

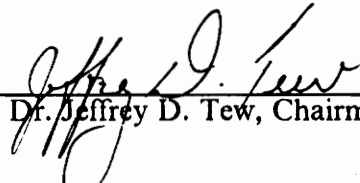
**Simulation-Optimization Studies : Under Efficient Simulation  
Strategies, and a Novel Response Surface Methodology Algorithm**

by

Shirish Joshi

Dissertation submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of  
Doctor of Philosophy  
in  
Industrial and Systems Engineering

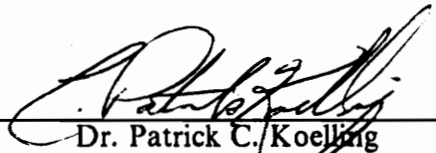
APPROVED:



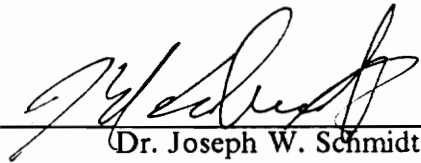
Dr. Jeffrey D. Tew, Chairman



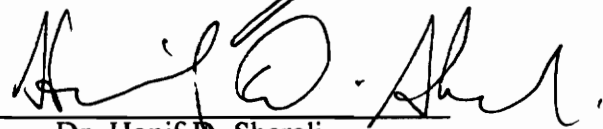
Dr. Ernest C. Houck



Dr. Patrick C. Koelling



Dr. Joseph W. Schmidt



Dr. Hanif D. Serali

9 August, 1993.

Blacksburg, Virginia

C.2

LD  
5655  
V856  
1993  
J678  
C.2

**Simulation-Optimization Studies : Under Efficient Simulation  
Strategies, and a Novel Response Surface Methodology Algorithm**

by

Shirish Joshi

Dr. Jeffrey D. Tew, Chairman

Industrial and Systems Engineering

(ABSTRACT)

While attempting to solve optimization problems, the lack of an explicit mathematical expression of the problem may preclude the application of the standard methods of optimization which prove valuable in an analytical framework. In such situations, computer simulations are used to obtain the mean response values for the required settings of the independent variables. Procedures for optimizing on the mean response values, which are in turn obtained through computer simulation experiments, are called simulation-optimization techniques.

The focus of this work is on the simulation-optimization technique of response surface methodology (RSM). RSM is a collection of mathematical and statistical techniques for experimental optimization. Correlation induction strategies can be employed in RSM to achieve improved statistical inferences on experimental designs and sequential experimentations. Also, the search procedures currently employed by RSM algorithms can be improved by incorporating gradient deflection methods.

This dissertation has three major goals: (a) develop analytical results to quantitatively express the gains of using the common random number (CRN) strategy of variance reduction over direct simulation (independent streams or IS strategy) at each stage of

RSM, (b) develop a new RSM algorithm by incorporating gradient deflection methods in existing RSM algorithms, and (c) to conduct extensive empirical studies to quantify: (i) the use of CRN strategy over direct simulation in a standard RSM algorithm, and (ii) the gains of the new RSM algorithm over a standard existing RSM algorithm.

The method of CRN is one of the easiest correlation-induction strategies to employ and can be used judiciously in many multiple-run simulation experiments. Its use is therefore proposed for the sequential experiments used in RSM. Theoretical results that quantify the gains of using the CRN strategy over direct simulation at each stage of the RSM algorithm are developed. The existing RSM algorithm is further modified to improve its performance by incorporating gradient deflection strategies in conjunction with appropriate restarting criteria instead of using the method of steepest descent only.

