Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

AN ENHANCED RSM ALGORITHM USING GRADIENT-DEFLECTION AND SECOND-ORDER SEARCH STRATEGIES

Shirish S. Joshi

Center for Transporation Analysis
Oak Ridge National Laboratory
P. O. Box 2008
Oak Ridge, Tennessee 37831-6206

Hanif D. Sherali

Department of Industrial and Systems Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

Jeffrey D. Tew

ISQ Department
P.O. Box 6696
Consolidated Freightways, Inc.
Portland, Oregon 97208-6696

ABSTRACT

This paper presents a new methodology for improving the search techniques currently being used in standard Response Surface Methodology (RSM) algorithms. RSM is a collection of mathematical and statistical techniques used for experimental optimization. Our improved RSM algorithm incorporates certain gradient deflection methods, augmented with appropriate restarting criteria, as opposed to using the path of steepest descent as the only search direction. In order to empirically investigate our new RSM algorithm in comparison with the standard RSM techniques, a set of standard test functions is used. We consider two cases for each test function; with a random perturbation added to the function and without a random perturbation added to the function. Computational results exhibit the improvements achieved under the proposed algorithm.

1 INTRODUCTION

The focus of this paper is on improving the standard algorithm of response surface methodology (RSM) as it is applied to the optimization of simulation models. First proposed by Box and Wilson (1951) in the context of optimization problems concerned with chemical process engineering, RSM is a collection of mathematical and statistical techniques for experimental optmization. The search procedures currently employed by RSM use the method of steepest descent *only*. This paper develops a new RSM algorithm which incorporates certain gradient deflection methods and is further augmented with suitable restarting criteria. We use a set of standard test functions taken from the literature to empirically investigate the merits of this new RSM algorithm relative to the standard RSM algorithm. Computational results exhibit the improvements achieved under the proposed algorithm.

This paper is organized as follows. The remainder of this section presents an overview of RSM and defines the standard RSM algorithm. A discussion of various competitive gradient deflection techniques is also given. Section 2 gives a definition of our proposed RSM algorithm. A summary of extensive empirical results of a comparision study are given in Section 3. Section 4 gives conclusions and recommendations for future research.

1.1 Definition of Response Surface Methodolgy

Throughout this paper we focus only on an *unconstrained* minimization problem and that the functional form of the objective function is unknown. Typically, to start the RSM procedure, an experiment is designed in a small sub-region of the factor space, and a low-degree polynomial (usually first-order) is used to represent the data obtained from the responses. This polynomial helps the experimenter decide the next region of the factor space that should be explored by identifying a *search direction* along the path of steepest descent (in the case of minimization). If the goal is to minimize the response, this method tries to "climb down" the response surface toward the overall, response minimum rather than exploring a large region of the factor space. Its success depends on the assumption that the overall, response minimum can be reached via such a path of descent (see Davies 1956, p. 503). (Throughout the remainder of this paper, we assume that problem at hand is to *minimize* the expected mean response over the factor space of interest.)

If the experimenter has a prior notion of the general vicinity of the location of the minimum, then Myers (1976) gives the following five-step procedure (see Myers 1976, p. 88):

STEP 1: Fit a first-order regression model to the mean response over some restricted (usually taken to be small) region of the factor space.

STEP 2: Based on the results of Step 1, estimate a path of steepest descent.

STEP 3: Perform a series of experiments along the estimated steepest descent path until no additional decrease in the mean response is evident.

STEP 4: Based on the results of Steps 1, 2, and 3, estimate the overall, minimum mean response.

STEP 5: Repeat Steps 1, 2, 3, and 4 over a *new* region centered at the current estimate of the overall, minimum mean response. If curvature is evident and the experimenter is satisfied that little or no significant, additional improvement in the estimate of the overall, minimum mean response can be obtained from conducting further searches, a more elaborate experiment and analysis study is conducted using a second-order design. From this design, a final estimate of the minimum mean response is obtained.

