LITERATURE REVIEW AND BIBLIOGRAPHY OF SIMULATION OPTIMITATION

William Farrell
Graduate School of Business
University of Pittsburgh

Management Scientists and systems analysts often wish to find the values of input variables which optimize (maximize or minimize) some function of system performance. If the system can be described analytically, mathematical programming is used to find the optimum. When systems are too complicated to be described analytically, simulation is the appropriate tool for modelling systems. Therefore, methods of optimization through simulation are quite important to the management scientist. This paper consists of a discussion and a bibliography of the optimization of simulated systems.

I. INTRODUCTION

The attempt to find a system optimum is of prime importance to the management scientists or systems analyst. The systems defined by management scientists include a set of input variables, constraints on attainable values of input variables, and an objective function of the input variables. This objective function is to be optimized (maximized or minimized) by finding (among variable values which do not violate any constraints) values which yield a best solution to the objective function. The simplest such problems can be solved analytically. Somewhat more complicated problems have objective functions and constraints which can be defined analytically and solved via one of the iterative techniques of mathematical programming.

For the most complicated problems, the objective function can not be described analytically and the analyst must resort to simulation to describe the system. The simulation can be viewed as an objective function whereby feasible input variable values are converted into an output value. It would be desirable to use mathematical programming techniques to determine an optimum of a system described in this manner; however, very little is known about the mathematical form of this objective function. There is no way of directly knowing its shape. It is not possible to take directional derivatives, as is necessary for steepest ascent algorithms.

In spite of this difficulty, several techniques have been proposed for finding best, or at least improved, solutions to optimization problems of this sort. Some of these methods are naive, such as generating several input values at random and running the simulation at each of these inputs. Some are relatively sophisticated, such as running the simulation at several points near some intermediate solution, using those points to define a hyperplane which approximates the shape of the objective function near the intermediate solution, and taking directional derivates of the hyperplane to approximate a steepest ascent direction. None of these techniques is sure to work for all problems, but each has characteristics which make it useful for certain types of problems.

In this paper, we divide the techniques into three categories: mathematically naive techniques; methods appropriate to unimodal objective functions, and techniques useful for multimodal objective functions. Additionally, we discuss papers which compare some of these techniques in test cases. Section 2 contains the discussion and section 3 the bibliography. For more complete discussions of each technique, the reader is referred to Farrell, McCall and Russell [44].

2. DISCUSSIONS

This section consists of brief discussions of many techniques which may be used to find improved solutions to optimization problems. In section 2.1, we consider naive techniques, which do not try to infer any mathematical properties of the objective function.

Section 2.2 consists of techniques appropriate to unimodal objective functions. Several nonlinear programming techniques are found here. Section 2.2 is divided into two parts. Section 2.2.A includes methods of approximating the shape of the objective function. The approximate objective function is differentiated to provide an approximate steepest ascent direction. Section 2.2.B contains those nonlinear programming techniques which do not require a derivative of the objective function.

The techniques discussed in section 2.3 attempt to find optima for problems with multimodal surfaces. Finally, Setion 2.4 notes studies which have reviewed the literature of this area or compared several techniques.

2.1 Naive Techniques

The Characteristic which identifies the naive techniques is that no mathematical knowledge of the objective function is required. We discuss three such methods of finding improved solutions, together with variations on two of them. These are heuristic search, complete enumeration, and random search, with variations on complete enumeration and random search.

Heuristic Search. Often the analyst will use his knowledge about the system to repeatedly guess at input values, run the simulation, and stop when he is satisfied he has a very good solution. We refer to this approach as heuristic search. It is impossible to evaluate the success of such a method, since it depends entirely on the knowledge, experience and insight of the analyst.

Complete enumeration is possible when each input variable has only a finite numer of feasible values. In this case the simulation can be run for all possible combinations of input variable values. If there are no uncontrollable components in the model, this technique will assure the analyst of an optimal solution. When uncontrollable components are present, the analyst normally will attempt to optimize some expected return (e.g. minimize expected cost). Random variables are included in the model and a random number generator is used to determine values for these random variables. For problems of this nature, each set of input values can be run several times with different values from the random number generator. By choosing the number of replications at each input point, (set of controllable variable values) the analyst can approximate the expected value of the output at that point with any desired accuracy. The largest or smallest expected value is a good approximation to the optimum. Hence, if the set of feasible input values is finite, complete enumeration will yield at least a near optimal solution. Of course, if any input variable has infinitely many feasible values, complete enumeration is impossible.