In order to conduct an empirical investigation to study the gains of the CRN strategy over direct simulation, a response surface is created. The goal then is to find the settings of the independent variables that minimize the mean value of this response surface. This problem is solved under the two simulation strategies (independent streams and common random numbers). To justify the validity of these quantitative results, numerous different randomly selected starting points are chosen. Random selection of starting points provides an opportunity to make probabilistic assertions about the relative performance of the two simulation strategies. The relative performance of the two strategies is quantified with computational results using numerous performance measures that focus on different algorithm characteristics. The computational results are categorized into two classes which characterize the accuracy and speed of the RSM algorithm. An important statistic of interest which relates to both the speed and the accuracy of the algorithm is the average gain in response values per simulation run (or design point). The

overall evaluation of these performance measures clearly indicates overwhelming gains of using the CRN strategy over direct simulation.

In order to empirically investigate the use of the new RSM algorithm which incorporates certain gradient deflection methods augmented with some restarting criteria, over the standard existing one, a set of standard test functions is selected from the nonlinear optimization literature. Some randomness is incorporated in these test functions. The two algorithms are then employed to obtain best solutions. Evaluation of the relative performances of the two algorithms indicates improvement of the new algorithm over the existing one. Also, the use of different gradient deflection techniques are compared in the context of RSM which helps the practitioner choose among the various available methods. Three replications for solving the optimization problems from the same starting points are performed under each algorithm so that the conclusions from this empirical study are assertive in spite of the randomness in the system.

## Acknowledgements

I sincerely express my gratitude to my committee chairman, Dr. Jeff Tew for inspiring, encouraging and guiding me through this dissertation. He is a true friend and philosopher. I truly enjoyed working under him.

I am extremely thankful to Dr. Hanif Sherali for his contribution to this work. Some parts of this work would not have been possible without his guidance, encouragement, and support. Dr. Schmidt and Dr. Houck always kept me on my toes on logistics. I thank them for serving on my committee. Dr. Koelling was a great support for me throughout my entire graduate studies. His off-beat humor and encouragement helped me a lot. I sincerely appreciate his help and advice as my committee member. I would also like to thank Dr. John Kobza for agreeing to serve as a committee member. Dr. Peter Kiessler has a separate place in my life. His attitude to life will always be an inspiration.

An integral part of a graduate student's life is his/her office. Pack building will never be forgotten; and neither will the coffee pots or the coffee-stained mugs! The best part

ofcourse are the friends in the building. Life would never be as differently enjoyable without the super and honest friendship of Cihan, Kim, Koi, and Naresh. We kept each other going. I wish them all the best in their lives. The late-night sponge basketball games in the office corridor were an added attraction. Chandra and Koi used to be the competition (not!). Other great friends who left the Pack building, but provided a great time while they were here are Sridhar, Rajesh, Rajul, and Ritu. This work would not have been possible but for the great friendship and support from Alankar and Namita, and now their adorable twin daughters Aneesha and Anuja.

One of the most important couple due to whom I am here is my sister Geeta, and her husband Achyut. I sincerely thank them for the love, affection and support that they have constantly provided. True love and encouragement was provided by Shukla's parents which kept us going through thick and thin.

Finally, the three most important people in my life.... Aai, Baba, and my wife, Shukla. Nothing was or is possible without their love. One of the major reasons that I could complete my doctorate is my parents' constant love, support, and encouragement. No words can ever express my indebtedness to them.

This dissertation is dedicated to Shukla. She has understood and helped more than I can tell her. Nothing in my life is possible without her love and support.

# Table of Contents

<b>CHAPTER I Introduction</b> .....	<b>1</b>
<b>CHAPTER II Literature Review</b> .....	<b>7</b>
2.1 Notation and Definitions for Simulation Experiments .....	7
2.2 Variance Reduction Techniques .....	11
2.2.1 Single Population Experiments .....	12
2.2.2 Multipopulation Experiments .....	18
2.3 Metamodel Estimation in Simulation Experiments .....	21
2.4. Statistical Analysis for the First-Order Metamodel Under the Common Random Numbers Strategy .....	23
2.5 Response Surface Methodology .....	30
2.5.1 Overview .....	31
2.5.2 RSM with Computer Simulation .....	39
2.5.3 NLP Techniques in RSM .....	41
2.6 Quasi-Newton and Conjugate Gradient Methods .....	46
2.6.1 Quasi-Newton Methods .....	46
2.6.2 Conjugate Gradient Methods .....	49