In Step 1, typically, replications of a factorial or fractional factorial experiment are performed. The unknown parameters of the fitted model are then computed. A judicious selection of an experimental design having desirable properties such as minimum variance of model parameter estimates is needed in order to obtain better estimates of the unknown parameters. The first-order model linear model is represented as

$$y_{ij} = \beta_0 + \sum_{l=1}^{k} \beta_l x_l + \epsilon_{ij}, \qquad (1)$$

where y_{ij} is the response of the jth replication at the ith design point, β_l is the lth unknown model parameter, x_l is the lth experimental factor, and ϵ_{ij} is the error term at the ith design point and jth replication, (for $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, r$; and $l = 1, 2, \ldots, k$).

If, for example, a full 2^k factorial experiment is used in Step 1, then estimates of all the β_l can be obtained. We denote these estimates by b_l (for $l = 1, 2, \ldots, k$). These estimates are then used in Step 2 to locate the path of steepest descent, which is given by the vector $\mathbf{d} = (-b_1, -b_2, \ldots, -b_k)$. Responses are oberved along this path using some prescribed search strategy until there is no observed, significant improvement in the mean response. In the standard RSM approach, a simple sequence of fixed steps along the direction $\mathbf{d}$ are taken. If curvature is evident, then a second-order model is fitted. The second-order model is represented as:

$$y_{ij} = \beta_0 + \sum_{l=1}^{k} \beta_l x_l + \sum_{h>l} \beta_{hl} x_h x_l + \sum_{l=1}^{k} \beta_{ll} x_l^2 + \epsilon_{ij}, \qquad (2)$$

where β_{hl} is the coefficient of interaction between factors h and l, β_{ll} is the second-order coefficient for factor l, and all other terms are as defined in (1). The model in (2) is used to evaluate a stationary point of the response surface. A canonical analysis is performed that typically involves locating the stationary point as well as determining the nature of this point. From this analysis, an estimate of the optimum is obtained; perhaps following additional investigation in the case of a detected ridge system.

The principal shortcomings of the standard RSM algorithm described in this section are two-fold. First, the inital gradient search can be prone to "zigzaging" and can be slow to converge (see Bazaraa, Sherali, and Shetty, 1993). Moreover, there is no information from previous iterations that is successively employed to provide improved search directions (the process is essentially memoryless). Second, in the event that curvature is detected, there is no attempt to conduct a continuous, iterative, second-order search using the information available from the canonical analysis. In this paper we present a revised RSM algorithm which attempts to rectify these shortcomings using a technique called gradient deflection.

1.2 Gradient Deflection Methods

This section discusses the four gradient deflection methods we used in revising the standard RSM algorithm presented in the previous section. Originally developed by Hestenes and Stiefel (1952), conjugate gradient (gradient deflection) methods were first applied to unconstrained minimization problems by Fletcher and Reeves (1964). In this approach, a sequence of design points, $\mathbf{x}_j$, and a sequence of directions $\mathbf{d}_j$, are generated iteratively as

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \lambda_j \mathbf{d}_j, \tag{3}$$

where

$$\mathbf{d}_j = -\mathbf{g}_j + \kappa_j \mathbf{d}_{j-1}, \tag{4}$$

and $\mathbf{d}_0 = \mathbf{0}$, κ_j is a scalar multiplier that scales the direction vector of the previous iteration (the calculation of κ_j varys under different gradient deflection methods), $\mathbf{g}_j$ is the gradient of the objective function at the operating point $\mathbf{x}_j$ (assumed to be nonzero, or else the method is terminated), and λ_j is the step length adopted along the descent direction $\mathbf{d}_j$. As seen from (4), a deflected direction $\mathbf{d}_j$ at the jth iteration is comprised of a linear combination of the negative gradient at the jth iteration and the direction vector of the previous iteration. This implies a possible advantage in this method over the method of steepest descent, since the deflection strategy attempts to capture the second-order curvature effects over successive iterations. For a quadratic function in k variables, this approach can be made to converge to an optimum within k iterations (see Bazarra et al., 1993).

