

SIMULATION OPTIMIZATION USING RESPONSE SURFACES BASED ON SPLINE APPROXIMATIONS

Andrew F. Daughety

Mark A. Turnquist

ABSTRACT

This paper presents an approach designed to increase the efficiency and utility of search for optima of simulation models. Specifically, spline functions (odd-order polynomials fitted between simulation run outputs that match curvature at the end points) are used to approximate the simulation along suitably chosen directions of search. The splines are used to generate "pseudo-experiments" which enrich the data base formed from actual simulation runs. An overall (grand) function is then fit to this data base, yielding new direction(s) of search for the next iteration. Several characteristics of this technique are examined, including its sensitivity to experimental budget, number of iterations allowed, and size of feasible region.

This approach results in not only an estimate of the optimal response from the simulation, but also a response surface estimated over a larger domain useful both for sensitivity analysis and in some cases as an approximate representation of the simulation for use in other modeling efforts. The paper describes an application of the technique to a model of a railroad classification yard. The objective is to find the numbers and sizes of inbound and outbound trains, and the dispatching policy within the yard, which minimize total car delay.

I. INTRODUCTION

The use of simulation models for purposes of system optimization is a topic of considerable interest to many analysts. Most approaches to this problem have been based on the use of direct search procedures originally developed primarily for non-linear programming applications. Examples of such procedures are the gradient-projection method described by Rosen (14,15), the simplex-based procedure of Spendley, et al (16), and the "complex" (constrained simplex) approach described by Box (3). More recent developments in non-linear programming have lead to other types of procedures. These newer algorithms, however, often are constructed so as to exploit various elements of structure in non-linear programming problems, and have not been generally applied to a simulation environment. A concise summary of the state-of-the-art in simulation optimization is provided by Farrell (6).

Because these procedures were originally designed to be used in an environment where evaluation of the objective function to be optimized is not costly, they tend to require a large number of such evaluations. For example, Box (3) describes application of his procedure to a problem with five controllable inputs, or decision variables. Locating the point of near-optimum response required 881 evaluations of the objective function. While the number of such evaluations required in any particular application depends on a number of factors, including how good a starting solution is available, it is clear that procedures of this type generally require a substantial number of function evaluations.

This becomes a problem when applying these procedures to simulation models. In this environment, each evaluation of the objective function corresponds to a "run" of the simulation model to generate an experimental outcome. In many cases, this is expensive in terms of both computer time and analyst time. Thus, procedures which in concept are directly applicable to a simulation environment are often too computationally demanding to be really usable.

A premise on which this paper is based is that the problem of optimum-seeking with simulation models needs to be redefined to be more in keeping with the objectives of the user of the simulation results. This redefinition leads to use of a modified set of analytic tools for solution of the problem. Section II of the paper discusses problem definition and proposes a new procedure for solution. The sensitivity of this procedure to its parameters is described in Section III. Application of the procedure to a realistic simulation problem is described in Section IV; this application involves a model of a railroad classification yard. Finally, conclusions and recommendations for further work are given in Section V.

II. DEFINITION OF THE OPTIMIZATION PROBLEM

The problem addressed by most direct-search procedures is to find the point (or at least a very small region containing the point) of optimum response, possibly constrained by bounds on the input variables and/or more complicated constraints involving several of the input variables. Notably missing from this problem definition are notions of available budget for attainment of the solution, and sensitivity of the objective function to changes in the input variables over some region about the optimum solution.

Several applications of direct-search procedures to simulation have attempted to deal with the problem of response sensitivity by fitting response surfaces (usually quadratic forms) to the experimental observations in the neighborhood of the estimated optimum point. Examples of such procedures are discussed by Biles (1), Eldredge (5), and Mihram (11), among others. A general discussion of response-surface methodology is given by Myers (12).

It should also be noted that such an estimated response surface can be useful for purposes other than local sensitivity analysis. If the response surface is estimated over a sufficiently large domain, it can often be used as a replacement for the simulation model itself, for some purposes. For example, a simulation of some component of a larger system gives detailed information on its behavior; but if questions pertaining to the larger system as a whole are of interest, it may be sufficient to use an analytic approximation to the simulation model as part of the overall analysis. Such an application, in fact, has served as motivation for the work presented here. In this case, we wish to approximate the response of a simulation of railroad-yard performance for use in a larger model of the rail firm as a whole.

Concern with the available budget for experimentation is quite rare in the literature. This becomes a serious problem when the model in question has a large number of input variables. In view of this, we have sought to solve a problem which might be expressed roughly as follows:

Given a budget of N experiments and an initial feasible region, determine a sub-region of minimum volume containing the optimal solution, and estimate a response surface over a sub-region at least as large as that predicted to contain the optimum.

As one attempts to refine this problem statement into more precise terms, a number of important questions are raised. Must the initial feasible region be convex? Must the response function be unimodal? Will the identified sub-region contain the optimal solution with certainty? Over how large a sub-region should a response surface be estimated?

The work presented here does not provide answers to all of these questions, but it does attempt some initial steps toward development of a procedure which can be used to solve problems of this type. The particular formulation developed for this work is stated more precisely in the next section.

GENERAL PROCEDURE AND ASSOCIATED OPTIMIZATION PROBLEMS

In the analysis that follows, we shall assume that the response surface φ is continuous. Then in any particular direction we know that φ can be approximated, on an interval, arbitrarily closely by a polynomial. While it is no doubt more realistic to assume that the response surface is discontinuous, little can be done to approximate such a surface at reasonable expense. We will attempt to build up a

map of the region around the optimum based on the continuity assumption.