Complete enumeration works best when the number of feasible values for each variable are both small. An example of the type of problem which may realistically be solved by complete enumeration is described in Schmidt, Taylor, and Bennet [5]. In their problem there is one controllable variable which has five possible values. Additionally, there are several random variables and a complicated objective function (expected cost) which must be minimized. Each of the five values of the input may be repeated several times in order to approximate the expected cost at that input value.

Random Search. Instead of running the program at every possible set of input values, as in complete enumeration, input values of the controllable variables are chosen at random and the simulation program is run at several of these values. It is obvious that such a technique will not guarantee an optimum, but as the number of computer runs is increased, a better approximation for the optimum is found.

When there are uncontrollable input variables, it is desirable to run several replications of the simulation at each set of controllable input values in order to get a good estimate of the mean output for a given set of controllable inputs. However, since with random search the analyst will not run the program for every possible set of controllable input values, he must decide whether it is better to get an accurate estimate on a few sets of controllable values or a less accurate estimate at more values.

Variations on complete enumeration. The techniques described here apply to problems with only a few feasible values of the controllable variables, but with uncontrollable variables as well. These techniques aid the analyst in determining when to stop replicating at a given set of input values.

All of these techniques utilize complete enumeration of the controllable variables, and they determine optimal expected value of the output through replication. Kleijnen, Naylor and Seaks [3] describe multiple ranking procedures, and Schmidt, Taylor, and Bennett [5] discuss an heuristic elimination procedure. The techniques described by Kleijnen, et al., allow the analyst to do a small sample at each input point, calculate a mean output value for each input point, and rank the mean outputs. With their procedure, the analyst can be asssured that the largest sample mean is the true largest output with some predetermined probability.

Schmidt, et al., have a less rigorous technique for determining the optimum. They run several replications of the simulation for each vector of input variables, compute confidence limits on the mean value of each output and compare them. The point with the highest lower limit is the sample maximum. If the intent is to maximize, the points whose confidence limits do not overlap with those of the sample maximum are eliminated from further consideration.

For those sets of inputs which remain, more sample values are determined and confidence limits are recomputed. This procedure is continued until only one set of input values remains or until the remaining vectors of alternative inputs have output values so close that the cost of future simulations is greater than the gain asssociate with the difference between the output values.

The usefulness of these techniques is limited by the fact that they are only practical on a certain class of problems. On this class of problems the techniques described are not only useful, but seem to be the only types of techniques with any chance of success.

Variation on Random search. This technique is reported by Luus and Jaakola [4]. The approach taken is to perform a sequence of random searches. After each search, the point with the highest values (assuming maximization) is retained. Then a further random search is run over a diminished range of input variables centered at the highest point from the previous search. The highest point is updated, the variable range is diminished and the process is continued. The greatest advantage of this technique is the combination of its simplicity and its apparent success on several problems (Luus and Jaakola describe six problems on which it is quite successful).

2.2 Techniques for Unimodal Objective Functions

For the techniques discussed below we must know (or assume) that the objective function is unimodal. We then apply nonlinear programming techniques to optimize this objective function. This section is divided into two parts. In part A we consider approaches to steepest ascent algorithms, and in part B techniques which do not require derivatives.

2.2.A Steepest Ascent

For steepest ascent, the function to be optimized is viewed as a surface in a Euclidean space of n+1 dimensions, where n is the number of controllable factors. If there are two controllable inputs, (x_1 and x_2), we can easily visualize the maximization problem as an attempt to find the highest point $f(x^*_1, x^*_2)$ on the surface and the vector (x^*_1, x^*_2) which yields that point.

The steepest ascent (descent in the minimization problem) algorithm has the following general sequence steps:

1. Begin at some point $x^0 = (x_1^0, x_2^0, \ldots, x_m^0)$.

2. Estimate the shape of the function near the point (x^0).

3. Determine the direction which would increase the value of the function, f(x), the fastest near x^0, if the approximation were the actual function (the steepest ascent direction).

4. Determine x^1, where $(x^1 - x^0)$ is a vector in the steepest ascent direction and compute $f(x^1)$.

5. Compute f(x) at points along the steepest ascent vector until the maximum (or an approximation to it) in that direction has been reached.