Various gradient deflection methods use different techniques for computing the deflection parameters κ_j at the jth iteration. In this paper, we consider four such techniques that have been suggested in the literature. A brief description of each of these techniques follows.

A gradient deflection method proposed by Sherali and Ulular (1989) that was found to be promising even for nondifferentiable functions (where $\mathbf{g}_j$ represents a subgradient of the function) uses the *average direction strategy* and is denoted here as GD1. The deflected direction at the jth iteration for this method is given by

$$\mathbf{d}_j = -\mathbf{g}_j + \frac{||\mathbf{g}_j||}{||\mathbf{d}_{j-1}||}\mathbf{d}_{j-1}, \tag{5}$$

where $\mathbf{d}_{j-1}$ is the search direction adopted at the $(j-1)$st iteration, with $\mathbf{d}_1 = -\mathbf{g}_1$.

In the context of proposing a conjugate gradient algorithm that adopts quasi-Newton types of updates and permits inexact line searches, Sherali and Ulular (1989) develop yet another scheme for computing κ_j. Their method, denoted in this paper as GD2, generates the deflected direction at the jth iteration as

$$\mathbf{d}_j = -\mathbf{g}_j + \frac{\mathbf{q}'_j \mathbf{g}_j - (\frac{1}{s_j})\mathbf{p}'_j \mathbf{g}_j}{\mathbf{q}'_j \mathbf{d}_{j-1}}\mathbf{d}_{j-1}, \tag{6}$$

where

$$\mathbf{p}_j = \mathbf{x}_j - \mathbf{x}_{j-1}, \tag{7}$$

$$\mathbf{q}_j = \mathbf{g}_j - \mathbf{g}_{j-1}, \tag{8}$$

and s_j is a scale parameter that, if suitably chosen, permits $s_j \mathbf{d}_j$ to be the Newton direction at the jth iteration, given that this direction is spanned by $-\mathbf{g}_j$ and $\mathbf{d}_{j-1}$. From a computational viewpoint, the parameter s_j is prescribed as

$$s_j = \lambda_{j-1}. \tag{9}$$

Motivated by the computational success of BFGS quasi-Newton updates, Shanno proposed a related conjugate gradient strategy for which the direction at the jth ($j = 2, 3, \ldots$) iteration is given by

$$\mathbf{d}_j = -\left[\mathbf{I} - \frac{\mathbf{p}_j \mathbf{q}'_j + \mathbf{q}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j}\right]\mathbf{g}_j, \tag{10}$$

where $\mathbf{p}_j$ and $\mathbf{q}_j$ are as defined in (7) and (8), respectively. We denote this gradient deflection method as GD3.

Finally, a fourth gradient deflection method we considered, denoted by GD4, is a modification of GD2 suggested by Sherali and Ulular (1990) that converts this strategy into a symmetric, memoryless BFGS type of an update. The prescribed search direction under this modification is given by

$$\begin{aligned} \mathbf{d}_j \; = \; & -\left[\mathbf{I} - \frac{\mathbf{p}_j \mathbf{q}'_j - \mathbf{q}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j} + \right. \\ & \left. + \left[\frac{1}{s_j} + \frac{\mathbf{q}'_j \mathbf{q}_j}{\mathbf{q}'_j \mathbf{p}_j}\right] \frac{\mathbf{p}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j}\right]\mathbf{g}_j, \end{aligned} \tag{11}$$

where $\mathbf{p}_j$, $\mathbf{q}_j$, and s_j are as defined in (7), (8), and (9), respectively.

The four gradient deflection methods described above were used in our emprical study whose partial results are summarized in Section 3.

1.3 Restarting Techniques

In addition to the gradient deflection techniques presented in the previous section, restarting techniques, in conjuction with gradient deflection methods, can significantly enhance their performance. Beale (1972) and Powell (1977) have offered various criteria for improving the performance of gradient deflection methods. In the present context of RSM, restarting at any iteration would imply following the path of steepest descent at that iteration; although, as suggested by Powell (1977), in the context of conjugate gradient methods, one could restart using the current deflected direction in order to preserve the accumulated second-order information. However, this would require an additional term in the deflection scheme.

