APPLICATION OF A SIMULATION OPTIMIZATION SYSTEM FOR A CONTINUOUS REVIEW INVENTORY MODEL

Jorge Haddock
Golgen Bengu
Department of Decision Sciences
and Engineering Systems
Rensselaer Polytechnic Institute
Troy, New York 12180-3590

ABSTRACT

The system described here is a combination of a simulation generator, output analysis techniques, and optimization procedures. The simulation generator translates the description of a continuous review inventory control system into a SIMAN simulation model. The output analysis techniques estimate the mean and variance of the observations generated by the simulation program. The optimization techniques systematically generate the alternative scenarios and identify the optimal scenario. The simulation results are validated using the analytical solution to the problem.

1. INTRODUCTION

The system described here is a combination of a simulation generator, output analysis techniques, and optimization procedures. The simulation generator translates the description of a continuous review inventory control system into a SIMAN simulation model. The output analysis techniques estimate the mean and variance of the observations (i.e., the performance measure of the inventory system) generated by the simulation program. The optimization techniques systematically generate the alternative scenarios and identify the optimal scenario by performing a pairwise comparison of the mean and variance of these scenarios. Figure 1 illustrates the various simulation optimization techniques and their applications, reproduced from Bengu and Haddock (1987). The example problem considered here can be classified as having non-countable infinite number of alternatives. The general search procedures with the computer simulation technique are applied to optimize this problem in the context of a completely automated system. It may be noted that techniques such as Response Surface Methodologies, or sensitivity analysis techniques such as perturbation analysis have not been implemented in an automated framework as described here.

A stochastic inventory control problem is considered where the objective (function) is to maximize the total profit. The stochastic inventory process can be characterized as a single objective, multi-variable, optimization problem.

\[
\text{Max } F(X_1, X_2, \ldots, X_N)
\]

where, \( X_i \in \mathbb{R} = \{ y | y \in \mathbb{R} \} \), \( y \) integer or continuous

Figure 1. Types of simulation optimization systems based on number of alternatives

The decision variables are the reorder point and the reorder quantity. The input parameters are characterized by random variables and the inventory system is representative of real-life processes with price breakdowns and non-stationary distributions of input parameters. The analytical representation of such a complex inventory system is mathematically intractable to solve. In such a case, a viable alternative is the implementation of a simulation optimization system.

The simulation optimization system is validated by comparing the simulation results with the analytical results of the simplified version of the example problem. This simulation-optimization procedure can therefore be expected to yield improved simulation solutions for the original problem. This automated optimum-seeking procedure has the desirable features of simplifying the analyst’s task and shortening the overall decision making process.

2. SYSTEM DESCRIPTION

The system is organized into three primary modules: the simulation model generator, the output analysis methods and the search procedures. The simulation model generator creates the SIMAN model. Schruben’s initialization bias test (1983) and Law and Carson’s batch means method (1979) are used as output analysis techniques. Initial observations of the simulation model are truncated using the Schruben’s
procedure. The remaining observations are analyzed using the batch means method.

Schruben's procedure is a family of hypothesis test procedures based on standardized time series data analysis. The null hypothesis is that the observations analyzed have no initialization bias. The procedure increases the number of observations or the warm-up period until the null hypothesis is rejected. The variance term used in this procedure is computed using the Fishman's batch means variance estimator (Fishman 1974). The sequential version of the Law and Carson batch means procedure is used to estimate the mean and variance of the performance measure of the inventory system. The simulation run length for this procedure depends on the estimated lag 1 autocorrelation value based on the Jackknifed estimator, with threshold value of 0.4. The number of batches and the batch sizes needed to obtain independent observations are computed with respect to this run length.

The comparison of simulation experimental results can be considered as the Behrens-Fisher problem and a t-statistic (Hicks 1982) is used to compare the means. If there is no significant difference between the means, the variance terms are compared using the F-statistic. If there is no difference between these either, then one of the alternatives is chosen arbitrarily for comparison with the rest of the alternatives.

The search procedures systematically decrease the number of alternatives to compare. They start from a user defined alternative and progressively search over a continuous domain (or over lattice vertices) until the best alternative is found for a desired accuracy. A set of search procedures is included in the system to assist the user in the optimization process. The continuous variable search procedures used in the study are Pattern Search and Nelder and Mead search procedures. An integer variable search procedure similar to the Nelder and Mead procedure (Bengu 1987) was developed in conjunction with this system. Figure 2 sketches the interface between the simulation program and the search procedures.

The Nelder Method is an extension of the Simplex method by Spendley, Hext and Himsworth (Nelder and Mead, 1965). It adapts itself to a local landscape, using reflected, expanded, and contracted points to locate the minimum. At each point, the objective function is evaluated by using simulation. The user inputs the initial estimates of the independent variables, the reflection, expansion, and contraction coefficients, and the convergence criteria.

