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

SIMULATION-BASED OPTIMIZATION

Averill M. Law
Michael G. McComas

Averill M. Law and Associates, Inc.
P.O. Box 40996
Tucson, AZ 85717, U.S.A.

ABSTRACT

In this tutorial we present an introduction to simulation-based optimization, which is, perhaps, the most important new simulation technology in the last five years. We give a precise statement of the problem being addressed and also experimental results for two commercial optimization packages applied to a manufacturing example with seven decision variables.

1 INTRODUCTION

One of the disadvantages of simulation historically is that it was not an optimization technique. An analyst would simulate a relatively small number of system configurations and choose the one that appeared to give the best performance. However, based on the availability of faster PCs and improved heuristic optimization search techniques (evolution strategies, simulated annealing, tabu search, etc.), most discrete-event simulation-software vendors have now integrated optimization packages into their simulation software. It could arguably be said that optimization is the most significant new simulation technology in the last five years [see Fu 2001 and Law and Kelton 2000 (Section 12.6) for further discussion].

The goal of an "optimization" package is to orchestrate the simulation of a sequence of system configurations [each configuration corresponds to particular settings of the decision variables (factors)] so that a system configuration is eventually obtained that provides an optimal or near optimal solution. Furthermore, it is hoped that this "optimal" solution can be reached by simulating only a small percentage of the configurations that would be required by exhaustive enumeration.

In Section 2, we give a precise statement of the problem that is being addressed by simulation-based optimization. Section 3 discusses available optimization packages and the search techniques that they use. In Section 4 we give the results that were obtained from applying two commercial optimization packages to a manufacturing example with seven decision variables, and Section 5 provides a summary of the paper.

2 STATEMENT OF THE PROBLEM

Let $V_1, V_2, \ldots, V_k$ be decision variables (quantitative factors) for a simulation model. Let $f(v_1, v_2, \ldots, v_k)$ be an output random variable for the simulation model corresponding to the set of values $V_1 = v_1, V_2 = v_2, \ldots, V_k = v_k$.

Example 1. Consider the manufacturing system consisting of four work stations and three buffers (queues) as shown in Figure 1. Whenever a machine in work station 1 is idle, it will pull a blank (new) part in from an infinite supply. A machine cannot discharge a part if the succeeding buffer is full. Processing times have an exponential distribution with a mean that is given in Table 1. Let V_i (for $i = 1, 2, \ldots, 4$) be the number of machines in work station i and let V_i (for $i = 5, 6, 7$) be the number of buffer positions in buffer $i - 3$. Then $f(3, 2, 2, 3, 3, 1, 2)$ could, for example, be the number of completed parts for a 30-day period for the configuration shown in Figure 1.

Then the optimization problem of interest, *in general*, is given by the following:

$$\max E[f(v_1, v_2, \ldots, v_k)]$$

$$l_i \le v_i \le u_i$$

subject to the p linear constraints:

$$\begin{array}{c} a_{11}v_1 + a_{12}v_2 + \ldots + a_{1k}v_k \le c_1 \\ a_{21}v_1 + a_{22}v_2 + \ldots + a_{2k}v_k \le c_2 \\ \vdots \\ a_{p1}v_1 + a_{p2}v_2 + \ldots + a_{pk}v_k \le c_p. \end{array} \quad (1)$$