We considered two restarting techniques, RSA and RSB, which were used in conjuction with the four gradient deflection methods discussed in the previous section.

First, we consider RSA. For a k-factor problem, the optimization procedure is restarted at the jth iteration, by setting $\mathbf{d}_j = -\mathbf{g}_j$, if *any* one of the following three conditions is satisfied:

CONDITION 1: $j = k$,

CONDITION 2: $||\mathbf{g}'_j \mathbf{g}_{j+1}|| \geq (0.2)||\mathbf{g}_j||^2$,

CONDITION 3:

$$(-1.2)||\mathbf{g}_i||^2 \leq \mathbf{d}'_{proj,j}\mathbf{g}_{proj,j} \leq (-0.8)(\mathbf{g}'_j \mathbf{g}_{proj,j})$$

is false,

where $\mathbf{d}_{proj,j}$ and $\mathbf{g}_{proj,j}$, respectively, project $\mathbf{d}_j$ and $-\mathbf{g}_j$ onto the box (boundary) constraints by zeroing out the components corresponding to the active bounds that would be immediately violated by moving along the corresponding direction $\mathbf{d}_j$ or $-\mathbf{g}_j$. The rationale for these conditions is provided in Section 4 of Powell (1977) and in Bazaraa et al. (1993).

Next, we consider RSB. For a k-factor problem, the optimization procedure is restarted at the jth iteration if *any* one of the following two conditions is satisfied:

CONDITION 1: $j = k$,

CONDITION 2: $\mathbf{d}'_{proj,j} > (-0.8)(\mathbf{g}'_j \mathbf{g}_{proj,j})$.

Notice that the conditons given for RSB are a subset of the conditions given for RSA, and that in particular, since restarting is triggered whenever $\mathbf{d}'_{proj,j}\mathbf{g}_j > (-0.8)(\mathbf{g}'_j \mathbf{g}_{proj,j})$, the adopted direction $\mathbf{d}_{proj,j}$ is *always* a descent direction having a sufficiently negative directional derivative. In the next section we formally present an enhanced RSM algorithm that incorporates the deflection and restart schemes discussed in this section and also adopts second-order search directions whenever significant curvature is detected.

2 AN ENHANCED RSM ALGORITHM

This section presents an enhanced version of the standard RSM algorithm outlined in Section 1. It incorporates the gradient deflection methods which were presented in Section 1.2, along with second-order search directions. In the following, we assume that the problem is a minimization one in k factors. All factors are assumed to have upper and lower bounds that cannot be violated at a prescribed solution; although, in any experimental design we can set the levels of a factor above or below that factor's bounds (i.e., when the center of a design lies on the boundary of any factor). However, along a search direction if some factors reach their lower or upper bound, then their levels are set to that respective boundary value and the search is continued. The steps of the enhanced RSM algorithm are given below:

STEP 1: Select a starting point as the current incumbent solution.

STEP 2: Construct a 2^k full factorial experiment, using the current incumbent solution as the center of the design. If the first-order model is a good fit, then proceed to Step 3. Otherwise, go to Step 4.

STEP 3: For the first pass of the algorithm ($j = 1$), adopt the path of steepest descent as the search direction $\mathbf{d}_1$. For subsequent iterations, adopt the appropriate deflected gradient direction, $\mathbf{d}_j$, as prescribed in Section 1.2, using any one of the deflection techniques GD1, GD2, GD3, or GD4 unless a restart is triggered by the selected criterion RSA or RSB in which case follow the path of steepest descent (reset $\mathbf{d}_j = -\mathbf{g}_j$). If any variable is at its upper or lower bound, and if the search direction being followed immediately violates this bound, then fix the corresponding component of the search direction, $\mathbf{d}_j$, to zero. If the appropriate restarting criterion is triggered, then restart with the negative gradient direction. (Note that this restarting check is performed only when using deflected diretions.) If the direction adopted is a projection of the negative gradient direction, and this direction is zero, then proceed to Step 9. Otherwise, the present direction is a descent direction. Call this the revised search direction $\mathbf{d}_j$. Determine a maximum step-length, λ_{max}, that can be taken along this direction without violating any bounds and conduct a line search on this interval to determine a suitable step-length $\lambda \in (0, \lambda_{max}]$. If the optimal step-length $\lambda^* \in (0, \lambda_{max}]$, return to Step 2 with the resulting solution as the current incumbent solution. Otherwise, if $\lambda^* = \lambda_{max}$, then further project the current direction by zeroing out the components corresponding to the factors that have just hit their bounds and continue searching along such projected directions until no further improvement can be obtained.