Hook's pattern search algorithm is a direct search method. It performs two types of search as shown in Figure 3. Given a point \( x_n \), an exploratory search along the coordinate axis is performed resulting in the point \( y \). Then an acceleration step starting from \( y \) in the direction of \( y-x_n \) leads to the

new point \( x_{n+1} \). The user inputs the starting points and the initial step sizes for these points, and the factors for reducing and extending these step sizes. The procedures are terminated when the convergence criterion is satisfied, or the iteration limit has been reached. The convergence criteria is specified in terms of the difference between the current value and the previous stage value.

Both the Nelder and Mead method and the integer variable method use three operating movements. The difference between them is that the latter searches over lattice vertices rather than over a continuous domain. Such a method accelerates the optimization procedure of the integer variable objective function when compared to methods which discretize the continuous domain (Bengu 1987).

The search procedures included in this study yield either a local or a global optimal point. If unimodality does not exist or the characteristics of the objective function are unknown, then a multistart approach using several starting points have to be considered. The algorithm can be executed by choosing alternate starting points and proceeding from each point to a local optima. The global optimum is assumed to be the best of the local optima but there is no guarantee that this is the global optimum. However, application of global optimization techniques such as the multistart technique (Schittkowski, 1985), can
provide a certain confidence level for the global optimum.

The search procedures used here are restricted to unconstrained optimization of the objective function. A successful, and frequently used approach to handle constraints, is to define an auxiliary unconstrained problem such that the solution of the unconstrained problem yields the solution to the constrained problem (Bazara and Shetty, 1979) (i.e., penalty functions).

The choice of an appropriate search procedure is an important factor that depends on the characteristics of the problem. The user has the option of choosing an appropriate search procedure in the developed simulation system (Bengu and Haddock, 1986). It is also possible to use more than one search procedure and compare the solutions.

3. SYSTEM EXECUTION

To execute the simulation system, the user inputs the system description interactively. The simulation generator creates the SIMAN experimental framework. SIMAN's software structure requires two files: the model and experimental framework. The model used for this particular application is maintained the same while the dynamics of the problem is simulated by changing the experimental framework. The performance of the inventory system for a given set of system parameters is estimated using the simulation model. Since input parameters are stochastic, the simulation results include randomness. Therefore, simulation experiment results need to be analyzed using the output analysis techniques. These techniques estimate the mean and the variance of the performance measure of the inventory system. The estimated performance values are subsequently analyzed by the simulation-optimization system to determine the optimal value of the decision variables (i.e., reorder point and reorder quantity) that maximize the objective function. The simulation optimization procedure obtains the optimal solution by comparing sequentially the simulation experiment results for each pair of alternatives, which are feasible combinations of the decision variables. The comparison is made using the difference of both mean and variance terms of the pair of alternatives and a modified t-statistic.

Figure 2 illustrates the interface between the simulation generator, the output analysis and the search procedures. The starting points and the operating movement coefficients of the search procedure are input to the SIMAN model (experimental frame). The search procedures act as an executive to the combined simulation and optimization program and determines the number of simulation runs (i.e., the number of alternatives to be searched). The output analysis procedure defines the terminating criteria and serves as an executive to the simulation model to determine the run length based on the given accuracy (Law and Carson procedure). The organization of the optimization subroutine and the SIMAN programs within SIMAN are illustrated in Figure 4.

![Figure 4. SIMAN subprogram configuration for simulation optimization system](image)

Figure 5 exhibits the commands and procedures required to complete the interface requirements between the generator and the simulation-optimization program. The interface requirements are satisfied automatically by the executive-command files which run both the generator and the optimization programs. The modular interface of simulation optimization system is accomplished using the executive commands. The command "X.GENERATE" is used to edit/input the simulation model. The command "X.OPTIMIZE" executes the optimization program. The command "X.OUTPUT" prints the optimal value of the decision variables and the maximum (optimal) revenue, as well as the results of each iteration on the screen. The command "X.GPLOT" plots the values of all the variables at each iteration. The search procedure is coded as a FORTRAN subroutine in the SIMAN software structure. The simulation model generator is also written in FORTRAN. The system was developed on the IBM 3081K system.

