Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

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,

$X_I \ \varepsilon\ \mathbb{R}: = \{\, y \mid y \ \varepsilon\ \mathbb{R}\}$, y integer or continuous

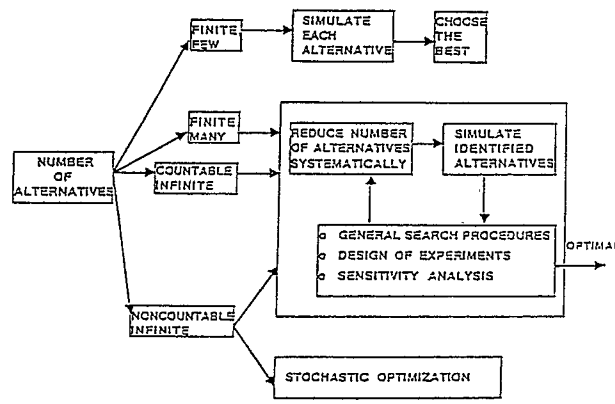

Figure 1. Types of simulation optimization systems based on number of alternatives

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 nonstationary 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. The 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.

Hooke's pattern search algorithm is a direct search method. It performs two types of search as shown in Figure 3. Given a point X_k, 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 $Y-X_k$ leads to the new point X_{k+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.

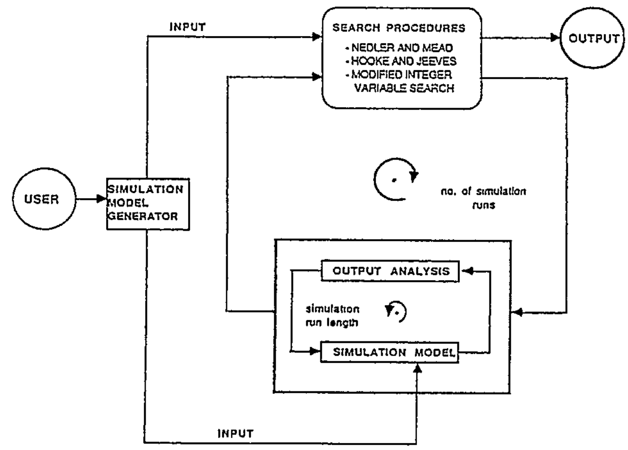

Figure 2. Modular interface of simulation optimization systems

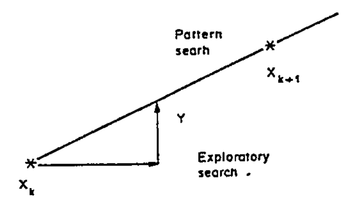

Figure 3. Illustration of Hooke and Jeeves method

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 multi-start 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.

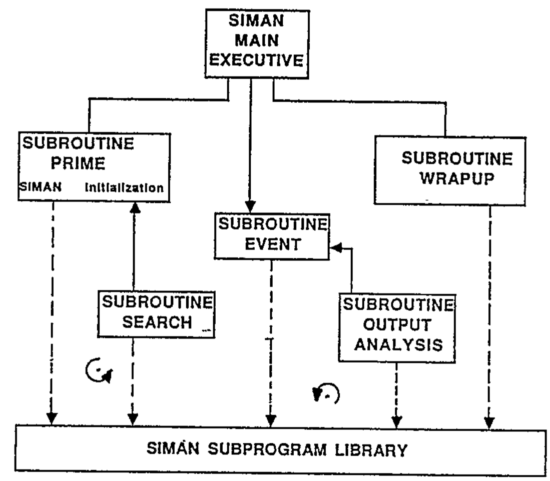

Figure 4. SIMAN subprogram configuration for simulation optimization system

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 independent 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.