For the rest of the paper we will employ the following notation:

- n - number of input variables
- $x \in R^n$ - vector of input variables
- $\varphi: R^n \rightarrow R$ - unknown relation between inputs, x , and output, $\varphi(x)$
- $f: R^n \times R^l \rightarrow R$ - specified function, to be estimated, to approximate $\varphi(x)$. This will be typically written as a quadratic in x :

$$f(x; \alpha^k) = \alpha_{00}^k + \sum_{i=1}^n \alpha_{i0}^k x_i + \sum_{i=1}^n \sum_{j=1}^n \alpha_{ij}^k x_i x_j$$

with $l = \frac{(n+1)(n+2)}{2}$. Thus,

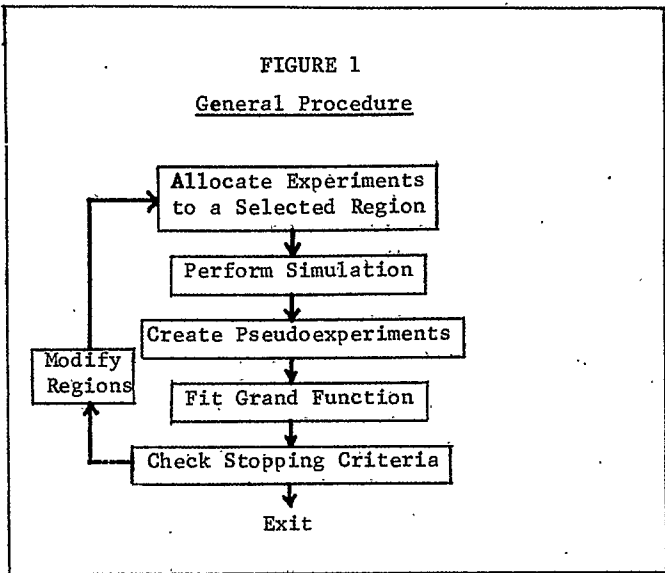
- $f(x; \alpha^k)$ would be the estimated approximation at stage k .
- $S_{im}^k: R \rightarrow R$ - m -th spline function for input i at stage k
- $\| \cdot \|_p$ - norm. For functions g , $\|g\|_p = \left[\int |g|^p \right]^{1/p}$, while for vectors x , $\|x\|_p = \left[\sum_j |x_j|^p \right]^{1/p}$
- x^t, x_i^t - t -th vector and i -th element of t -th vector, respectively.
- $\$ \subset I$ - subset of the index set of the input vectors that provides the index set of integer variables. Thus, $i \in \$$ means that x_i is an integer variable.
- \underline{x}, \bar{x} - lower and upper bounds on x . We assume $\underline{x} \geq 0$.
- x^{*k} - optimizer of $f(x; \alpha^k)$

The general procedure employed is shown in Figure 1. We will use this diagram and the notation above to explain the procedure. Given a maximum number of experiments, N , to be performed, and a simulation model that produces an output $\varphi(x^k)$ for a given input x^k , we wish to find the solution to the following problem:

$$\min_{\alpha} \sum_{i=1}^N (f(x^i; \alpha) - \varphi(x^i))^2$$

In summary, the steps at stage k are to (1) allocate some of the experiments (i.e., find suitable x^k) to a feasible region; (2) run the simulation on these experiments; (3) create "pseudoexperiments" (from the splines S_{im}^k) as interpolations among the true ones run in (2) above; (4) find an α^k for the grand function $f(x; \alpha)$; (5) find x^{*k} , which either indicates termination ($\|x^{*k} - x^{*(k-1)}\| < \epsilon_x$) or provides a new starting point for the $k+1$ stage. Other stopping criteria are $\|f(x; \alpha^k) - f(x; \alpha^{(k-1)})\| < \epsilon_f$ or the total number of allowable experiments (N) has been exhausted. We examine each of these problems in turn, below.

FIGURE 1
General Procedure



Allocation of Experiments

Given, at stage k , some number of experiments we can run (say, $N_k \leq N$), we would like to allocate them to the stage k experimental region F^k :

$$F^k = \{x: x \in R^n, \underline{x}^k \leq x \leq \bar{x}^k\}$$

in a way so as to observe as much of the response surface ϕ as possible. The obvious solution is to arrange the N_k points so as to maximize the minimum distance between any two points; this will spread them out as far from each other as possible. Thus we might wish to solve the following problem (given a specific norm, say L_1 or L_2 — i.e., $p=1$ or 2):

$$\max_x \min_{i,j} \|x^i - x^j\|_p$$

$$\text{S.T. } \underline{x}^k \leq x^i \leq \bar{x}^k \quad i=1, \dots, N_k$$

This formulation is incomplete in two ways. First, we may require some of the x_j^i to be integer. This is because some of the inputs to the simulation model may be integer in nature (e.g. number of telephone lines in an airline-ticketing simulation, or number of trains in a railroad-yard simulation). Second, because of the costliness of simulation runs, we will generate interpolative experiments — pseudoexperiments via a polynomial approximation technique to be discussed in the next subsection. In order to take advantage of this technique the experiments must be arranged on axes so that the approximation can be solved for one variable at a time. The axes simply amount to a Cartesian coordinate system with an origin in F^k . Thus, let \hat{x} be some point in F ; for example, \hat{x} might be $\bar{x}^{*(k-1)}$. Then we pose our allocation problem (P1) (again, for a specific p , \hat{x} and N_k) as follows:

$$(P1) \quad \max_{x,y} \min_{i,j} \|x^i - x^j\|_p$$

S.T.

$$\begin{aligned} (1) \quad & y_\ell^i \hat{x}_\ell \leq y_\ell^i x_\ell^i \leq y_\ell^i \bar{x}_\ell & \ell=1, \dots, n \\ (2) \quad & \sum_{\ell=1}^n y_\ell^i = n-1 & i=1, \dots, N_k \\ (3) \quad & y_\ell^i: 0 \text{ or } 1 & \ell=1, \dots, n \\ (4) \quad & \underline{x}^k \leq x^i \leq \bar{x}^k \\ (5) \quad & x^i: \text{integer} \quad i \in \mathcal{I} \end{aligned}$$

This is a non-linear integer programming problem. The first three constraints use a zero-one variable y to force the x -variables onto a coordinate system through the point \hat{x} . One can picture the problem as a rectangular solid described by \underline{x}^k and \bar{x}^k with the point \hat{x} inside it. One then would create axes through \hat{x} until they intersected the walls of the solid. Problem (P1) locates N_k points on this axis system so as to maximize the minimum distance between any two points.