2.6.3 Restarting Criteria .....	54
<b>CHAPTER III RSM Algorithm Under the CRN Strategy .....</b>	<b>56</b>
3.1. RSM algorithm and discussion .....	56
3.2. The First-Order Model under CRN .....	65
3.3 Gradient Search Procedure .....	69
3.4. The Second-Order model .....	74
3.5. Ridge Analysis .....	79
<b>CHAPTER IV Example .....</b>	<b>82</b>
4.1 Problem Statement .....	82
4.2 Computational Results .....	90
4.3 Empirical Study for a Ridge System .....	107
<b>CHAPTER V A Novel RSM Algorithm .....</b>	<b>110</b>
5.1 Modified RSM Algorithm .....	110
5.2 Gradient Deflection Methods .....	115
<b>CHAPTER VI Example .....</b>	<b>120</b>
<b>CHAPTER VII Conclusions and Future Research .....</b>	<b>134</b>
<b>References .....</b>	<b>136</b>
<b>Appendix 1 .....</b>	<b>143</b>

## List of Illustrations

Figure 1.	Search Procedure for Starting Point # 21 under the CRN strategy. . . . .	88
Figure 2.	Search Procedure continued. . . . .	89
Figure 3.	Scatter Plot for Locations of Starting Points. . . . .	91
Figure 4.	Scatter Plot for Locations of Stopping Points Under the CRN Strategy. . . . .	92
Figure 5.	Scatter Plot for Locations of Stopping Points Under the IS Strategy. . . . .	93
Figure 6.	Pie-Charts to compare CRN and IS strategies under RSM. . . . .	102
Figure 7.	Histogram of mean responses observed under the CRN strategy. . . . .	103
Figure 8.	Histogram of mean responses observed under the IS strategy. . . . .	104

# List of Tables

Table 1. Performance Measures Characterizing Accuracy. . . . .	100
Table 2. Performance Measures Characterizing Speed. . . . .	101
Table 3. Performance of test functions under different methods of RSM. . . . .	124
Table 4. Performance of test function # 3 under different restarting conditions. .	125
Table 5. Performance of test function # 6 under different restarting conditions. .	126
Table 6. Performance of test function # 8 under different restarting conditions. .	127
Table 7. Performance of test function # 3 under randomness. . . . .	130
Table 8. Performance of test function # 6 under randomness. . . . .	131
Table 9. Performance of test function # 8 under randomness. . . . .	132

# CHAPTER I Introduction

A scientist or engineer involved with process design and development often characterizes experimental problems as the investigation of the mean of some response variable of interest that can be expressed as a mathematical function,  $f$ , of  $k$  independent variables. The mean univariate response variable  $y$  can be expressed in terms of the independent variables for some functional relationship of unknown form as

$$y = f(x_1, x_2, \dots, x_k) + \varepsilon,$$

where  $\varepsilon$  is the unknown error term due to randomness in the system. Often the goal of the investigation is to find settings of the levels of the independent variables that optimize the mean of this response variable. Many different optimization techniques have been developed to achieve this goal. They can be classified as: (a) statistical techniques, and (b) deterministic techniques. Statistical techniques include response surface methodology (RSM) which typically attempts to determine the appropriate functional model representation for the mean of the response variable and then utilizes this information for prediction, exploration, and optimization of the mean response within the region in the independent variables. Deterministic techniques can be subdivided into two categories: (a) gradient search methods, and (b) non-gradient search methods. The gradient search methods (e.g. finite difference approximation method, etc.) use derivatives in determining the search directions, whereas the non-gradient search methods (e.g., Hooke-Jeeves, Rosenbrock, Nelder and Mead, etc.) use only functional evaluations during the

course of optimization. Optimization techniques can also be classified as: (a) adaptive, and (b) non-adaptive. Adaptive search techniques employ sequential search. That is, the results at each stage of the technique are used in the successive search for the optimum. RSM, Nelder and Mead, etc., are some of the examples of adaptive search techniques. In contrast, the non-adaptive search techniques, like random search, do not use the information of the findings of the search at any stage, but randomly select the settings for the independent variables, and the best mean response obtained is chosen as the optimum.

