the University of Louisville. He received the PhD in Industrial Engineering from Virginia Polytechnic Institute and State University. Dr. Biles has authored more than 100 journal articles and conference papers, two books, and 15 chapters in edited books and handbooks. He is a registered professional engineer and a fellow of the Institute of Industrial Engineers. His recent research has focused on optimization in simulation, web-based simulation, and the fusion of simulation and geographic information systems.

**Jack P. C. Kleijnen** is Professor of Simulation and Information Systems in the Department of Information Systems and Management of Tilburg University (Universiteit van Tilburg) in the Netherlands, where he is also associated with the Center for Economic Research (CentER). He received his PhD in Management Science from Tilburg University. Dr. Kleijnen has published six books and nearly 200 articles. His current research interests are in simulation, mathematical statistics, information systems, and logistics. He has lectured in numerous conferences throughout the USA, Europe, Israel, and Turkey. He has been awarded numerous fellowships around the globe.