Return to Step 2 with the current incumbent solution. In the rare instance that $\lambda^* = 0$, due to inaccuracies in experimental, functional evaluations, terminate the algorithm by proceeding to Step 9.

STEP 4: Construct a second-order design and check if the second-order model represents the data adequately. If the second-order model is a good fit, then go to Step 6. Otherwise, proceed to Step 5.

STEP 5: Expand (or contract) the design size appropriately and return to Step 2.

STEP 6: Suppose the center of the current design is located at $\bar{\mathbf{x}}$, let $\mathbf{g}_j = \nabla f(\bar{\mathbf{x}})$ as determined by the second-order model. Perform a canonical analysis. Let $\mathbf{H}$ denote the Hessian of the predicted quadratic response function with k eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$. Also, let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ be the corresponding k *normalized* eigenvectors, $\mathbf{Q} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$, $\mathbf{\Lambda} = \text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_k]$, $I_+ = \{i : \lambda_i > 0 \text{ and } \mathbf{v}'_i \mathbf{g}_j \neq 0\}$, and $I_- = \{i : \lambda_i < 0\}$.

If $\mathbf{g}_i \neq \mathbf{0}$, then proceed to Step 7. Otherwise, go to Step 8.

Step 7: If $I_+ = \phi$, then return to Step 3 with $\mathbf{d}_j = -\mathbf{g}_j$. Otherwise, let

$$\mathbf{d}_j = -\sum_{i \in I_+} \frac{\mathbf{v}_i \mathbf{v}'_i \mathbf{g}_j}{\lambda_i}, \tag{12}$$

so that

$$\mathbf{d}'_j \mathbf{g}_j = -\sum_{i \in I_+} \frac{(\mathbf{v}'_i \mathbf{g}_j)^2}{\lambda_i} < 0. \tag{13}$$

If $\mathbf{d}'_{proj,j} \mathbf{g}_j < 0$, then perform Step 3 using $\mathbf{d}_j$ as the prescribed search direction (no restarting criterion is checked in this case). Otherwise, perform Step 3 after resetting $\mathbf{d}_j = -\mathbf{g}_j$ and return to Step 2 with the current incumbent solution. If the optimal step-length λ^* equals zero, then terminate the algorithm by proceeding to Step 9.

Step 8: If $I_- = \phi$, then terminate the algorithm by proceeding to Step 9. Otherwise, let $\lambda_t = \min\{\lambda_i : i \in I_-\}$. Pick $\mathbf{d}_j = \mathbf{v}_t$. In this case, we note that

$$\begin{aligned} f(\bar{\mathbf{x}} + \lambda \mathbf{d}_j) &= f(\bar{\mathbf{x}}) + \lambda \mathbf{g}'_j \mathbf{d}_j + \\ &\quad + \frac{\lambda^2}{2} \mathbf{d}'_j \mathbf{H} \mathbf{d}_j + \\ &\quad + \lambda^2 ||\mathbf{d}_j||^2 O(\lambda - 0), \end{aligned} \tag{14}$$

where $O(\lambda \to 0)$ is a function that approaches zero as $\lambda \to 0$. Since $\mathbf{g}_j = 0$, this yields