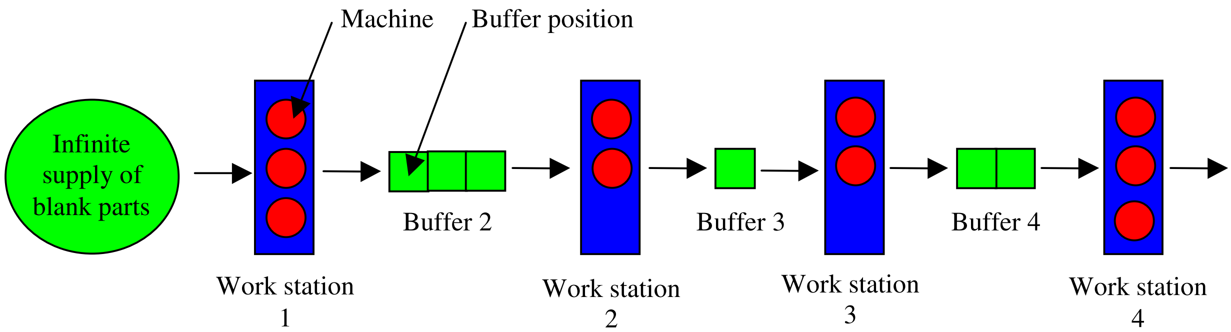

Figure 1: Manufacturing System Consisting of Four Work Stations and Three Buffers

Table 1: Mean Processing Times for Machines in the Four Work Stations

Work station	**Mean processing time for a machine (in hours)**
1	0.33333
2	0.50000
3	0.20000
4	0.25000

Thus, we want to maximize the objective function $E[f(v_1, v_2, \ldots, v_k)]$ ["*E*" means the expected value (or mean) of the random variable $f(v_1, v_2, \ldots, v_k)$] over all possible values of $v_1, v_2, \ldots, v_k$ that satisfy that range constraints $l_i \le v_i \le u_i$ (for $i = 1, 2, \ldots, k$) and the linear constraints given by (1). Note that l_i and u_i are lower and upper bounds for v_i. Also, the a_{ji}'s and c_j's in the constraints (1) are constants. Finally, "max" can be replaced by "min" in the objective function and the "less than or equal" in a linear constraint can be replaced by "equal" or by "greater than or equal." In Example 1, a possible constraint might be

$$v_1 + v_2 + v_3 + v_4 \le 10$$

i.e., the total number of machines cannot exceed 10.

In general, we will need to make n independent replications of the simulation for system configuration $v_1, v_2, \ldots, v_k$ and to use the sample mean over the n replications, $\overline{f}_n(v_1, v_2, \ldots, v_k)$, as an estimate of $E[f(v_1, v_2, \ldots, v_k)]$, since $f(v_1, v_2, \ldots, v_k)$ is a random variable. Note also that the form of the response surface $E[f(v_1, v_2, \ldots, v_k)]$ will not be known before any configurations are simulated, and it could, for example, have several local maxima.

3 AVAILABLE OPTIMIZATION PACKAGES

Table 2 lists the most prominent optimization packages available at the time of this writing, their vendors, the simulation-software products that they support, and the search techniques used. As can be seen, the four packages use different search heuristics, including evolution strategies (Bäck 1996, Bäck and Schwefel 1993), neural networks (Bishop 1995, Haykin 1999), scatter search (Glover 1999), simulated annealing (Aarts and Korst 1989), and tabu search (Glover and Laguna 1997). Note that genetic algorithms (Michalewicz 1996) is another well-known heuristic that is used for optimization.

Table 2: Optimization Packages

Optimization package	**Vendor**	**Simulation software supported**	**Heuristic procedures used**
AutoStat	Brooks-PRI Automation	AutoMod, AutoSched	Evolution strategies
Extend Optimizer	Imagine That	Extend	Evolution strategies
OptQuest	Optimization Technologies	Arena, Flexsim ED, Micro Saint, Pro-Model, QUEST, SIMUL8	Scatter search, tabu search, neural networks
WITNESS Optimizer	Lanner Group	WITNESS	Simulated annealing, tabu search

4 A DETAILED EXAMPLE

In this section we apply the OptQuest (Glover et al. 2002) (as implemented in Arena) and WITNESS Optimizer (Lanner 2002) optimization packages to the manufacturing system discussed in Example 1. There are seven decision variables and we assume that $u_i = 3$ for $i = 1, 2, \ldots, 4$ and $u_i = 10$ for $i = 5, 6, 7$; $l_i = 1$ for all values of i. Thus, there are $81{,}000 = 3^4 \cdot 10^3$ different combinations of the decision variables. There are no linear constraints for this problem.