Needless to say, at first glance (P1) is a very difficult problem to solve. In fact, a direct attack would undoubtedly be very expensive. There is, however, a very simple but elegant way to find the optimal solution to the above problem, using an algorithm by R. Chandrasekaran (4). For convenience we will refer to \hat{x} as a hub and to each piece of axis sticking out of \hat{x} as a spoke. Thus, there are $2n$ spokes of length $\bar{x}_i^k - \hat{x}_i$ or $\hat{x}_i - \underline{x}_i^k$ — i.e., always non-negative in length. Let λ be the distance between points and $N(\lambda)$ be the number of points that can be placed on the spokes when λ is the separation distance. Now consider the following algorithm (for the case $p=1$, i.e., $\Sigma |\cdot|$):

1. Pick λ
2. Find the largest spoke. If it is greater in length than λ , place a point on the end of the spoke and cut off λ . Continue with the spokes until all are $< \lambda$ in length.
3. Now arrange the leftover spokes from largest to smallest. Find spoke j such that $b_j + b_{j+1} > \lambda$ and $b_{j+1} + b_{j+2} < \lambda$, where b_j is the reduced length of spoke j . Place $j+1$ points on the outside of the $j+1$ spokes. Thus, the total number of points is:

$$N(\lambda) = \sum_{i=1}^{2n} \left\lfloor \frac{a_i}{\lambda} \right\rfloor + j+1$$

where $a_i = \bar{x}_i^k - \hat{x}_i$ or $\hat{x}_i - \underline{x}_i^k$ as appropriate, and $\lfloor u \rfloor$ is the greatest integer less than or equal to u .

If $N(\lambda) > N_k$, then λ is too small and should be increased; while if $N(\lambda) < N_k$, then λ is too large and should be decreased. There are two possible results:

1. $\exists \lambda^* \ni N(\lambda^*) = N_k$
2. $\exists \lambda^* \ni N(\lambda^{*-}) > N_k > N(\lambda^{*+})$

$$\text{where } \lambda \in [\lambda^{*-}, \lambda^{*+}]$$

$$\text{and } |\lambda^{*+} - \lambda^{*-}| < \epsilon \text{ for some small } \epsilon.$$

The first case is obvious. In the second case there are multiple optima, so that by selecting the λ yielding the larger N (i.e., $N(\lambda^{*-})$) and then dropping any $N(\lambda^{*-}) - N_k$ points, we have the optimal solution. [Note that for integer variables, cuts of size $\lfloor \lambda \rfloor + 1$ must be used.] In the case of norms other than $p=1$, step 3 must be modified in terms of computing $l_j + l_{j+1}$.

It should be remembered that the allocation of points was based on spending N_k experiments (of the total N experiments) on stage k exploration. Thus, we have to allocate the N experiments across the stages. Consider the sequence $\{b_i\}_1^T$ such that $l \geq b_i > 0$. Let v be the total sum of the sequence, i.e.:

$$v = \sum_{i=1}^T b_i$$

If we let $N_i = \frac{b_i}{v} N$, then $\sum N_i = N$. Now the selection of b_i should be made optimally. Unfortunately, optimality would require perfect foresight of all the possible simulation results, an impossibility.

One possible sequence is suggested by observing that at each stage x^{*k} will, in general, be in one of the 2^n orthants around x^k , the starting point. Thus, one possible sequence is

$$b_i = \frac{1}{2^{m(i-1)}}$$

If T is the total number of stages, we get a decreasing number of experiments allocated to each succeeding region F^k . If V is the volume associated with the original starting region, then after k stages we have a region of uncertainty of size $V/2^{mk}$.

Unfortunately, while this seems to have a natural appeal, for a reasonable budget (50-100 experiments) and a few variables (say, 5), such a sequence loads almost all the experiments on the first stage, providing limited information (no cross effects $x_i x_j, j \neq i$). This provides a rationale for using a sequence which decreases less rapidly, such as $b_i = 1/i$. This sequence has been used in the tests reported here. While this does not affect the size of the region of uncertainty, it has the effect of using the experiments more slowly, preserving a larger number for later stages.

Thus, given a total number of stages (which is equivalent to a projected region of uncertainty), a starting point x^1 and initial upper and lower bounds, \bar{x}^1 and \underline{x}^1 , and a total number of experiments, N , the allocation stage computes N_k (experiments to be run at each stage) and (as each stage occurs) where they should be located relative to an updated starting point x^k (e.g., $x^{*(k-1)}, k > 1$), and an updated region F^k (found by using the quadrant in which $x^{*(k-1)}$ occurred). The collection of experiments from a stage, $\{(x^k)_i\}_{i=1}^{N_k}$ are then sent to the simulation model which produces an output sequence $\{(\varphi(x^k))_i\}_{i=1}^{N_k}$.

Generating Pseudoexperiments

The purpose of generating pseudoexperiments is to get a little something for almost nothing. Relative to the cost of simulation runs, it costs very little to use any of various polynomial approximations techniques to generate extra points and their approximated outputs. What such points buy is that they allow the actual runs to provide a little more information which might not be captured otherwise. As we have assumed φ to be continuous, it is therefore continuous in the direction of the axes discussed in the last section. Our approach will be to approximate φ along each axis. We will then use the approximation to construct the pseudoexperiments on each axis. It is this extra information, captured by the polynomial approximation, that we will add to our optimum-seeking activity, to be described below.

The specific type of approximation to be used is known as a spline function (9). Given a strictly increasing sequence of real numbers, u_1, \dots, u_n , a spline function $S(u)$ of degree m with knots u_1, \dots, u_n is a function defined on the entire real line such that

- (1) in each interval (u_i, u_{i+1}) $i=0, \dots, n$ (with $u_0 = -\infty$ and $u_{n+1} = \infty$), $S(u)$ is given by some polynomial of degree m or less;
- (2) $S(u)$ and its derivatives of orders $1, \dots, m-1$ are continuous everywhere.

Natural splines (splines of odd degree $2q-1$) are splines wherein the polynomials on $(-\infty, u_1)$ and (u_n, ∞) are of degree $q-1$. In particular, cubic natural splines require $S''(u_1) = S''(u_n) = 0$.

The reason for using spline functions, instead of some other approximation, is their smoothness. Simply fitting a polynomial to a number of data points can often (for high enough degree) result in severely undulating curves. Let $p(u)$ be any polynomial which (1) agrees with the function to be approximated at the knots and (2) has continuous second derivatives on $[u_1, u_n]$. Then it can be shown that there exists a spline $\hat{S}(u)$ such that

$$\|S''(u)\|_2 \leq \|p''(u)\|_2 \quad (\text{on } [u_1, u_n])$$

i.e., $S(u)$ is the smoothest interpolation.