6. Return to step (2) using the maximum point in the previous steepest ascent direction as its starting point. The algorithm terminates when no further improvement can be made.

We now focus on steps two and three of the above algorithm. One must compute the value of the objective function at points near an intermediate solution point and use those points to fit a hyperplane which approximates the objective function at that intermediate solution point. The steepest ascent vector is the vector whose components are the derivatives of the hyperplane function with respect to the input variables. This is an approximation to the steepest ascent direction for the objective function.

We must consider methods of determining the "nearby points" at which to compute the objective function value (i.e. run the simulation). The literature of experimental design represents several methods for determining these points [8], [12], [13]. The four most common designs are full factorial [16], fractional factorial [16], [13], [21], [22], simplex [7], [8], [27], and central composite [16] [17].

2.2.B Techniques Without Derivatives

In this part, we discuss five nonlinear programming techniques for which no derivative need be approximated.

Coordinate search, the simplest of these, changes one variable at a time. Given some initial point $(X_1,...,X_n)$, the first coordinate is changed a little to $(X_1 - \delta, X_2..., X_n)$ and $(X_1 + \delta, X_2,...,X_n)$ and the simulation is run. If the objective function improves, the increment is icreased and the simulation run again, until a maximum in the X_1 coordinate is reached. Then the second input variable is considered, and so on. If any improvement has been found after considering all the variables, the new point becomes the starting point, and the process is repeated again. When a pass through all variables yields no improvement, the algorithm terminates. Lefkowitz and Schriber [33] describe a problem for which this technique is successful. However, it is easy to develop examples for which this algorithm would terminate at solutions which are nowhere near optimum.

Pattern Search [Hooke and Jeeves, 31] starts at an initial point and uses coordinate search for one pass through all the variables. The starting point (b_0) is subtracted from the final point (b_1) of the coordinate search to yield a "pattern" direction. A jump is made in the pattern direction and the new point (p) is tested for improvement. If there is improvement at p, a new coordinate search is conducted yielding b_3 (which may equal p), and $b_3 - b_2$ is the new pattern direction. If p was not an improvement the pattern is destroyed and a coordinate search is conducted at b_2. If b_2 is not improved a coordinate search is tried with smaller increments. This action is repeated until improvement is achieved and a new pattern is developed or until the increments reach some prescribed minimum. If the minimum is reached, the algorithm terminates. The difficulty found with coordinate search can occur here if the pattern is destroyed at a termination point of coordinate search.

Rotating Coordinates [40]. As the name implies, Rosenbrock's procedure changes coordinate systems so that the pattern direction becomes a coordinate direction and all other coordinate directions are orthogonal to the pattern direction. Then, during the coordinate search step, directional changes of a ridge can be readily determined. The algebra required to change coordinates is relatively simple, is called the Gram-Schmidt Orthogonolization, and can be found in many linear algebra books, as well as in Rosenbrock's paper [40]. Rosenbrock reported good success with this method on a difficult example problem.

Mugele's Ridge Follow [37]. This technique moves in a coordinate direction whenever that coordinate direction represents an increase and utilizes a pattern only when coordinate directions fail. Patterns for this technique do not increase like Hook and Jeeves, but when coordinate changes are unsuccessful, an alternative exists.

If all coordinate directions lead to diminished functional values, pick the two largest coordinate values found (call them vectors a^1 and a^2) and try the point midway between them $(a^1 + a^2)/2$. If that is still smaller than the original point, use the maximum of a quadratic approximation of the curve through those three points.

If this solution is not greater than the original we can try other pairs of coordinate points. If all pairs of points fail we are probably near a local optimum.

Razor Search. This technique, described for simulation by Nelson and Krisebergh [38], is a variation on pattern search described earlier. The technique is to use pattern search until it fails, make a random jump and use pattern search until it again fails. The two failure points are then used to generate a new pattern. The process is repeated until some terminating criterion is satisfied.

The principle advantage of this technique is that it can use the random jump to decrease the probability of terminating at a suboptimum.