$$\begin{aligned} \lim_{\lambda \to 0} \frac{f(\bar{\mathbf{x}} + \lambda \mathbf{d}_j) - f(\bar{\mathbf{x}})}{\frac{\lambda^2}{2}} &= \mathbf{d}'_j \mathbf{H} \mathbf{d}_j \\ &= \mathbf{v}'_t \mathbf{H} \mathbf{v}_t \\ &= \mathbf{v}'_t \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}' \mathbf{v}_t \\ &= \mathbf{e}'_j \mathbf{\Lambda} \mathbf{e}_t \\ &= \lambda_t < 0, \end{aligned} \tag{15}$$

where $\mathbf{e}_t$ is a vector having a 1 in position t and 0's everywhere else. Thus, $\mathbf{d}_j$ is a descent direction. Perform a search as in Step 3 using $\mathbf{d}_j$ as the prescribed search direction (no restarting criterion is checked in this case). If an improvement results, then return to Step 2. Otherwise, if $\lambda^* = 0$ (which would be the case if $\mathbf{d}'_{proj,j} \mathbf{g}_j \geq 0$) repeat this step by replacing I_- with $I_- - \{t\}$.

STEP 9: Terminate the algorithm and report the incumbent solution.

The next section contains a summary of a computational assessment of the enhanced RSM algorithm and draws some conclusions about the relative performance of the four gradient deflection methods discussed in this paper. Recommendations for adoption of a suitable restart policy under each of the gradient deflection methods are also given.

3 EMPIRICAL RESULTS

This section provides a brief summary of experimental results obtained as a result of empirical evaluations of the enhanced RSM algorithm to the standard one presented in Section 1. First, we performed a comparative study of the four gradient deflection methods with no restarting criterion in the context of the RSM algorithm using ten deterministic "test functions" given by Sherali and Ulular (1990). Comprehensive tables of results obtained as a result of the experimental trials are given in Section 3 of Joshi, Sherali, and Tew (1993).

Table I contains computational results showing the best (smallest response value) solutions obtained using the four gradient deflection methods and the method of steepest descent along with the number of simulation runs needed to reach these solutions. Overall, GD1 performed as well as or significantly better than POSD on 9 of the 10 test functions and, sometimes with significantly fewer simulation runs. Moreover, GD1 performed the best of all methods considered in 6 of the 10 cases. In particular, interesting results were obtained for test functions 3, 6, and 8. For functions 3 and 6, GD1, GD2, and GD3 exhibited superior performance to the POSD; again, often with significantly fewer simulation runs. For function 8, all gradient deflection methods performed worse than the method of steepest descent. For the other test functions, in general, POSD did not perform significantly worse or better than the gradient deflection methods. The focus for the remainder of the empirical study was therefore restricted to test functions 3, 6, and 8.

In order to improve the performance of the enhanced RSM algorithm, the starting criteria discussed in Section 2.2, RSA and RSB, were employed. The results for these two criteria were tabulated in Section 3 of Joshi, Sherali, and Tew (1993) and indicated that GD2 performs better with RSA and GD1, GD3, and GD4 work well with RSB. This allocation of the restarting criteria to the gradient deflection methods was preserved throughout the remainder of the study.

A point of interest for function 8 is the number of second-order models employed under the gradient deflection method. Recall from Section 1 that the standard RSM algorithm stops after the *first* second-order model adequately fits the data and reports the most favorable design point obtained using that design as the optimum. The added feature of the modified algorithm that allows the search to be continued using more designs, if needed, *after* the data has been adequately represented by a second-order model at any stage of the algorithm, improves the mean response of the best design point reported. Table 5 of Joshi, Sherali, and Tew (1993) illustrates such an improvement achieved on function 8 under different starting conditions. In particular, note that while POSD achieves a minimum response value of 3.42 if only one second-order model is employed, it attains an improved objective function value of 0.06 if a succession of second-order model runs (5 in this case) are employed. Similarly, GD3 uses 6 second-order models under the restarting criterion RSB to reach a minimum value of 0.02, with the true optimal value being 0. If the same algorithm is used in conjunction with GD3, but only *one* second-order model is employed, then the minimum value obtained was 4.71. This illustrates the improvement possible under the enhanced second-order search steps.