At stage k we find M_i splines $S_{im}^k(x_i), m=1, \dots, M_i$, for axis i , where M_i+1 is the number of distinct points on axis i . These splines are then interpolated at each interval midpoint, yielding M_i pseudoexperiments per axis. Integer axes are interpolated only if their length exceeds one, and then only at integer midpoints. The true experiments and outcomes $\{(x^t)_i, (\varphi(x^t))_i\}_{i=1}^{N_t}$ and the pseudoexperiments and outcomes $\{(\hat{x}^t)_i, (\hat{\varphi}(\hat{x}^t))_i\}_{i=1}^{N_t}$, all for $t=1, \dots, k$, are transmitted to the next phase, where a specified multivariate function is fit to them ($f(x; \alpha)$). This is a "grand function" which will be used to redirect the search for the region containing the optimum.

Generating a Grand Function

At this point we have for stage k true experiments and outcomes $\{(x^k)^i, (\varphi(x^k))^i\}_{i=1}^{N_k}$ and pseudo-experiments and outcomes $\{(\hat{x}^k)^i, (\hat{\varphi}(x^k))^i\}_{i=1}^{N_k}$. For convenience of notation, we will drop the \sim and let the total number of points be $s_k = N_k + N_k$. Let $f(x; \alpha)$ be the grand function, e.g.,

$$f(x; \alpha) = \alpha_{00} + \sum_i \alpha_{i0} x_i + \sum_i \sum_j \alpha_{ij} x_i x_j$$

where $\alpha = (\alpha_{00}, \alpha_{10}, \dots, \alpha_n, \alpha_{11}, \dots, \alpha_{n1}, \dots, \alpha_{nn})'$ is of length $(n+1)(n+2)/2$. Obviously, other more complicated functions are equally admissible. The quadratic function simply provides a convenient example. To calibrate the grand function, we solve the following problem.

$$\min_{\alpha} \sum_{l=1}^{t_k} (f(x^l; \alpha) - \varphi(x^l))^2$$

where $t_k = s_k + t_{k-1}$, i.e. we are using all points (true and pseudo) generated up to and including stage k ($t_0 = 0$). The solution is $\hat{\alpha}^k$. In stage one, since we have no points to estimate $\alpha_{ij}^{i \neq j}$; $i, j \geq 1$, we only estimate the main effects and the own-second order effects (α_{ii}).

Of course, since the observations are not independent (due to the presence of the pseudoexperiments) the estimator $\hat{\alpha}^k$ has none of the usual desirable properties. To improve this situation we elected to, upon meeting a stopping criterion (see next section), reestimate α after deleting all pseudo experiments. While this does not eliminate all statistical problems, it would seem that the remaining ones are second-order in nature.

Checking Stopping Criteria: Considerations for Further Optimum Seeking Activity

Finally, given $f(x; \hat{\alpha}^k)$ we can find the solution x^{*k} to the following problem:

$$\min_x f(x; \hat{\alpha}^k) \quad x^l \leq x \leq \bar{x}^l$$

The experimental region for stage $k+1$, denoted R^{k+1} , is determined once x^{*k} has been found. The upper and lower bounds for the elements x_i^{k+1} may be expressed as follows:

$$x_i^{k+1} = \begin{cases} \bar{x}_i^1 & , x_i^{*k} < \bar{x}_i^k \\ x_i^k & , \bar{x}_i^k \leq x_i^{*k} \leq \hat{x}_i^k \\ \hat{x}_i^k & , \hat{x}_i^k < x_i^{*k} \end{cases}$$

$$x_i^{k+1} = \begin{cases} \hat{x}_i^k & , x_i^{*k} < \hat{x}_i^k \\ x_i^k & , \hat{x}_i^k \leq x_i^{*k} \leq \bar{x}_i^k \\ \bar{x}_i^1 & , \bar{x}_i^k < x_i^{*k} \end{cases}$$

This allows the experimental region to be reduced if the optimal value at stage k is within the stage k region, and re-expanded if the optimum value at stage k is outside the region of experimentation at that stage.

There are a number of potential criteria for stopping:

1. Running out of experiments; i.e.

$$\sum_{i=1}^k N_i = N$$

2. A norm test; i.e. given ϵ_x or ϵ_f :

$$\text{i) } \|x^{*k} - x^{*(k-1)}\| < \epsilon_x$$

$$\text{ii) } \|f(x; \hat{\alpha}^k) - f(x; \hat{\alpha}^{(k-1)})\| < \epsilon_f$$

In general, one would expect condition (1) to occur before either 2 (i) or 2 (ii). Furthermore 2(ii) would undoubtedly be expensive to compute. Thus, an appropriate strategy would be to check 2(i) if k

$\sum_{i=1}^k N_i < N$. If stopping criteria are not met, then $\hat{x}^{k+1} = x^{*k}$ and we return to the allocation phase with \hat{x}^{k+1} as the new starting point, or hub.

Extending the Approach

In general one would expect that the region for optimization should be convex. Thus, nonlinear constraints can be added to the extent that convexity is not violated. The problem with nonconvexity can be viewed in three cases.

1. If the nonconvexity amounts to convex, disconnected sets, then if these are well known beforehand one could run each region separately, and use the separate functions to describe a piecewise approximation to the simulation.
2. If the constraints are nonconvex, then local optima may result causing confusion for the user unless they are enumerated. This can be a costly process.
3. If the nonconvexity stems from holes in the feasible region, an entirely new problem arises in that allocating experiments without accounting for the holes may produce pointless and wasted results.

Often, nonconvexities in practical applications are a mixture of the above. The best that can be said

is that one could still use the technique proposed above, screening for infeasible true or pseudo experiments, and settle for a local optimum and associated grand function.

Thus if the feasible region reflects nonlinear constraints, these may be incorporated into the simulation allocation problem (P1) by solving for the appropriate (\underline{x}, \bar{x}) values that reflect the constraints. The nonlinear constraints then also become part of the problem of finding x^{*k} .

III. TESTING THE TECHNIQUE

PROGRAM DESCRIPTION

The program follows the pattern illustrated in Figure 1. The spline functions are determined by accessing a standard spline routine in the International Mathematical and Statistical Library (IMSL). The grand-function fitting and optimization is performed using the Generalized Reduced Gradient 2 (GRG2) package.