2.3 Techniques for Multimodal Surfaces

For the two techniques presented in this section, the problem is viewed as a response surface in n+1 dimensional Euclidean space, where n is the number of controllable variables. These response surfaces have several local optima and the purpose of these two techniques is to find the global optimum of such a surface. Eldridge [41] uses a random factorial design and regression to divide the space into several unimodal surfaces which can then be optimized by whatever technique one wants to use. Hartman [42] suggests a method of determining several starting points for any search algorithm which yields a local optimum. Eldridge subdivides the space of inputs and searches each subspace, while Hartman merely spreads the starting points of his search to increase the chance of finding the global optimum.

Hartman. Hartman's procedure is to:

1. Describe the territory which has been searched.

2. Pick the starting point for each search "as far as possible" from the territory previously searched.

3. Modify the description of previously searched territory to include the newly searched territory.

4. Terminate a search if it reaches previously searched territory.

The first step includes partitioning the input variable values into disjoint sets a list of "searched" (third step) cells is generated and increased when each search is conducted. This approach assumes that the cells are small enough so that there is at most one local optimum in a cell. The cell sizes are generated by the user.

The third step is to determine what cells to add to the list of "searched cells". Most unimodal serch techniques involve a succession of linear searches; while searching along a line they ignore nearby optima. However, at the endpoint of a linear serch a new search direction is tried. Hence, cells which contain the endpoints of linear searches are added to the list, if they are not already there.

The final step involves determining when to stop a search. Hartman has three algorithms which differ in this step. Algorithm A1 completes each search to a local optimum; A2 terminates if it reaches a cell which as been found to contain a local optimum; and A3 stops if it enters any cell on the search list.

Eldridge. Eldridge's approach is to divide the sample space (set of possible input values) into regions with a single local optimum and to search in each of these regions. He uses "fractional random factorial designs" and regression to determine these regions. When analyzing the regression, if cubic or higher order effects are found in some region, the region is divided into smaller regions until a quadratic fits the function well. Since a quadratic can have at most one local optimum, it is reasonable to assume that each final region will have at most one local optimum.

His algorithm is as follows:

1. The simulation program is run at four values equally spaced in the interval for each variable (this includes the endpoints).

2. From these an analysis of variance is run and F values of linear, quadratic, and cubic effects are computed together with first order interactions.

3. If the cubic effect in either variable is significant, the range of that variable will be divided in half and the function will be evaluated at enough more points to determine a full factorial on each of the subranges.

4. The process of computing F values and dividing the space of input values is done until only quadratic or lower effects are sigificant. Once the above has been accomplished, each region is searched with a unimodal search technique, and the optimal value is the largest (smallest) of the several computed.

These two techniques allow a multimodal function to be searched to increase the probability of finding a global optimum.

2.4 Comparisons and Literature Reviews

The main source of information comparing techniques is Smith [50]. Brooks [43] and McArthur [48] did earlier studies referenced by Smith. Smith compared random search, coordinate, full factorial steepest ascent, and simplex steepest ascent. He found that as he increased the number of computer runs the factorial steepest ascent became relatively better than the others.

When the number of computer runs was small, random search was best, while for more computer runs the factorial design was most successful. It takes any steepest ascent technique a few runs to find a search direction, so one would expect it to start out slowly. Coordinate Search did poorly.

McArthur and Brooks on the other hand, found that for large numbers of factors, random search worked best. This may have been because not enough computer runs were used to allow steepest ascent to achieve the success that is possible.

A large scale literature review of optimization in simulation is contained in Farrell, McCall and Russell [44]. That paper is of monograph length, contains more complete discussions of all techniques described here, and contains one or two techniques not discussed here. Wilde [51] presents a few of the twchniques described here, as well as several other methods of optimization. However, Wilde makes no reference to simulation, and many of the techniques he describes do not apply to simulation. Similarly, nonlinear programming text books Mangasarian,[49], Zangwill,[52]discuss several of these techniques, but do not present any information about simulation.

3. BIBLIOGRAPHY

This bibliography is extensive, but certainly not exhaustive. The papers referenced in this bibliography are divided into sections and alphabetized by sections. However, they are numbered sequentially with the first article of section 3.1 numbered 1, that of 3.2.A numbered 6, and so on. Sections 3.1 through 3.4 are identical with sections 2.1 through 2.4. For instance, references to steepest ascent algorithms and experimental designs are found in sections 3.2.A. Sections 3.5 contains papers that can not be placed elsewhere. These include background information, applications papers and optimization techniques which don't fit any of the categories or are not directly applicable to simulation. Numbers followed by asterisks are those papers which are referenced in section 1 or 2.