The three test functions were then modified to include a measure of random behavior and all five algorithms (POSD, GD1, GD2, GD3, and GD4) were performed on them. For these runs, at each design point, the response function was augmented by a random error term that was normally distributed with mean 0 and variance equal to 10% of the *true* reponse value at the center of the corresponding design. Two independent replications were performed at each design point and the mean of these two responses was recorded. Independent replications were also performed *across* design points. While performing line searches, the variance on the error term was the same as that used for the first-order design. For all test functions, the uncoded values of the factors for the first-order model were set at $+0.1$ and -0.1 as the high and low levels, respectively. The axial points for the second-order model were selected so as to make the design rotatable (see Section 7.3 in Myers 1976 for a discussion on rotatable designs). Computational results for test functions 3, 6, and 8 with random error terms added are given in Tables 6, 7, and 8 of Joshi, Sherali, and Tew (1993).

With the pressence of a random error term for function 3, GD1 performed considerably better than all of the other methods. Also, GD2, GD3, and GD4 failed to perform as well as POSD in this situation. For function 6, GD1 again performed superior to all other methods. GD4 performed competitively with POSD. For function 8, all methods performed well in terms of the minimum value returned. However, GD1 required significantly more simulation runs than the other methods. The results for function 8 indicated that (contrary to the deterministic case) POSD nolonger performed better than the gradient deflection methods and on average did not perform as well as GD2, GD3, and GD4 while performing only marginally better than GD1. This indicates, as expected, that the relative effectiveness of the methods is significantly affected by the level of random variation present in the response. Overall, GD1 continued to perform better than the other methods and is therefore recommended for future use in RSM optimization problems.

These results, although preliminary, illustrate the relative performance of the enhanced RSM algorithm to the standard RSM algorithm and indicate that significant improvements can be obtained when using gradient deflection methods. The numerical results also indicate that GD1 offers, perhaps, the most potential improvement over POSD.

4 SUMMARY AND CONCLUSIONS

The standard RSM approach uses a pure gradient search method that can be prone to "zig zaging" (i.e., slow to converge). There is no information from pervious iterations that is successively employed to provide improved search directions; the process is essentially memoryless. Moreover, after curvature is detected there is no attempt to conduct a contin-

Table I
Minimum Response with no Restarts

Function #	POSD	GD1	GD2	GD3	GD4
1	0.01 (23)	0.01 (23)	0.01 (23)	0.01 (23)	0.01 (23)
2	0.12 (25)	0.12 (25)	0.12 (25)	0.12 (25)	0.12 (25)
3	16.93 (162)	2.55 (83)	128.76 (52)	2.76 (183)	9.66 (161)
4	1.31 (90)	0.88 (139)	73.05 (49)	1.24 (91)	0.95 (161)
5	0.01 (61)	0.00 (56)	0.00 (55)	0.77 (26)	0.75 (26)
6	6.47 (102)	3.73 (174)	130.04 (63)	5.47 (87)	3.44 (87)
7	0.00 (101)	0.00 (123)	0.00 (102)	0.00 (83)	0.00 (83)
8	0.06 (156)	18.38 (86)	2.25 (131)	2.64 (90)	2.63 (119)
9	2.94 (40)	2.50 (68)	2.41 (68)	2.94 (40)	2.94 (40)
10	3.00 (19)	3.00 (19)	3.00 (19)	3.00 (19)	3.00 (19)

uous iterative second-order search using the information available from the canonical analysis. This paper presents a new RSM algorithm which rectifies these shortcomings by incorporating gradient deflection techniques in conjunction with certain restarting criteria as well as adopting second-order search directions whenever significant curvature is evident. An empirical study comparing our revised RSM algorithm to the standard RSM algorithm using a set of test functions taken from the literature suggests using the average direction strategy proposed by Sherali and Ulular (1989) in conjunction with the criterion of RSB.