The entire program is written in FORTRAN. Rather than strip out parts of the GRG2 package that were irrelevant to our problem, we accessed the entire package as a subroutine. The result is that our run times are longer than one would expect for a production routine. The GRG2 package used approximately 90% of the run time (excluding simulation.)

TESTING ON A KNOWN FUNCTION

In order to test the technique, a "simulation model" was developed as the following function:

$$\varphi(x_1, x_2) = 10(1 - \exp(-1.12 + .0462x_1 + .0588x_2 - .0014x_1^2 - .0014x_2^2 - .0014x_1x_2))$$

This function has a minimum at $x_1 = 8$ and $x_2 = 17$. The function is not convex, but it is pseudoconvex (10). A function is pseudoconvex on R^n if the following holds:

$$(y-x)' \nabla f(x) \geq 0 \Rightarrow f(y) \geq f(x) \quad \forall x, y \in R^n$$

This is the weakest condition on an objective function minimized over a convex set that still ensures that the Kuhn-Tucker conditions are sufficient (10). On the other hand, clearly φ is not convex; it is an upside-down normal density function. Thus we are testing the technique with a reasonably difficult function; it is globally minimizable, but not without some serious effort.

The technique was applied to φ . Table 1 shows the results of the application with $\underline{x} = (0,0)$, $\bar{x} = (20,20)$ and $\hat{x}^1 = (10,10)$. Finally, we considered x_2 to be integer.

Number of Stages (T)	TABLE 1 - TEST RESULTS		Distance From True Optimum
	Total Experiments (N)	Time(Second)	
1	10	.84	3.507
1	20	1.02	3.532
2	20	1.99	.401
3	20	2.84	.449
3	50	4.39	.568
4	50	5.7	.572

The distance from the true optimum is computed using an L_2 norm:

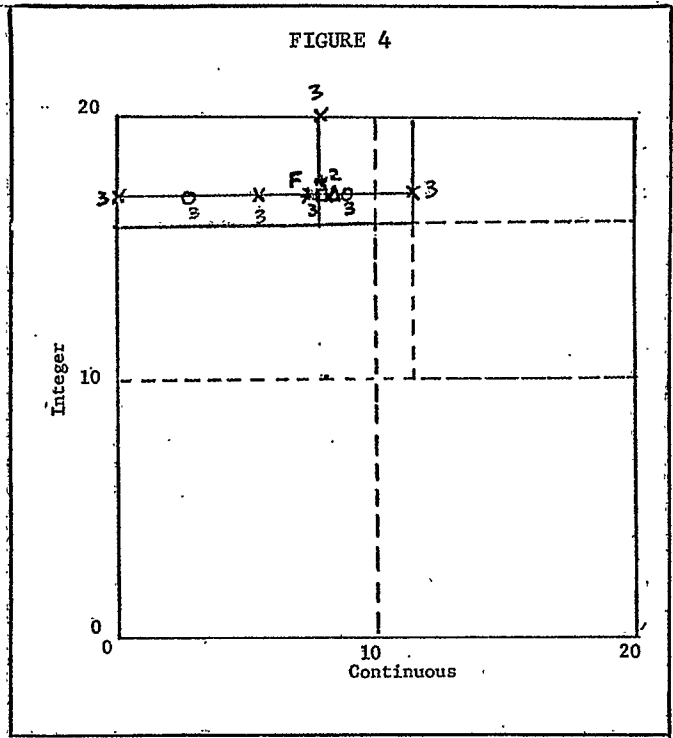
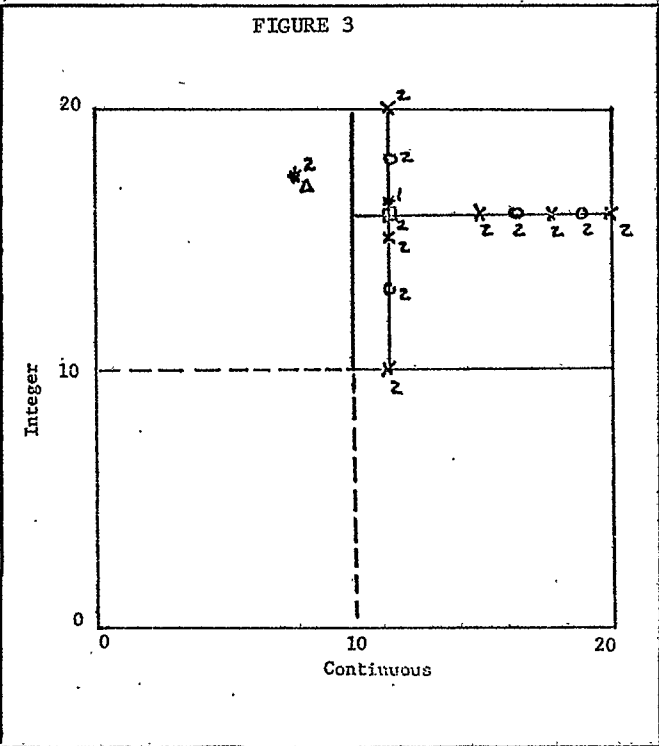
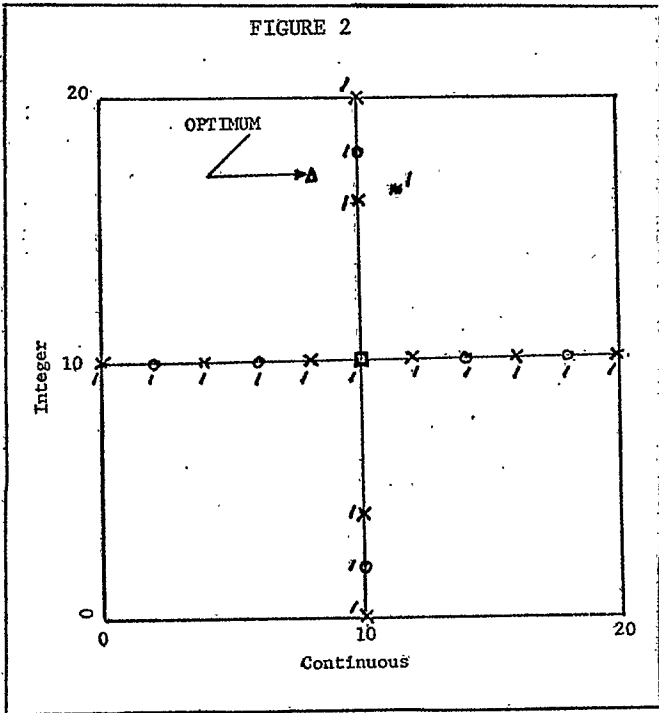
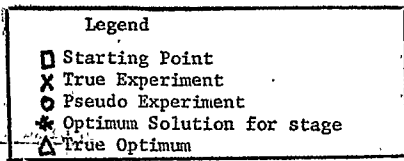
$$\text{distance} = \left\| \left\| x^{*T} - x^* \right\| \right\| = \left((x_1^{*T} - 8)^2 + (x_2^{*T} - 17)^2 \right)^{\frac{1}{2}}$$