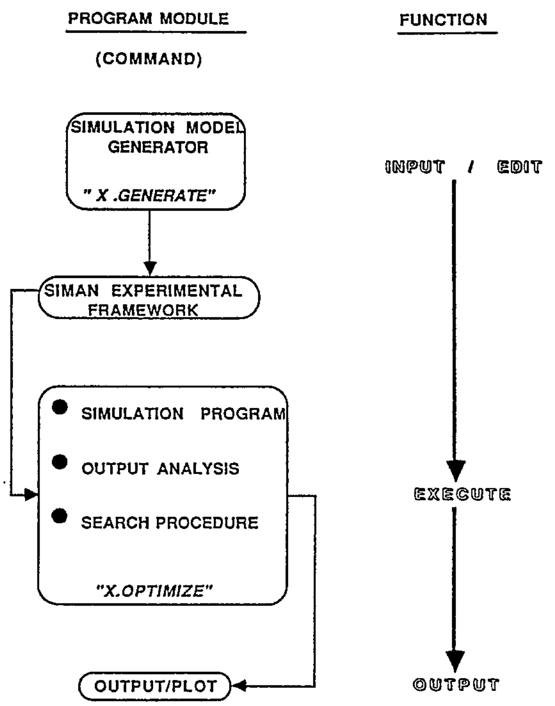

Figure 5. Executive commands for the interface of the generative simulation optimization system

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 =< RQ =< 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;
;   *** function:
;    simulation of an continuous review (q,r) inventory system
;    with fixed reorder quantity and partial lost sale.
;   *** variables:
;    X(1),  selling price                  A(1),  demand size
;    X(9),  inventory cost                 X(3),  lost sale
;    X(10), $ amount of discount           X(2),  inventory level(on
;    X(11), inventory on hand level               hand - on order)
;    X(7),  holding cost                   X(4),  reorder point
;    X(6),  shortage cost                  X(5),  reorder quantity
;   *** initialization
        CREATE;                                     initialization
        EVENT:1:DESTROY;
        CREATE,1;
             BRANCH,1:
             IF,X(2).LE.X(4),ORDER;
ORDER    ASSIGN:X(25)=X(25)-X(5);
         ASSIGN:X(2)=X(2)+X(5);
             BRANCH,1:
             IF,X(2).LE.X(4),ORDER:
             ELSE,CONTUE;
CONTUE   ASSIGN:X(17)=X(25);                total inventory ordered
         DELAY:ED(3);                          delay by lead time
         ASSIGN:X(11)=X(11)+X(25):DISPOSE; increase invt.on hand
;
;   *** model
         CREATE:ED(1);                             demand arrival
         ASSIGN:A(1)=ED(2);                           demand size
;                                         is there any discount
             BRANCH,1:
             IF,A(1).GE.P(4,1),DISCOUN:
             ELSE,COUNT;
DISCOUN  ASSIGN:X(9)=X(9)-X(10);        invt.cost with discount
COUNT    ASSIGN:X(8)=DAVG(2)*X(1)-DAVG(3)*X(9)-DAVG(4)*X(6)
                    -DAVG(5)*X(7);                total revenue
;                                     demand > invt.on hand ?
         ASSIGN:X(48)=X(8);                   passing parameter
             BRANCH,1:
             IF,A(1).GT.X(11),LOSTSAL:
             ELSE,SATISFY;
LOSTSAL  ASSIGN:X(2)=X(2)-X(11);            decrease invt. level
         ASSIGN:X(11)=0.0;                decrease invt. on hand
         ASSIGN:X(16)=X(11);             update satisfied demand
         ASSIGN:X(3)=A(1)-X(11):NEXT(EVALUA8);  update lost sale
;
SATISFY  ASSIGN:X(2)=X(2)-A(1);             decrease invt. level
         ASSIGN:X(11)=X(11)-A(1);          decrease invt.on hand
         ASSIGN:X(16)=A(1);                    update lost sale
;                            evaluate invt. level with reorder p
EVALUA8  BRANCH,1:
             IF,X(2).LE.X(4),REORDER:
             ELSE,LEAVE;
REORDER  ASSIGN:X(2)=X(2)+X(5);             increase invt. level
         ASSIGN:X(17)=X(5);             update inventory ordered
         DELAY:ED(3);                          delay by lead time
         ASSIGN:X(11)=X(11)+X(5);          increase invt.on hand
BYE      EVENT:2;                          control of run length
         TALLY:1,X(2):DISPOSE;
LEAVE    ASSIGN:X(17)=0.0:NEXT(BYE);    update inventory ordered