3.1 Naive Techniques

1 BECHHOFER, R.E., DUNNETT, C.W., and SOBEL, M., "A Two-Sample Multiple Decision Procedure for Ranking Means of Normal Populations with a Common Unknown Variance," Biometrika, Vol. 41 (1954), pp. 170-176.

2 CLOUGH, Donald J., "An Asmptotic Extreme-Value Sampling Theory for Estimation of a Global Maximum", Canadian operations Research Society Journal (Canada), Vol. 7, No. 2 (July, 1969), pp. 102-115.

3* KLEIJNEN, Jack P.C., NAYLOR, Thomas H., and SEAKS, Terry G., "The Use of Multiple Ranking Procedures to Analyze Simulations of Management Systems: A Tutorial," Management Science, Vol. 18 (February 1972), p. B245-B257.

4* LUSS, Rein, and JAAKOLA, T.H.I., "Optimization by Direct Search and Systematic Reduction of the Size of the Search Region," American Institute of Chemical Engineers Journal, Vol. 19, No. 4 (July, 1973), pp. 760-766.

5* SCHMIDT, J.W., TAYLOR, R.E., and BENNETT, G.K., "A Heuristic Simulation Method for Optimum Selection of Alternatives Through Simulatin," Simulation, Vol. 21, No. 5 (November, 1973), pp. 129-134.

3.2 Techniques for Unimodal Surfaces

3.2.a. Steepest Ascent

6 ANSCOMBE, F. J., "Quick Analysis Methods for Random Balance Screening Experiments," Technometrics, Vol. 1, No. 2 (May, 1959), pp. 195-209.

7* BOX, G.E.P., "Multi-factor Designs of First Order," Biometrika, Vol. 39 (1952), pp. 49-57.

8* BOX, G.E.P., and BEHNKEN, D.W., "Simplex-Sum Designs: A Class of Second-Order Designs Derivable from Those of First Order," Annals of Mathematical Statistics, Vol. 31 (1960), pp. 838-862.

9 BOX G.E.P., and HUNTER, J.S., "Experimental Designs for the Exploration and Exploitation of Response Surface," in Chew, Victor, ed., Experimental Designs in Industry, New York, John Wiley & Sons, (1958), pp. 138-190.

10 BUDNE, T. A., "The Application of Random Balance Design", Technometrics, Vol. 1, No. 2 (May, 1959), pp. 139-155.

11 CHEW, Victor, Experimental Designs in Industry, John Wiley & Sons, Inc. New York, New York (1958).

12* COCHRAN, W.G., and COX, G.M., Experimental Designs. New York, John Wiley & Sons, (1957).

13* DAVIES, O.L., ed., Design and Analysis of Industrial Experiments, New York, Hafner Publishing Co., (1960).

14 DEBAUN, R., "Response Surface Design for Three Factors at Three Levels," Technometrics Vol. 11, No. 3., (August, 1969), pp. 461-475.

15 HADER, F.J., KARSON, J.J., MANSON, A.R., "Minimum Bias Estimation and Experimental Design for Response Surfaces", Technometrics, Vol. 11, No. 3., (August, 1969), pp. 461-475.

16* HUNTER, J.S., "Some Applications of Statistics to Experimentation," Chemical Engineering Progress Symposium Series, American Institute of Chemical Engineers, Vol. 56, No. 31 (1960), pp. 10-26.

17* HUNTER, J.S., and NAYLOR, T.H., "Experimental Designs for Computer Simulation Experiments," Management Science, Vol. 16 (March 1970), pp. 422-434.

18 IGNALL, Edward J., "On Experimental Designs for Computer Simulation Experiments," Management Science, Vol. 18, No. 7, (March, 1972), pp. 384-388.

19 LIAU, T.L., HARTLEY, H.O., SIELDEN, R.L., Jr., "Confidence Regions for Global Optima in Nonlinear Programming", Technical Report No. 43, Texas A & M University, (August, 1973).

20 MANDEL, J., "A Method of Fitting Empirical Surfaces to Physical and Chemical Data", Technometrics, Vol. 11, No. 3, (August 1969), pp. 411-429.