REFERENCES

Bazaraa, M. S., H. D. Sherali, and C. M. Shetty. 1993. *Nonlinear programming: theory and algorithms*, 2d. ed. New York: John Wiley & Sons.

Beale, E. M. 1972. A derivation of conjugate gradients, *Numerical Methods for Nonlinear Optimization*, F. A. Lootsma, ed. London: Academic Press, 39–43.

Box, G. E. P. and K. B. Wilson. 1951. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society*, B13:1–38.

Davies, O. L. 1956. *Design and analysis of industrial experiments*, 2d. ed. New York: Hafner Publishing Co., Inc.

Fletcher, R. and C. M. Reeves. 1964. Function minimization by conjugate gradients. *Computer Journal*, 7:149–154.

Hestenes, M. R. and E. Stiefel. 1952. Methods of conjugate gradients for solving linear systems. *Journal of Reasearch of the National Bureau of Standards*, 48:409–436.

Joshi, S., H. D. Sherali, and J. D. Tew. 1993. An Enhanced Response Surface Methodology Algorithm Using Gradient-Deflection and Second-Order Search Strategies. *Technical Report*, Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University.

Myers, R. H. 1976. *Response surface methodology.* Ann Arbor: Edwards Brothers, Inc.

Powell, M. J. D. 1977. Restart procedures for the conjugate gradient method. *Mathematical Programming*, 12:241–254.

Sherali, H. D. and O. Ulular. 1989. A primal-dual conjugate subgradient algorithm for specially structured linear and convex programming problems. *Applied Mathematics and Optimization*, 20:193–221.

Sherali, H. D. and O. Ulular. 1990. Conjugate gradient methods using quasi-newton uptdates with inexact line searches. *Journal of Mathematical Analysis and Applications*, 150:1359–377.

AUTHOR BIOGRAPHIES

SHIRISH JOSHI is a post-doctoral research associate at the Oak Ridge National Laboratory in the Intelligent Vehicle Highway Systems (IVHS) research group. He received his B.S. in Electrical Engineering from Bombay University, Bombay, India, in 1986, a M.S. in Mathematical Sciences from Virginia Commonwealth University, Virginia, M.S. and Ph.D in Industrial and Systems Engineering from Virginia Polytechnic Institute and State University, Blacksburg, Virginia in 1991 and 1993, respectively. His current research interests include output analysis of traffic simulations, variance reduction techniques, simulation-optimization technques, and design of simulation experiments.

HANIF D. SHERALI is the Charles O. Gordon Professor of Industrial Engineering in the Department of Industrial and Systems Engineering at Virginia Polytechnic Institute and State University. He has published numerous articles in many archival journals such as *Applied Mathematics and Optimization*, *EJOR*, *IIE*, *Journal of Mathematical Analysis and Applications*, *Management Science*, and *Operations Research*, among other. He is also a co-author of text *Nonlinear Programming: Theory and Algorithms* 2nd. edition. His research interests are in applied mathematics and mathematical programming. He is also a member of numerous professional societies.

JEFFREY D. TEW is a Senior Systems Engineer at Consolidated Freightways, Inc. and an Adjunct Associate Professor of Computer Simulation at the Oregon Graduate Institute. Prior to these positions he was on the facluty in the Industrial and Systems Engineering Department at Virginia Polytechnic Institute and State University. He has published numberous articles in many archival journals such as *EJOR*, *JSCS*, *Management Science*, *Operations Research*, and *SCS Transactions* among others. He is the *1994 WSC Proceedings* Editor as well as the Vice President/President Elect of the TIMS College on Simulation. He received a B.S. in mathematics from Purdue University in 1979, a M.S. in statistics from Purdue University in 1981, and a Ph.D. in industrial engineering from Purdue University in 1986. His current research interests include variance reduction techniques, simulation optimization, and the design of simulation experiments. He is a member of Alpha Pi Mu, ACM, ASA, IIE, IMS, ORSA, SCS, Sigma Xi, and TIMS.