END;
```

Figure 6. SIMAN simulation model

reorder value and quantity are found to be 323 and 27 (Table 1).

The characteristics of the objective function and the search procedures require that more than one starting points be tried. The decision maker has to judiciously adjust the design and magnitude of the coefficients. The exhaustive search procedure reduces the range of these initial base points. As the optimum is approached, the multivariable

```
BEGIN;
PROJECT,ASSIGNMENT,G.BENGU,01/01/85;
DISCRETE,500,1;
RESOURCES:1,INVENT,1;
INITIALIZE,X(9)=4000.0,X(1)=10000.0,X(6)=200.0,
          X(7)=200.0,X(2)=200.0,X(10)=200.0,
          X(50)=30.0,X(21)=3.0,X(22)=30.0,X(23)=450.0,
          X(24)=3.0,X(11)=200.0,X(26)=1.0,X(27)=0.5,
          X(28)=2.0,X(29)=50.0,X(30)=0.00010,X(31)=300.0,
          X(32)=20.0,
          X(4)=3,X(5)=163.58;
PARAMETERS:1,1.0:
          2,20.0:
          3,3.0,9.0:
          4,200.0;
DISTRIBUTIONS:1,CO(1):
          2,EX(2,2):
          3,UN(3,3);
TALLIES:1,INVENT. ON HAND;
DSTAT:1,X(8),TOTAL REVENUE:2,X(16),SATISFIED DEMAND:
     3,X(17),ORDERED QUANTITY:4,X(3),LOSS DEMAND:
     5,X(11),INVENTORYON HAND:6,X(4),R. POINT:
     7,X(5),R. QUANTITY;
SEEDS:1,,YES:2,,YES:3,,YES;
REPLICATE;
END;
```

Figure 7. SIMAN experimental framework

```
******* TERMINAL SAMPLE SESSION ********

                LEGEND:  >> INDICATES USER RESPOND

>> X.GENERATE

   ************************************************
   *                                              *
   *  WELCOME TO A DECISION SUPPORTED GENERATIVE  *
   *  INVENTORY SIMULATION-OPTIMIZATION  MODEL    *
   ************************************************

   DO YOU NEED  INFORMATION ? (Y/N)
>> Y

 THIS IS AN INVENTORY CONTROL SIMULATION MODEL APPENDED WITH A
 DECISION SUPPORT SYSTEM. THE POLICY OF THIS INVENTORY SYSTEM IS
 CONTINUOUS REWIEV(Q,R) POLICY WITH FIXED REORDER QUANTITY (OR POINT)
 AND WITH PARTIAL LOST SALE CASE (AND NO BACKLOG)
 THE DYNAMIC VERSION OF THIS SIMULATION MODEL CALLED EXPERIMENTAL
 MODEL IS CREATED INTERACTIVELY WITH USER DECISIONS.
 THE LONG RUN AVERAGE SYSTEM REVENUE IS MAXIMIZED BY A USERCHOSEN
 SEARCH PROCEDURE. THE AVAILABLE SEARCH ALGORITHMS PROVIDESTO SEARCH
 FOR OPTIMAL REORDER QUANTITY OR REORDER POINT OR BOTH OF THEM.