Let

n_machines =	the total number of machines in all work stations
n_positions =	the total number of positions in all buffers
throughput =	the total number of parts produced in a 30-day period of time

Then define the objective-function random variable f (profit) as follows:

$$f = (\$200 \cdot \text{throughput}) - (\$25{,}000 \cdot \text{n\_machines}) - (\$1000 \cdot \text{n\_positions}).$$

The simulation run length for our experiments was 720 hours (30 days) with an *additional* warmup period of 240 hours (10 days). The throughput was computed from the final 720 hours of each 960-hour replication. We made $n = 5$ replications for each system configuration for each optimization package.

For OptQuest, we used a stopping rule that lets the optimization algorithm run until a user-specified number of configurations (*NC*) has been completed. (An alternative stopping rule is to let the optimization algorithm run until a user-specified amount of wall-clock time has elapsed.) We set *NC* = 300 and made five trials with different random numbers, with the average (across the five trials) profit for the best configuration being given in Table 3.

Table 3: Average Profit for the Best Configuration for the Two Optimization Packages

Optimization package	Average profit
OptQuest (as implemented in Arena)	\$593,816
WITNESS Optimizer	\$589,256

The stopping rule for the WITNESS Optimizer has two user-specified parameters: the maximum number of configurations (*MC*) and the number of configurations for which there is no improvement (*CNI*) in the value of the objective function. For example, suppose that *MC* = 500 and *CNI* = 25, and that the objective function value at configuration i is the largest up to that point. Then the algorithm will terminate at configuration $i + 25$ if the objective function values at configurations $i + 1, i + 2, \ldots, i + 25$ are all less than or equal to the objective function value at configuration i; however, the algorithm will never go beyond 500 configurations. We set *MC* = 500 and *CNI* = 75, and made five trials using different random numbers, with the average (across the five trials) profit for the best configuration being given in Table 3. Note that the results for the two optimization packages are within 1 percent of each other.

We have seen that the average profit is approximately \$590,000 for the two optimization packages that we considered. One might ask how close this is to the expected profit for the optimal solution and, also, what is the optimal system configuration? Work station 2 is potentially the bottleneck since its processing rate, 2 parts/hour, is the smallest of the four work stations. Therefore, we can argue heuristically that station 2 should have 3 machines, which gives station 2 a *potential* overall processing rate of 6 parts/hour. It follows that station 3 should have 2 machines – if it had only 1 (an overall processing rate of 5 parts/hour), then station 3 would be the bottleneck. (Three machines at station 3 are clearly not necessary.) By similar reasoning, station 4 should also have 2 machines. The question, then, is how many machines do we need at station 1? It might seem that we need 2 machines at station 1, since its maximum overall processing rate of 6 parts/hour would equal the maximum processing rate of station 2. However, it turns out that 3 machines are preferable, since this results in less idle time and a greater actual processing rate for station 2. (The moral is "never starve the bottleneck.") The resulting additional profit more than compensates for the cost of one more machine at station 1.

Thus, it would *appear* that 3, 3, 2, and 2 machines at stations 1, 2, 3, and 4, respectively, are optimal. This is substantiated by the fact that the configuration 3, 3, 2, and 2 was selected in 10 out of the 10 trials (five trials for each of the two packages).

We therefore fixed the machines at 3, 3, 2, and 2 and set out to determine the optimal number of buffer positions for each of the three buffers. We used the WITNESS Optimizer for this purpose, since it has an option that allows one to do an exhaustive enumeration of all possible system configurations. For each of the 1000 combinations of the numbers of buffer positions, we made $n = 50$ independent replications of the simulation model – 50,000 replications were made in all. (There was only one trial.) We found that buffer configuration 7, 8, and 4 had an *estimated* profit of \$591,588, which was the largest for the 1000 possible configurations. Furthermore, a 90 percent confidence for the expected profit for this configuration was [\$591,456, \$591,721]. The buffer configuration 7, 7, and 4 was a close second with an estimated profit of \$591,512. Therefore, it would *appear* that the optimal configuration should be close to 3, 3, 2, 2, 7, 8, and 4. Note that the estimated profit for the configuration 2, 3, 2, 2, 7, 8, and 4 was only $548,488.