21* NATIONAL BUREAU OF STANDARDS, Fractional Factorial Experiment Designs for Factors at Two Levels, Applied Mathematics Series, No. 48, U.S. Government Printing Office, Washington 25, D.C. (1957).

22* NATRELLA, Mary G., Experimental Statistics, National Bureau of Standards Handbook 91, U.S. Government Printing Office, Washington, D.C., (1966).

23 SATTERTHWAITE, F.D., "Random Balance Experimentation," Technometrics, Vol. 1, No. 2 (May 1959), p. 111.

24 SMITH, Dennis E., "An Optimizer for Use in Computer Simulation: Background and Design Concepts," HRB-Singer Report No. 4352.22-R-1. (AD 708-765), (July, 1970).

25 SMITH, Dennis E., "An Optimizer for Use in Computer Simulation: Studies with a Prototype," HRB-Singer Report No. 4352.11-R-2. (AD 730-086), September, 1971).

26 SMITH, Dennis E. and STORCK, C. Edward, "Research on an Optimizer Computer Program for Use In Simulation Studies," NTIS, (AD766-089/7WC), (August, 1973).

27* SPENDLEY, W., HEXT, G.R., and HIMSWORTH, F.R. "Sequential Application of Simplex Designs in Optimization and Evolutionary Operations," Technometrics, Vol. 4, No. 4 (November, 1962), pp. 441-461.

28 YOUDEN, W. J., KEMPTHORNE, Oscar, TUKEY, J.W., BOX. G.E.P., HUNTER, J.S., "Discussion of the Papers of Messrs. Satterthwaite and Budne," Technometrics, Vol. 1, No. 2, (May, 1959), pp. 156-193.

3.2.B Techniques Without Derivatives

29 BILLES, W.E., "A Gradient-Regression Search Procedure for Simulation Experimentation", Proceedings of the 1974 Winter Simulation Conference, Vol. 2, pp. 491-497.

30 DRAPER, Norman, "Ridge Analysis of Response Surfaces", Technometrics, Vol. 5, No. 4, (November, 1963), pp. 469-479.

31* HOOKE, R., and JEEVES, T.A., "A 'Direct Search' Solution of Numerical and Statistical Problems," Journal of the Association for Computing Machiner., Vol. 8, No. 2 (April, 1961), pp. 212-229.

32 KARR, H.W., LUTHER, E.L., MARKOWITZ, H.M., and RUSSELL, E.C., "SimOptimization Research Phase I," (November, 1965), CACI, Inc., (Sponsored by Office of Naval Research).

33* LEFKOWITS, Robert M., and SCHRIBER, Thomas J., "Use of an External Optimizing Algorithm with a GPSS Model," Proceedings of the 1971 Winter Simulation Conference pp. 162-171.

34 LEININGER, G.C., and MARTENS, H.R., "The Determination of Digital Simulation Models for Continuous Systems by Direct Search Minimization", Proceedings of the 23rd Association for Computing Machinery Conference, (1968), pp. 251-258.

35 LUTHER, E.L., and MARKOWITZ, H.M., "SimOptimization Research Phase II," CACI, Inc., (July 1966).

36 LUTHER, E.L., and WRIGHT, N.H., "SimOptimization Research Phase III," CACI, Inc., (1969).

37* MUGELE, R.A.,"A Nonlinear Digital Optimizing Program for Process Control Systems", Proceedings of Western Joint Computer Conference, (1962).

38* NELSON, Carl. W., and KRISBERGH, Harold M. "A Search Procedure for Policy Oriented Simulations: Applications to Urgan Dynamics," Management Science, Vo. 20, No. 8 (April 1964), pp. 1164-1174.

39 POWELL, M.J.D., "An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives,", Computer Journal, Vol. 7, No. 2, (1964) pp. 115-164.

40* ROSENBROCK, H.H., "An Automatic Method for Finding the Greatest or Least Value of a Function," Computer Journal, Vol. 3, No. 3 (October 1960), pp. 175-184.

3.3 Techniques for Multimodal Surfaces

41* ELDRIDGE, David L., "Experimental Optimization of Statistical Simulation", Proceedings of the 1974 Winter Simulation Conference, pp. 503-510.

42* HARTMAN, James K., "A New Method for Global Optimization," Paper NPS55HH73041 A, Naval Postgradujate School, Monterey, California, (April, 1973).