 THE OBJECTIVE FUNCTION FOR THIS INVENTORY SYSTEM IS AS FOLLOW:

 LONG RUN AVG. REVENUE  =    LONG RUN AVG TOTAL PROFIT
                           - LONG RUN AVG TOTAL INVENTORY COST
                           - LONG RUN AVG TOTAL SHORTAGE COST
                           - LONG RUN AVG TOTAL HOLDING COST

 THE INPUT REQUIRED TO THE SYSTEM BY THE USER IS AS FOLLOW:

      1  - PURCHASE COST
      2  - SELLING PRICE
      3  - SHORTAGE COST
      4  - HOLDING COST
      5  - INITIAL INVENTORY
      6  - DISCOUNT SIZE
      8  - DECISION VARIABLE
      7  - REORDER POINT OR REORDER QUANTITY
      9  - DEMAND ARRIVAL RATE DISTRIBUTION
     10  - DEMAND SIZE DISTRIBUTION
     11  - REORDERING LEAD TIME DISTR

      ARE YOU READY TO INPUT ABOVE INFORMATION? (Y/N)
 >> Y
      PLEASE ENTER ALL NUMBERS AS REAL NUMBER. (WITH A PERIOD)
ENTER :
      YOUR NAME
>> G. BENGU
      THE DATE
>>  1/1/87
      THE PURCHASE COST  $/UNIT
>> 4000.00
      THE SELLING PRICE $/UNIT
>> 10000.00
      THE SHORTAGE COST $/UNIT
>>   200.0
      THE HOLDING COST $/UNIT
>>   200.0
      THE INITIAL INVENTORY ON HAND
>>  200.0
      THE QUANTITY WHERE DISCOUNT STARTS (IF ANY).
>>  200.0
      THE $ AMOUNT OF DISCOUNT (IF ANY).
>>  200.0
```

DISTRIBUTION		PARAMETERS		
NUMBER	NAME	(P1)	(P2)	(P3)
01	CONSTANT	CONSTANT	.	.
02	EXPONENTIAL	MEAN	.	.
03	POISSON	MEAN	.	.
04	ERLANG	MEAN	K	
05	UNIFORM	MIN	MAX	.
06	NORMAL	MEAN	STD.DEV	.
07	LOGNORMAL	MEAN	STD.DEV	.
08	GAMMA	BETA	ALPHA	.
09	BETA	THETA	PHI	.
10	WEIBULL	BETA	ALPHA	.
11	TRIANGULAR	MIN	MODE	MAX
12	E. D. P. D.	PK, K=1,3.. CUM. PROBABL. PK, K=2,4.. VALUES OF R.V		

```
         ** REFER ABOVE TABLE, AND  **

ENTER :
      DISTRIBUTION NUMBER FOR DEMAND ARRIVAL:
>> 01
      DISTRIBUTION NUMBER FOR  DEMAND SIZE
>> 02
      DISTRIBUTION NUMBER FOR  REORDER LEAD TIME
>> 05

ENTER:  PARAMETER VALUES (P1,P2,P3) OF DISTRIBUTIONS WITH ( , )

      FOR  DEMAND ARRIVAL
>> 1
      FOR DEMAND SIZE
>> 20.
      FOR REORDER LEAD TIME
>> 3,9

DO YOU NEED TO CHANGE ANY DISTRIBUTION NUMBER
            OR PARAMETERS ?  ( Y / N )
>> N
     ENTER SIMULATION TIME  (DEFAULT IS 500)
>> 500

ENTER:
     (1) TO FIND OPTIMAL REORDER POINT
     (2) TO FIND OPTIMAL REORDER QUANTITY
     (3) BOTH OPTIMAL R. POINT AND R. QUANTITY
>> 3

ENTER :
     (1) TO USE EXHAUSTIVE (SEQUENTIAL) SEARCH PROCEDURE
     (2) TO USE PATTERN SEARCH
     (3) TO USE NELDER & MEAD SEARCH
     (4) HELP
>> 1

ENTER:

 THE INTERVAL BOUNDRIES OF R. POINT TO SEARCH WITHIN (MIN.&MAX.)
>> 30 450
 THE NUMBER OF POINTS TO BE EVALUTED WITHIN THIS INTERVAL
                                              (DEFAULT IS 10)
>> 6
 THE INTERVAL BOUNDRIES OF R. QUANTITY TO SEARCH WITHIN (MIN &
>> 30 450                                                  MAX.)
 THE NUMBER OF POINTS TO BE EVALUTED WITHIN THIS INTERVAL
                                              (DEFAULT IS 10)
>> 6
     PLEASE WAIT... WHEN YOU SEE "JOB IS FINISHED "
            PRESS F1 FUNCTION KEY OR TYPE :
                     X. OPTIMIZE

            ****  JOB 0010 IS FINISHED  ****
>> X.OPTIMIZE

     PLEASE WAIT... WHEN YOU SEE "JOB IS FINISHED "
            PRESS F2 FUNCTION KEY OR TYPE :
                     X. OUTPUT

            ****  JOB 0011 IS FINISHED  ****
>> X.OUTPUT
```

```
**********************************************
*                                            *
*   SORT BY MAXIMUM REVENUE $                *
*                                            *
*   OBS R. POINT R. QUANT. AVG. REVENUE      *
*                                            *
*   ---  ------  -------   -----------       *
*    1     .       .        .                *
*    2    450     450     -120694            *
*    3    380     450      -88805            *
*    4    450     380      -71030            *
*    5    310     450      -59583            *
*    6    380     380      -46043            *
*    7    240     450      -28586            *
*    8    450     310      -27255            *
*    9    310     380      -18474            *
*   10    170     450      -10660            *
*   11    380     310       -3502            *
*   12     30     450         434            *

*   13    100     450        1849            *
*   14    240     380        7948            *
*   15     30      30        8907            *
*   16    450     240       16157            *
*   17    310     310       17520            *
*   18    170     380       21138            *
*   19     30     380       24141            *
*   20    100     380       30438            *
*   21    380     240       31826            *
*   22     30     100       33843            *
*   23    100      30       36979            *
*   24    240     310       38639            *
*   25     30     310       39198            *
*   26     30     240       43810            *
*   27    170     310       45983            *
*   28     30     170       49023            *
*   29    100     310       49050            *
*   30    310     240       49294            *
*   31    450     170       52255            *
*   32    240     240       63757            *
*   33    100     240       64366            *
*   34    380     170       64659            *
*   35    100     100       65114            *
*   36    100     170       65592            *
*   37    170     240       69544            *
*   38    310     170       74755            *
*   39    170     170       83365            *
*   40    170      30       83444            *
*   41    450     100       83654            *
*   42    240     170       86086            *
*   43    170     100       86462            *
*   44    380     100       90112            *
*   45    310     100       96814            *
*   46    240     100       99250            *
*   47    240      30      104269            *
*   48    450      30      111200            *
*   49    380      30      113332            *
*   50    310      30      113427            *
*                                            *
**********************************************
*     OPTIMAL    OPTIMAL      MAXIMUM        *
*   R. POINT  R. QUANTITY  REVENUE $         *
*                                            *
*     310.00      30.00      113427.         *
*                                            *
**********************************************