where x^{*T} is the optimal solution given by the program at the end of T stages, after dropping the splines (as discussed in section 2) and estimating the grand function only on the actual simulation runs. It should be noted that the optimal solutions were found by relaxing the integrality condition on the integer x_i^1 , i.e. we solved the continuous NLP problem to provide x^{*k} and then rounded-off any variables that were supposed to be integer for the next stage. The distances computed above are on the real solution values so that rounding-off would not affect the comparisons.

Table 1 implies some interesting strategies. First, it appears that (at least initially) increasing the number of stages is more effective than increasing the number of experiments. Eventually, however, this strategy appears to be ineffective. Clearly, two stages seems to be called for, so that the technique can correct for a poor starting point. We allocated the experiments to stages based on the sequence $b_i = V_i$, as discussed in section two above. Thus, most of the experiments were allocated to the first stage (for example if $T = 3$ and $N_1 = 27$, $N_2 = 15$ and $N_3 = 8$). This decision was made so as to allocate experiments roughly in proportion to the decreasing volume of the expected search area. An alternative strategy appears to be to select a sequence that results in more of the experiments allocated to the second or middle stage. This would then trade-off the issue of a poor starting point with the problem of over-concentrating search in the declining volume search areas.

Another run was made with $T = 2, N = 20$ and a changed search area. Here the initial area was $\underline{x} = (5,0)$, $\bar{x} = (15,20)$. In this case the resulting solution's distance from the true optimum was .237, which was obtained in 1.8 seconds. Thus, as could be expected, reducing the initial region can be very helpful.

Figures 2, 3 and 4 show the progress of the technique for $T = 3, N = 20$. Figure 2 shows the first stage allocation of true experiments (x 's), the resulting spline interpolations (o's), the true optimum (Δ), the first-stage starting point (\square), and the first-stage optimal solution (*). The region is the entire square, and the starting point is at (10,10). The program found $x^{*1} = (11.49, 16.43)$; note that we did not solve a nonlinear integer program for the optimal solution. The starting point for the second stage thus was $x^{*2} = (11.49, 16)$,



which is indicated by a box in Figure 3. Notice also that the region for allocating experiments is now defined by $\underline{x} = (10,10)$, $\bar{x} = (20,20)$ as discussed in section two. It turns out, however, that x^{*2} is outside of this region, as shown in Figure 3. Thus, as seen in Figure 4, this generates a new region due to the relaxation of the violated bound as well as the shift of the upper bound. Notice that even at stage two we are almost at the true optimum (Δ). The stage three optimum is not shown in Figure 4, but it is approximately where the starting point is. The final optimum (found after dropping all spline generated points and reestimating the grand function), is labelled 'F' and lies slightly to the left of the stage three starting point.

3.3 Implications

Clearly one does not like to draw sweeping conclusions from a limited set of tests. In general, however, the following conclusions seem intuitively justifiable and warranted:

1. Stages help — i.e., we found that spreading experiments over stages is valuable. Clearly, however, there is a limit in terms of density of experiments per stage.
2. Run times should be short. This is true since the most difficult part of the effort (besides the simulation) is finding x^{*k} . Since this is a nonlinear optimization over very simple constraints (bounds), the expected costs are reasonably low. Obviously, in comparison to a typical simulation run, the run time for the methodology is low. Thus while more stages cost more, their marginal cost appears to be small.

3. As always, starting closer to the optimum with a smaller region helps. This can be seen in the special case tried wherein the starting region was reduced.

IV. APPLICATION OF THE TECHNIQUE

In order to demonstrate that the procedure developed in the previous section can be useful when applied to a realistic model, a case study has been undertaken involving a simulation model of a rail-classification yard. This section briefly describes the model and presents the results of the case study.

Operations in a rail-classification yard can be broken down into four major sets of activities:

- (1) inspection of inbound trains;
- (2) classification of cars in inbound trains into various blocks for departure;
- (3) assembly of outbound trains from these various blocks of cars; and
- (4) inspection and dispatching of outbound trains.

Each car entering the yard must be processed through these activities, and quite obviously there are delays associated with waiting and processing at each stage. Conceptually, one can think of the yard as being a sequence of queues through which a given car must pass.

There have been limited attempts to analyze rail-yard operations using analytic queuing models (see Petersen (13)), but most efforts have involved simulation. A number of these models are reviewed by Folk (8). Many of the yard models are extremely complicated and expensive to run. The model used in this study is a very simplified formulation which has been designed primarily for work in estimating costs of network operations. It is thus less detailed—and hence less realistic—than some other models, but is still a non-trivial simulation.

A number of possible outputs could be of interest in such a model. We have chosen to concentrate on one—mean total time in the yard for all cars—as an example. This output will be influenced by several controllable inputs, as well as by a number of exogenous factors. For purposes of example, we have chosen five controllable inputs to be the independent variables of interest for this problem:

- (1) average train length for inbound trains;
- (2) arrival rate of inbound trains;
- (3) number of outbound trains scheduled per day;
- (4) minimum allowable length of outbound trains; and
- (5) maximum allowable delay for dispatch of outbound train before cancellation.