The decision support-generative/interactive simulation model requires less effort on the part of the user. The generator and the optimization procedures work independently of each other. It is therefore possible to use the optimization module as a separate optimization routine for analyzing different models (so long as the passing parameters are kept the same). The system is illustrated by means of an inventory problem described below.
4. AN ILLUSTRATIVE EXAMPLE

4.1. Problem Definition

A single item continuous review inventory system is considered. The inventory level is observed after each transaction (demand occurrence). Whenever the inventory drops below the reorder point, a fixed amount of inventory (i.e., reorder quantity) is ordered. The demand intensity, demand size, and the lead time are random. The parameters are provided by the user. The demand which cannot be met is assumed to be lost. The system allows quantity discounts on ordered inventory items. The objective is to determine the optimal reorder quantity and the reorder point so that the total profit is maximized. The total profit is the difference in between the long run average total revenue and the total cost. The total cost is the sum of the long run average total inventory, demand lost, and holding costs. The objective function is implicitly defined in terms of decision variables, reorder quantity and reorder point.

4.2. Analysis of Results

The analytical solution procedures for this inventory problem trades off storage cost with the ordering cost to determine the economic order quantity. The annual demand quantity used in the above expected value analysis is either forecasted or fixed “a priori”. The performance measure of the inventory system is the total revenue. The SIMAN model and generated experimental framework are illustrated in Figures 6 and 7 respectively for the particular example problem. A terminal sample session which illustrates the user defined inputs are given in Figure 8. Figure 9 plots the average revenue within the feasible range of the reorder point and the reorder quantity for the example problem using an exhaustive search procedure. It can be seen from the results and the graphical plots, that the search space most likely to contain the optimal point is where 0 <= RP <= 50 units and, 350 <= RP <= 400 units. When the Pattern and Nelder, et al. search procedures are executed with the initial base points within this reduced subspace, the optimal...
BEGIN;
PROJECT ASSIGNMENT. A. BENGUI, 01/06/85;
DISCRETE. 500.1;
RESOURCES: 1, 1;
INITIATE. X(1)=4000.0, X(1)=x(6)=200.0,
X(1)=200.0. X(1)=200.0, X(1)=X(12)=50.0, X(1)=200.0,
X(1)=X(12)=50.0, X(23)=450.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
X(1)=X(12)=50.0, X(1)=200.0, X(1)=200.0,
Figure 9. The plot of the results of the exhaustive search procedure

search procedures provide more satisfactory results. As can be seen from Table 1, the results in the neighborhood of the (300,20) base point with different starting points are more satisfactory for both search procedures.

These results are obtained by executing the simulation model for a long period of time. Table 2 illustrates the results of the search procedures combined with output analysis techniques for the same examples. The previous examples of the search procedures were the case where simulation runs are analyzed with no output analysis techniques. Each set of simulation output data is analyzed for truncation point first (BIAS PT.), then the mean value of total profit (FUNCTION VALUE) and the precision term (ST. DEV.) are calculated using sequential systematic sampling statistical analysis technique. The required number of function evaluations and the number of redundant evaluations (DATA SEARCH) are illustrated in the table for each example. Generally, the average total profit value is increased when the output analysis techniques are included. Because, the negative bias introduced by the initial starting conditions of this example are eliminated. The runs marked with an asterisk in Table 2 had limited execution.

Table 1. Results of the Pattern search and Nelder et al. search procedures with different starting points

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>(450,50)</th>
<th>(500,30)</th>
<th>(375,20)</th>
<th>(450,20)</th>
<th>(500,20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.5</td>
<td>25</td>
<td>114,477</td>
<td>114,495</td>
<td>115,995</td>
<td>111,287</td>
<td>115,765</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>125</td>
<td>114,477</td>
<td>115,935</td>
<td>115,995</td>
<td>113,287</td>
<td>115,765</td>
</tr>
<tr>
<td>3</td>
<td>.5</td>
<td>20</td>
<td>114,477</td>
<td>115,935</td>
<td>115,995</td>
<td>113,287</td>
<td>115,765</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>125</td>
<td>114,477</td>
<td>115,935</td>
<td>115,995</td>
<td>113,287</td>
<td>115,765</td>
</tr>
<tr>
<td>5</td>
<td>.5</td>
<td>25</td>
<td>114,477</td>
<td>115,935</td>
<td>115,995</td>
<td>113,287</td>
<td>115,765</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>125</td>
<td>114,477</td>
<td>115,935</td>
<td>115,995</td>
<td>113,287</td>
<td>115,765</td>
</tr>
<tr>
<td>7</td>
<td>.5</td>
<td>20</td>
<td>114,477</td>
<td>115,935</td>
<td>115,995</td>
<td>113,287</td>
<td>115,765</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>125</td>
<td>114,477</td>
<td>115,935</td>
<td>115,995</td>
<td>113,287</td>
<td>115,765</td>
</tr>
</tbody>
</table>

Figure 8. Terminal sample session
Table 2. Results of the search procedures with different starting points and using output analysis techniques