ENTER:
   (1)  TO OPTIMIZE THE MODEL
   (2)  TO EDIT MODEL
   (3)  QUIT
>> 3
```

Figure 8. Terminal sample session

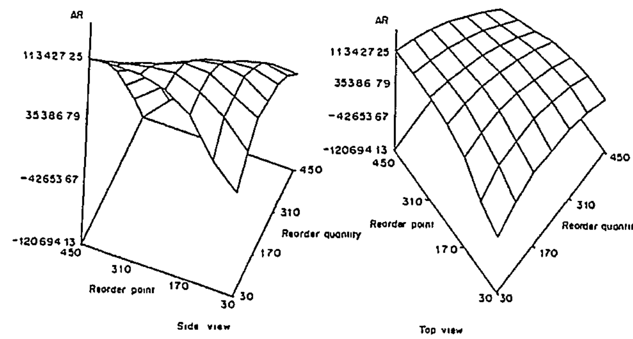

Figure 9. The plot of the results of the exhaustive search procedure

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

	A B C D	(450,50)	(300,50)	(375,30)	(450,20)	(300,20)	
NELDER MEAD SEARCH	1 .5 2 50	114277 (443,23)	114990. (303,29)	114899 (373,24)	113827 (462,23)	115356 (323,27)	30-40
	1 1 2 50	114277. (443,23)	115336 (324,27)	114899 (373,24)	113827 (462,23)	115336 (324,27)	
	1 .5 20 50	114277 (443,23)	114990 (303,29)	114899. (373,24)	113827 (462,23)	115140 (321,29)	
	1 1 2 100	113504 (515,23)	115224 (371,23)	114899 (373,24)	113827 (462,23)	113827 (462,23)	
	2 1 4 50	113445 (498,22)	114805 (348,25)	115032 (379,24)	113827 (462,23)	114923 (324,29)	
	3 1 3 40	113891 (459,23)	114982. (311,25)	114194 (408,22)	113795 (464,22)	115162 (320,29)	
	E F G1 G2						

	1 1 40 40	109893 (290,50)	110454. (300,50)	114025. (335,30)	112061. (530,20)	108674. (260,60)	
HOOKE	1 1 50 50	106013 (400,50)	110454 (300,50)	114359 (325,30)	112477 (500,20)	114110 (350,30)	
AND	1 1 20 20	114206. (370,30)	114484. (320,30)	114025 (335,30)	111120 (470,20)	112745 (300,40)	
JEVEES: (PATTERN)	1 1 10 10	114110 (380,30)	113896 (300,30)	113914 (355,30)	112477 (480,20)	114484 (320,30)	20-30
SEARCH	1 1 30 30	112360 (510,20)	110947 (270,50)	113800 (345,30)	112360 (510,20)	110947 (270,50)	
	5 1 30 30	111454 (630,20)	110947 (270,50)	113800 (345,30)	112360 (510,20)	110947 (270,50)	
	5 1 40 40	109893 (290,50)	110454 (300,50)	114025 (335,30)	112061 (530,20)	108674 (260,60)	

Table 2. Results of the search procedures with different starting points and using output analysis techniques

NEW SEARCH PROCEDURE	INITIAL BASE POINT				
	(450,50)	(300,50)	(375,30)	(450,20)	(300,20)
FUNCTION VALUE.	117435	114013	118189	119347	118674
ST. DEV.	460	447	250	200	461
VARIABLES	(465,22)	(300,49)	(377,22)	(438,22)	(308,27)
BIAS PT.	500	150	250	200	150
#FUNCTION EVALUATI.	40	11	37	32	19
DATA SEARCH.	10	3	19	7	3
CPU TIME (SEC.)	29.4	9.7	28.1	15.1	27.3
EXECUTION TIME(MIN)	2.5	1.8	3.4	4.5	3.0

NELDER ET AL. PROCEDURE	$\alpha=1$ $\beta.5$ $\gamma=2$ A=50				
	(450,50)	(300,50)	(375,30)	(450,20)	(300,20)
FUNCTION VALUE.	117581.	118418.	118391.	117026.	118967.
ST. DEV.	460	464	464	458	466
VARIABLES	(446,22)	(372,24)	(305,29)	(475,21)	(326,25)
BIAS PT.	500	150	150	200	200
#FUNCTION EVALUATI.	58*[1]	51	58*	52	58*
CPU TIME (SEC.)	71*	36	70*	66	72*
EXECUTION TIME(MIN)	7.5	4.9	6.75	6.4	6.9

PATTERN SEARCH PROCEDURE	$\alpha=1$ $\beta=1$ $\varepsilon(1)=50$ $\varepsilon(2)=50$ Ac=.99				
	(450,50)	(300,50)	(375,30)	(450,20)	(300,20)
FUNCTION VALUE.	117614	114003	117723.	115759	118269.
ST. DEV.	453	443	450	400	442
VARIABLES	(460,21)	(303,47)	(325,30)	(500,20)	(300,80)
BIAS PT.	500	150	200	150	150
#FUNCTION EVALUATI.	58*	58*	38	19	29
CPU TIME (SEC.)	72.4	71.1	27.2	15.6	21.3
EXECUTION TIME(MIN)	10	19	9	7	3

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 a 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 analyzing 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

Bazara, M.S., Shetty, C.M. (1979), Nonlinear Programming, John Wiley and Sons Inc.