The product of the first two factors determines the arrival rate of cars to be classified. As this rate increases, we expect more congestion in the classification facility, and hence longer waiting

times; however, as more cars are processed in a given yard, more frequent trains to various destinations can be operated, and hence delays waiting for outbound connections are likely to be shorter. The last three factors determine the policy on outbound train dispatches. At the scheduled departure time of each train, a check is made of the number of cars presently available to depart on that train. If sufficient traffic is present, the train is assembled; if the available traffic is insufficient, however, the train is delayed in expectation that more cars will soon arrive. As the delay time increases, the required number of cars for dispatch decreases, until it reaches some minimum value (factor 4) at a certain time (factor 5). If sufficient traffic is still not available, the train is cancelled. The interested reader is referred to Beckmann, et al. (2) or Folk (7) for more detailed discussion of such dispatching policies.

Note that of the five input factors, two are real variables and three are integers. Thus, this example will test both capabilities in the search algorithm. Table 1 summarizes the variables and bound values used for the experiments.

Variable Name	Definition	Lower Bound	Upper Bound
AVGTL (real)	average inbound train length (cars)	40.	80.
LAM (real)	inbound arrival rate (trains/hour)	.25	.50
NOUIT (integer)	scheduled outbound trains/day	5	12
MAXDEL (integer)	maximum delay on outbound trains (hours)	1	12
MINL (integer)	minimum allowable train length (cars)	15	40

As the ranges on the input variables are quite different, and because the output of the model can be expected to be more sensitive to a unit change in one variable than to a comparable change in another, the inputs were scaled to achieve some degree of comparability in range. This scaling involved division of AVGTL by 4, multiplication of LAM by 40, and division of MINL by 5.

Experimental Results

Based on the results of the earlier sensitivity tests, the number of stages of experimentation was set at three. An experimental budget of 55 simulation runs was assumed, leading to an allocation of

30 runs to the first stage, 16 to the second and 9 to the third.

In the first stage, the allocation algorithm placed 10 experiments along the AVGTTL axis, 9 on the LAM axis, 3 on the NOUTT axis, 1 on the MINL axis, and 7 on the MAXDEL axis. This allocation reflects the prior scaling done on the input variables. The experiment allocation algorithm attempts to distribute the experiments over the entire experimental region, and thus places more experiments on longer axes. After these 30 experiments, 17 pseudo-experiments are generated from the four fitted spline functions (omitting the MINL axis) at points midway between the simulation experiments on each axis.

The first-stage quadratic response surface is then estimated, based on the data from these real and pseudo-experiments. This function is minimized over the feasible region, resulting in an estimated minimum average time in yard of 14.2 hours, at the point:

AVGTTL = 59.
LAM = .25
NOUTT = 9
MINL = 15
MAXDEL = 6

The first experimental region is then updated to:

$40. \leq \text{AVGTTL} \leq 60.$
 $.25 \leq \text{LAM} \leq .35$
 $8 \leq \text{NOUTT} \leq 12$
 $15 \leq \text{MINL} \leq 25$
 $1 \leq \text{MAXDEL} \leq 12$

This defines the region for second-stage experiments, with the first-stage optimal solution as the starting point. The allocation algorithm places 16 experiments in this region, 4 on the AVGTTL axis, 4 on the LAM axis, 1 on the NOUTT axis, none on the MINL axis, and 7 on the MAXDEL axis. Eight more pseudo-experiments from splines are added, and then a second-stage quadratic function is estimated over the entire feasible region, including all 46 experimental observations and the 25 pseudo-observations.

The minimum of the second-stage quadratic is 13.2 hours, at the point:

AVGTTL = 80.
LAM = .25
NOUTT = 12
MINL = 15
MAXDEL = 12

Note that this point has all variables at either their lower or upper bounds. It is an extreme-point solution. Also, note that the second-stage optimum for AVGTTL is outside the experimental region for the second stage. This causes the region to be re-expanded along this dimension, as it indicates a possible error at the first stage. Thus, the experimental region for the third stage is:

$59. \leq \text{AVGTTL} \leq 80.$
 $.25 \leq \text{LAM} \leq .35$
 $9 \leq \text{NOUTT} \leq 12$
 $15 \leq \text{MINL} \leq 25$
 $6 \leq \text{MAXDEL} \leq 12$

In the third stage, the 9 remaining simulation experiments are allocated, with 3 on the AVGTTL axis, 3 on the LAM axis, and 3 on the MINL axis. Four pseudo-experiments are added, and the third-stage quadratic is estimated from the 55 simulation experiments and the 29 pseudo-experiments. The minimum of this quadratic is 16.8 hours, and occurs at:

AVGTTL = 80.
LAM = .25
NOUTT = 12
MINL = 15
MAXDEL = 12

This is the same point as identified in the second stage. Thus, the algorithm would terminate at this point, even if more stages had been planned originally.

As a final output, the pseudo-experiments are eliminated and the response surface is re-estimated, based only on the real experiments. The minimum of this re-estimated function is located at the same point as with the pseudo-experiments included, and also has the value 16.8 hours. While this would not be expected to happen regularly, it does in this case, in part because the optimal solution occurs at the boundary of the feasible region.

Qualitatively, this optimal point corresponds to a situation in which inbound trains are long but relatively infrequent, and outbound trains are scheduled frequently but with maximum flexibility on delay before departure and with the minimum limit on allowable train length. This ensures that few outbound trains will be cancelled—a situation which leads to long delays for waiting cars, as they must then wait for the next scheduled train. In the terms of the problem defined for this example, this solution is a reasonable one. Thus, while the true optimal solution is unknown, the optimization procedure has defined a solution which is at least plausible.

This solution point does not correspond to any of the actual experiments run during the search procedure; it is simply determined from the estimated response surface. In order to check the solution, the simulation model was run with inputs set at the indicated values. The result was a mean time in-yard of 21.3 hours. While this differs substantially from the estimated value given by the response surface, it does compare favorably with actual experimental results obtained for other parameter settings. The response of the simulation model is relatively insensitive in the neighborhood of this point, yielding results between 20.5 and 21.5 hours for several different experiments. These results are, however, quite distinguishable from results in other parts of the feasible region, where the mean time is generally between 30.0 and 35.0 hours. Thus, it appears that the optimization algorithm has successfully identified the region containing the optimal parameter values.