<table>
<thead>
<tr>
<th>NEW SEARCH PROCEDURE</th>
<th>INITIAL BASE POINT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(450,50)</td>
</tr>
<tr>
<td>FUNCTION VALUE.</td>
<td>117435</td>
</tr>
<tr>
<td>ST. DEV.</td>
<td>460</td>
</tr>
<tr>
<td>VARIABLEs</td>
<td>(465,22)</td>
</tr>
<tr>
<td>BIAS PT. #FUNCTION EVALUATI. DATA</td>
<td>500</td>
</tr>
<tr>
<td>SEARCH</td>
<td>40</td>
</tr>
<tr>
<td>CPU TIME (SEC.)</td>
<td>29.4</td>
</tr>
<tr>
<td>EXECUTION TIME (MIN)</td>
<td>2.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NELDER ET AL. PROCEDURE</th>
<th>α=1</th>
<th>β=0.5</th>
<th>γ=2</th>
<th>A=50</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(450,50)</td>
<td>(300,50)</td>
<td>(375,30)</td>
<td>(450,20)</td>
</tr>
<tr>
<td>FUNCTION VALUE.</td>
<td>117581</td>
<td>118418</td>
<td>118391</td>
<td>117026</td>
</tr>
<tr>
<td>ST. DEV.</td>
<td>460</td>
<td>464</td>
<td>464</td>
<td>456</td>
</tr>
<tr>
<td>VARIABLEs</td>
<td>(446,22)</td>
<td>(372,24)</td>
<td>(305,29)</td>
<td>(475,21)</td>
</tr>
<tr>
<td>BIAS PT. #FUNCTION EVALUATI. CPU TIME (SEC.)</td>
<td>500</td>
<td>150</td>
<td>150</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>58*</td>
<td>51</td>
<td>58*</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>71*</td>
<td>36</td>
<td>70*</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>7.5</td>
<td>4.9</td>
<td>6.75</td>
<td>6.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PATTERN SEARCH PROCEDURE</th>
<th>α=1</th>
<th>β=1</th>
<th>ε(1)=50</th>
<th>ε(2)=50</th>
<th>Acε=0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(450,50)</td>
<td>(300,50)</td>
<td>(375,30)</td>
<td>(450,20)</td>
<td>(300,20)</td>
</tr>
<tr>
<td>FUNCTION VALUE.</td>
<td>117614</td>
<td>114003</td>
<td>117723</td>
<td>115759</td>
<td>118269</td>
</tr>
<tr>
<td>ST. DEV.</td>
<td>453</td>
<td>443</td>
<td>450</td>
<td>400</td>
<td>442</td>
</tr>
<tr>
<td>VARIABLEs</td>
<td>(460,21)</td>
<td>(303,47)</td>
<td>(325,36)</td>
<td>(500,20)</td>
<td>(300,80)</td>
</tr>
<tr>
<td>BIAS PT. #FUNCTION EVALUATI. CPU TIME (SEC.)</td>
<td>500</td>
<td>150</td>
<td>200</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>58*</td>
<td>58*</td>
<td>38</td>
<td>19</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>72.4</td>
<td>71.1</td>
<td>27.2</td>
<td>15.6</td>
<td>21.3</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>19</td>
<td>9</td>
<td>7</td>
<td>3</td>
</tr>
</tbody>
</table>

388
5. RESULTS AND EXTENSIONS

This paper illustrates a self contained automated optimum-seeking procedure for practitioners. The generator program automates the experimental design of the system with the direct interaction of the decision maker. It eliminates the need to know the high level simulation language and provides decision support through an automatic optimization of the system. Integrated systems such as this provide capabilities of both model generation and automatic optimization. The combination of a simulation generator program with optimization procedures shortens the process time at each stage of the analysis.

This study also demonstrates a method of incorporating output analysis techniques within a simulation-optimization system. The system can be used as a tool to gain insight into new processes. The analyst can perform sensitivity analysis with respect to the desired parameters. For example, the decision maker can study the impact of changes in the demand and the lead time distributions on the holding and penalty costs.

Some of the other advantages that simulation generators can provide include model verification and cost savings. Model verification is imbedded in the verification of the simulation generator program, and this eliminates the need for separate model verification. In terms of cost savings, the simulation generators relieve the user from model building, coding, and debugging activities; all of which are time consuming and expensive. Simulation generators are an attractive alternative for both beginners and practitioners.

The modular configuration of the search procedures and the simulation generator program provides flexibility in combining other search methods for analysing more complex systems. Simulation experiments involving several response variables of interest to the analyst can also be handled by the system -- multi-criteria optimization.

REFERENCES

Basara, M.S., Shetty, C.M. (1979), Nonlinear Programming, John Wiley and Sons Inc.
Birta, L.G. (1977), "A Parameter Optimization Module for CSSL-Based Simulation Software", Simulation, April, 113-121
Haddock, J. (1987), "A Simulation Generator for FMS Control and Design", IIE Transactions, forthcoming