5 SUMMARY

We have tested two different optimization packages with certain settings for their stopping-rule parameters on *one sample problem*. We found that their performance was good for *this* problem and for *the parameter settings used*. One should definitely not infer from these results how these (or other) optimization packages will perform on different problems that might be considerably more difficult in terms of the number of possible system configurations, the shape of the response surface $E[f(v_1, v_2, \ldots, v_k)]$, or the amount of inherent variability in the simulation model. A major concern at this time is how should one select an optimization package's stopping-rule parameters for a particular problem of interest, since little guidance is given in this regard. In the actual conference presentation, we will give a more extensive set of experimental results, in terms of the number of example problems and of the number of optimization packages tested.

Simulation-based optimization is a relatively new technology. However, it appears that it will have a considerable impact on the practice of simulation in the future, particularly when computers become significantly faster.

REFERENCES

Aarts, E. H. L. and J. Korst. 1989. *Simulated Annealing and Boltzman Machines.* New York: John Wiley.

Bäck, T. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, and Genetic Algorithms.* New York: Oxford University Press.

Bäck, T. and H. P. Schwefel. 1993. An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation* 1: 1-23.

Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*. New York: Oxford University Press.

Fu, M. C. 2001. Simulation Optimization. In *Proceedings of the 2001 Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 53-61. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Glover, F. 1999. Scatter Search and Path Relinking. In *New Methods in Optimization*, ed. D. Corne, M. Dorigo, and F. Glover. New York: McGraw-Hill.

Glover, F. and M. Laguna. 1997. *Tabu Search.* New York: Kluwer Academic Publishers.

Glover, F., J. P. Kelly, and M. Laguna. 2002. *The OptQuest Callable Library User's Documentation.* Boulder, Colorado: Optimization Technologies, Inc.

Haykin, S. S. 1999. *Neural Networks: A Comprehensive Foundation.* 2d ed. Upper Saddle River, New Jersey: Prentice Hall.

Lanner Group, Inc. 2002. *WITNESS Optimizer Module.* Houston, Texas.

Law, A. M. and W. D. Kelton. 2000. *Simulation Modeling and Analysis.* 3d ed. New York: McGraw-Hill.

Michalewicz, Z. 1996. *Genetic Algorithms + Data Structures = Evolution Programs.* 3d ed. New York: Springer-Verlag.

AUTHOR BIOGRAPHIES

AVERILL M. LAW is President of Averill M. Law & Associates, a company specializing in simulation consulting, training, and software. He has been a simulation consultant to numerous organizations including Accenture, ARCO, Boeing, Compaq, Defense Modeling and Simulation Office, Kimberly-Clark, M&M/Mars, 3M, U.S. Air Force, and U.S. Army. He has presented more than 345 simulation short courses in 17 countries. He has written or coauthored numerous papers and books on simulation, operations research, statistics, and manufacturing including the book *Simulation Modeling and Analysis* that is used by more than 80,000 people and widely considered to be the "bible" of the simulation industry. He developed the ExpertFit distribution-fitting software and also several videotapes on simulation modeling. He has been the keynote speaker at simulation conferences worldwide. He wrote a regular column on simulation for *Industrial Engineering* magazine. He has been a tenured faculty member at the University of Wisconsin-Madison and the University of Arizona. He has a Ph.D. in industrial engineering and operations research from the University of California at Berkeley. His E-mail address is <averill@ix.netcom.com> and his Web site is <www.averill-law.com>.

MICHAEL G. MCCOMAS is Vice President of Averill M. Law & Associates for Consulting Services. He has considerable simulation modeling experience in application areas such as manufacturing, oil and gas distribution, transportation, defense, and communications networks. His educational background includes an M.S. in systems and industrial engineering from the University of Arizona. He is the coauthor of seven published papers on applications of simulation.