In evaluating the implications of this solution for a real yard, we must keep in mind the rather limited formulation of the problem adopted for purposes of example. The solution indicated is one in which there are a small number of long inbound trains, and a large number of short outbound trains. While this may optimize the performance of a single yard, we must remember that these outbound trains will be the in-

bound trains at other yards in the network. Thus, when we consider a larger network as a system, the parameters for operation of each individual yard are likely to change.

In terms of computational effort, this 3-stage, 5-variable problem required approximately 17 seconds (exclusive of simulation) on a CDC 6600. Most of this time is required for estimation and optimization of the response functions, which must be done four times. The software is still very much in a developmental phase, and as further work is done on it the efficiency can be expected to improve even further.

V. CONCLUSIONS

A technique has been presented for optimization with simulation models which differs in several respects from other techniques in the literature. First, it places emphasis on obtaining the best approximation to the optimal response within a given budget limitation, rather than relying upon algorithms that converge asymptotically to the optimal point. Second, in order to obtain estimates of the response surface with less simulation experimentation, the simulation observations are supplemented by "pseudo-experiments" generated from spline approximations along the relevant axes. These splines are very inexpensive to compute, and allow the experimental budget for simulation to be stretched further. Finally, a response surface is estimated over the entire feasible region of the problem, rather than just locally around the estimated optimum. While such a surface is likely to be a less exact approximation of the simulation output, it is more useful for gaining an understanding of the model's performance over a larger region of interest. Such an understanding can be useful to the analyst, both for optimization and also for replacement of the simulation in higher-order models of systems in which the simulation reflects behavior of only one component.

Results from using the technique on a simple two-variable problem in which the exact optimal solution is known indicate that it is quite capable of producing accurate estimates of optimal points with relatively small expenditure on experiments. These tests also indicate that in general it is advisable to increase the number of stages of experimentation rather than the size of the experiment pool, subject to a need to keep the number of experiments available in each stage above some minimal number. Finally, the tests indicate that careful specification of the feasible region is important. If the region of interest can be reduced, a given number of experiments can be used much more effectively.

Application of the technique to a simulation of a rail classification yard indicates that it can be useful in practical situations. This model involves five input variables and, with a budget of 55 experiments, the technique produces a solution which is a very plausible optimum. Furthermore, the computational requirements of the algorithm appear to be quite modest.

The technique presented in this paper must be considered a heuristic at this point, although effort is underway to establish the mathematical properties of the algorithm more precisely. This work is not yet complete; but because initial experiments with the procedure produced very promising results, it was felt that an initial presentation of the technique as a heuristic would be appropriate. Perhaps this exposition will stimulate other researchers to examine similar approaches.

A number of aspects of the algorithm are important areas for continuing research; work on the theoretical underpinnings of the algorithm is certainly one of these. Such work should include explicit consideration of the effects of stochastic error terms in the simulation output. These error terms are dealt with implicitly in the existing procedure, because we are doing a regression estimate of the response function to the experimental output. However, the presence of error terms often has serious effects on convergence properties of search algorithms, and these need to be addressed explicitly.

Further experience with constraints (other than bounds) on the input variables would also be useful. While addition of such constraints presents no conceptual problems as long as the feasible region remains convex, the additional computational requirements imposed by their presence may be significant.

At a more detailed level, an issue of significant interest is the allocation of experiments across stages. The results presented here are based on a reasonable but arbitrary sequence, using terms of the form $1/i$, where i is the stage number. The allocation of experiments involves a trade-off between the larger region of interest in early stages (requiring more experiments to examine) and the desire to place most experiments in the sub-region of greatest promise, which is identified only in later stages. Further work on this problem is likely to be very productive.

In summary, the work to date has resulted in an idea which appears to hold considerable promise for optimization with simulation models. Additional work now underway should help to determine the degree to which that promise can be realized.

REFERENCES

1. Biles, W.E., "A Gradient-Regression Search Procedure for Simulation Experimentation," Proceedings of the 1974 Winter Simulation Conference, Washington D.C., Jan. 1974, pp. 491-497.
2. Beckmann, M., C.B. McGuire and C.B. Winsten, Studies in the Economics of Transportation, Yale University Press, New Haven, 1956.
3. Box, M.J., "A New Method of Constrained Optimization and a Comparison with Other Methods," Computer Journal, 8, 1965, pp. 42-52.
4. Chandrasekeran, R., private communication.
5. Eldredge, D.L., "Experimental Optimization of Statistical Simulations," Proceedings of the 1974 Winter Simulation Conference, Washington, D.C., Jan. 1974, pp. 503-510.

6. Farrell, W., "Literature Review and Bibliography of Simulation Optimization," Proceedings of the 1977 Winter Simulation Conference, Gaithersburg, Md., Dec. 1977, pp. 117-124.
7. Folk, J.F., Models for Investigating Rail Trip Time Reliability, Studies in Railroad Operations and Economics, Vol. 5, Massachusetts Institute of Technology, Department of Civil Engineering, 1972.
8. Folk, J.F., A Brief Review of Various Network Models, Studies in Railroad Operations and Economics Vol.7, Massachusetts Institute of Technology, Department of Civil Engineering, 1972.
9. Greville, T.N.E. (ed.), Theory and Applications of Spline Functions, Academic Press, New York, 1969.
10. Mangasarian, O.L., Nonlinear Programming, McGraw-Hill, New York, 1969.
11. Mihram, G.A., "An Efficient Procedure for Locating the Optimal Simular Response," Proceedings of the Fourth Conference on the Applications of Simulation, New York, Dec. 1970, pp. 154-161.
12. Myers, R.H., Response Surface Methodology, Allyn-Bacon, Boston, 1971.
13. Petersen, E.R., "Railyard Modeling," Transportation Science, 11, 1977, pp. 37-59.
14. Rosen, J.B., "The Gradient Projection Method for Nonlinear Programming, Part I - Linear Constraints," Journal of the Society of Industrial and Applied Mathematics, 8, 1960, pp. 181-217.
15. Rosen, J.B., "The Gradient Projection Method for Nonlinear Programming, Part II - Nonlinear Constraints," Journal of the Society of Industrial and Applied Mathematics, 9, 1961, pp. 514-532.
16. Spendley, W., G.R. Hext, and R.F. Himsworth, "Sequential Application of Simplex Designs in Optimization and Evolutionary Operation," Technometrics, 4, 1962, pp. 441-461.