Beightler, C.S., Phillips, D.T., Wilde, D.J. (1979), Foundations of Optimization, Prentice Hall Inc.

Bengu, G. (1987), "A General Simulation Optimization System using the Output Analysis Techniques", unpublished Ph.D. dissertation, Clemson University, May

Bengu, G. and J. Haddock, "A Generative Simulation- Optimization System", Computers and Industrial Engineering, Vol. 10, No. 4, 1986

Bengu, G. and J. Haddock, "Implementation of General Search Procedures in a Simulation Optimization System", Working Paper, Department of Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute, Troy, New York, June 1987

Birta, L.G. (1977), "A Parameter Optimization Module for CSSL-Based Simulation Software", Simulation, April, 113-121

Clementson, A. T. (1973), "Extended Control and Simulation Language", University of Birmingham, Birmingham, England

Daugthert, A.F., Turnquist, M.A. (1978), "Simulation Optimization Using Response Method on Spline Approximations", Proceedings 1978 Winter Simulation Conference, 183-193

Fishman, G. S. (1974), Concepts and Methods in Discrete Event Digital-Simulation, John Wiley & Sons, Inc.

Haddock, J. (1987), "A Simulation Generator for FMS Control and Design", IIE Transactions, forthcoming

Hooke, R., Jeeves, T. A. (1961), "Direct Search Solution of Numerical and Statistical Problems", Journal of the Association of Computer Machines, Vol. 8, 212-229

Koester, J.L. and J.H. Mize (1973), "Optimization Techniques with FORTRAN", New York, McGraw- Hill

Law, A.M., Carson, J.S. (1979), "A Sequential Procedure for Determining the Length of a Steady-State Simulation", Operations Research, Vol. 27, No. 5, 1011-1025, October

Mathewson, S.C. (1984), "The Application of Program Generator Software and Its Extensions to Discrete Event Simulation Modeling",IIE Transactions, September

Nelder, J.A. and Mead, R. (1965), "A Simplex Method for Function Minimization", Computer Journal, Vol. 7, 308-313

Oldfather, P. M., Ginsberg A. S. (1966), Markowitz, H.M., "Programming by Questionnaire: How to Construct a Program Generator", Rand Report RM-5129-PR, November

Pegden, C. D. (1982), Introduction to SIMAN, Systems Modeling Corporation, State College, Pennsylvania

Pegden, C. D., Gately, M. P. (1980), "A Decision Optimization Module for SLAM", Simulation Vol. 34 No. 1, 18-25

SAS/GRAPH USER'S GUIDE (1981), SAS Institute Statistical Analysis System, Box 8000, Cary, North Carolina

Schittkowski, K. (1985), Computational Mathematical Programming, Series F: Computer and Systems Sciences, Vol. 15, Springer-Verlag Berlin Heidelberg

Schruben, L. and Singh, H. (1983), "Optimal Tests for Initialization Bias in Simulation Output", Operations Research, Vol. 31, No. 6, December

Smith, D. E. (1973), "An Empirical Investigation of Optimum-Seeking in the Computer Simulation Situation", Operations Research, Vol. 21, 475-497

Smith, D. E. (1976), "Automatic Optimum-Seeking Program for Digital Simulation", SIMULATION, 27-31, July

Talavage, J.J. (1984), "Simulation Optimization for Decision Support in Operating a Robotic Manufacturing System", Proceedings 1984 Winter Simulation Conference