3.4 Comparisons and Literature Reviews

43* BROOKS, S.H., "A Comparison of Maximum-Seeking Methods," Operations Research, Vol. 7 (1959), pp. 430-457.

44* FARRELL, W., C. McCALL, and E. RUSSELL, Optimization Techniques for Computerized Simulation Models, Report to Office of Naval Research NTIS #AD-A011 844/8GI (1975).

45 GOTTFRIED, Byron S., WEISMAN, Joel, Introduction to Optimization Theory, Prentice Hall, Englewood Cliffs, New Jersey, (1973).

46 HILL, W.J., HUNTER, William G., "A Review of Response Surface Methodology: A Literature Survey", Technometrics, Vol. 8, No. 4, (November, 1966), pp. 571-590.

47 LASDON, Leon S., Optimization Theory For Large Systems, New York, The Macmillan Company, (1972).

48* MCARTHUR, D.S., "Increasing the Efficiency of Empirical Research," Esso Research and Engineering Company, Report No. RL34M61, (1961).

49* MANGASARIAN, Olvi L., Nonlinear Programming, New York, McGraw-Hill, (1969).

50* SMITH, Dennis E., "An Empirical Investigation of Optimum-Seeking In the Computer Simulation Situation," Operations Research, Vol. 21, No. 2 (March-APril, 1973), pp. 475497.

51* WILDE, Douglas J., Optimum Seeking Methods, Englewood Cliffs, N.J., Prentice Hall, (1964).

52* ZANGWILL, Willard I., Nonlinear Programming: A Unified Approach, Englewood Cliffs, N.J., Prentice Hall, (1969).

3.5 Miscellaneous

53 BLATNY, J., CLARK, S.R., ROURKE, T.A., "On the Optimization of Performance of TimeSharing Systems by Simulation", Communications of the ACM, Vol. 15, No. 6, (June, 1972).

54 BURT, John M., Jr., GARMAN, Mark R., "A Simulation Technique for Stochastic Network Analysis", Management Science, Vol. 18, No. 3 (November, 1971), pp. 207-217.

55 CRANE, M.A., and IGLEHAET, D.L., "Simulating Stable Stochastic Systems I: General Multi-Server Queues," Journal of the Association for COmputing Machinery, Vol. 21 (1974), pp. 103-113.

56 CRANE, M.A., and IGLEHART, D.L., "Simulating Stable Stochastic Systems II: Markov Chains," Journal of the Association for Computing Machinery, Vol. 21 (1974), pp. 114-123.

47 CRANE, M.A., and IGLEHART, E.L., "Simulating Stable Stochastic Systems III: Regenerative Processes and Discrete-Event Simulations," Operations Research, Vol. 23, No. 1 (JanuaryFebruary 1975), pp. 33-45.

58 EMSHOFF, James R., SISSON, Roger L., Design and Use of Computer Simulation Models, MacMillan Company, New York, (1970).

59 FISHMAN, G.S., Concepts and Methods in Discrete Event Digital Simulation, New York, Johyn Wiley & Sons, (1973).

60 FISHMAN, G.A., "Problems in the Statistical Analysis of Simulation Experiments: The Comparison of Means and the Length of Sample Records," Communications of the Association for Computing Machinery, Vol. 10 (February 1967), pp. 94-99.

61 FORRESTER, J.W., Urban Dynamics, Cambridge, Massachusetts, M.I.T. Press, (1969).

62 MAISEL, H., GNUGNOLI, G., Simulation of Discrete Systems, Science Research Associates, Inc., Palo Alto, California, (1972).

63 MEIER, R.C., NEWELL,W.T., PAZER, H.L., Simulation in Business and Economics, Prentice Hall, Englewood Cliffs, New Jersey, (1969).

64 NAYLOR, T.H., ed., The Design of Computer Simulation Experiments, Duke University Press, Durham, North Carolina, (1969).

65 REITMAN, J., Computer Simulation Applications, John Wiley & Sons, Inc., New York, (1971).

66 ROWLAND, James R., "Optimal Digital Simulations for Random Linear Systems with Integration Constraints", Computers and Electrical Egnineering, Vol. I, No. 1, (June, 1973), pp. 111-118.

67 SHUBIK, Martin, "Simulation of the Industry and the Firm", American Economic Review, No. 5 (December 1960), p. 909.