While attempting to solve optimization problems, the lack of an explicit mathematical expression of the problem frequently precludes the application of the standard methods of optimization that prove valuable in an analytical framework. In such situations, computer simulation can be used to obtain the mean response values for the required settings of the independent variables ( $x_1, x_2, \dots, x_k$ ). A procedure for optimizing the mean response value, which is in turn estimated through computer simulation experiments, will be referred to as a simulation-optimization technique.

The focus of this work is on the simulation-optimization technique of response surface methodology. RSM is a collection of mathematical and statistical techniques for experimental optimization. Since the functional relationship between the response variable and the independent variables is typically unknown, linear or quadratic regression models are usually employed to fit the sample data of the observed responses. These regression models are used to predict and explore the general vicinity of the optimum. Further analysis, typically involving quadratic models, is used to locate the optimum.

Simulation offers unusual opportunities for deliberately and advantageously inducing correlations between responses. This can be exploited to achieve improved statistical

inferences on experimental designs and sequential experimentations. In particular, these *correlation-induction strategies* can be applied to response surface methods. Although in recent studies on RSM, a part of the literature deals with minimum variance and minimum mean squared error designs for the first-order and second-order models, most of these papers do not consider variance reduction techniques used in conjunction with *sequential* simulation experimentation. This dissertation has three major goals: (a) develop analytical results to quantitatively express the gains of using the common random numbers (CRN) strategy over direct simulation (or the strategy of independent streams, which will be abbreviated as the IS strategy) at each stage of RSM, (b) develop a new RSM algorithm by incorporating the method of conjugate gradients in existing RSM algorithms, and (c) to conduct extensive empirical studies to quantify (i) the use of the CRN strategy over direct simulation in a standard RSM algorithm, and (ii) the gains of the modified RSM algorithm over traditional implementations.

The strategy of CRN is one of the easiest correlation-induction strategies to employ and can be judiciously used for many multiple-run simulation experiments. Its use is therefore proposed for the sequential experiments used in RSM. Theoretical results that quantify the gains of using the CRN strategy over direct simulation at each stage of the RSM algorithm will be developed. The existing RSM algorithm is further modified to improve its performance by incorporating the method of conjugate gradients instead of using the method of steepest descent *only*. In addition to the theoretical development, Monte-Carlo analyses are undertaken for two purposes. First, to quantify the gains of employing the CRN strategy in simulation experiments over direct simulations. Second, to quantify the use of the newly proposed RSM algorithm over a standard one.

In order to conduct such an extensive empirical investigation to quantify the use of CRN strategy over direct simulation, a response surface has been created. This response surface is generated using a computer simulation model, with the output of the simulation experiment having components of Monte-Carlo random error. The goal then is to find the settings of the independent variables that minimize the functional value of this response surface. This problem is then solved under the two simulation strategies.

The output of the simulation model is the sojourn time of jobs in a modified job-shop. The two independent variables under consideration are the mean inter-arrival time, and the mean service time for the machines in the job-shop. The response of interest is a function of the sojourn time, with some cost function added to it. This cost function penalizes the response as the inter-arrival times increase and service times decrease, and rewards the response when the converse occurs. The rationale for this penalty function is as follows. If the inter-arrival time is larger, then the job-shop receives a smaller number of jobs and loses on its profits. Also, if the service times are reduced, then the machines have to be made more efficient, which results in added costs. The cost function also has a product term added to it, to account for the joint effects of the two variables, which in turn adds to the complexity of the generated surface.

The surface created has a nearly flat terrain with a sudden dip in response values at some point in the interior. Within this valley, there is both, a global and a local optimum located not very far from each other. The response surface thus generated becomes a formidable optimization problem. Also, since the true optimum is known *a priori*, it permits a comparison of the optimum found when using different simulation strategies to solve the same problem.

To justify the validity of the quantitative results presented, numerous randomly selected starting points are chosen across the design space. Random selection of starting points provides an opportunity to make probabilistic assertions about the relative performance of the two simulation strategies. The relative performance of the two strategies is quantified with computational results using numerous performance measures which focus on different algorithm characteristics across all searches. The computational results are categorized into two classes: (a) accuracy and (b) speed of the RSM algorithm.

In order to empirically investigate the use of the new RSM algorithm over the standard existing one, a subset of standard test functions is selected from the nonlinear optimization literature. Some randomness is incorporated in these test functions. The two algorithms are then employed to obtain optimal solutions. Using performance measures similar to the ones described above, inferences are made on the gains of using the modified algorithm. Three replications for solving the optimization problems from the same starting points are performed under each algorithm so that conclusions from this empirical study are assertive in spite of the randomness in the system.

The rest of this document is organized as follows. Chapter 2 presents a review of notation and the relevant literature. The literature reviewed is classified into six sections: (a) simulation experiments, (b) variance reduction techniques, (c) metamodel estimation in simulation experiments, (d) statistical analysis under the CRN strategy, (e) response surface methodology (RSM), and (f) conjugate gradient and quasi-Newton methods. In Chapter 3 the standard existing RSM algorithm is presented. This presentation includes a definition and exposition of all steps comprising the algorithm accompanied by a detailed explanation of each step. Statistical analysis for the first-order model is summarized, and the analysis for the second-order model developed. Theoretical results for

gains under the CRN strategy over direct simulation for the first-order model, the gradient search procedure, and the second-order model are also developed. Chapter 4 describes in detail the problem used for the empirical investigation, and also presents computational results for the standard RSM algorithm under direct simulation and the CRN strategy. Chapter 5 presents the new RSM algorithm which incorporates gradient deflection methods. Chapter 6 illustrates the use of the new RSM algorithm by applying it to a set of standard test functions and providing computational results.

## CHAPTER II Literature Review

In this chapter we review the literature, introduce notation, and define important concepts used throughout this document. This chapter is divided into six topical sections: (a) notation and definitions for simulation experiments, (b) variance reduction techniques, (c) metamodel estimation in simulation experiments, (d) statistical analysis under the CRN strategy, (e) response surface methodology (RSM), and (f) conjugate gradient and quasi-Newton methods. In each section, we give a concise overview of the specific topic. In Section 2.1 we introduce notation and definitions used in simulation experiments. Section 2.2 focusses on variance reduction techniques used in simulation. Section 2.3 provides the statistical framework necessary to formally define a simulation experiment. In Section 2.4 we review the statistical analysis procedures under the CRN strategy. An overview of RSM and its use in optimization using computer simulation, as well as the use of NLP techniques in RSM, is discussed in Section 2.5. Finally, Section 2.6 discusses the conjugate gradient and quasi-Newton methods used in unconstrained nonlinear optimization.

### *2.1 Notation and Definitions for Simulation Experiments*

Throughout this document we use  $\mathbf{1}$ , to denote an  $r$ -dimensional column vector whose elements are all 1, and  $\mathbf{I}$ , to denote a  $(r \times r)$  identity matrix. For the purpose of this document, any bold-face upper-case letter will signify a matrix, whereas a bold-face lower-case letter will signify a vector. On occasion, in the development of the statistical

methodologies, use will be made of the following matrix operation. For any  $(t \times s)$  matrix  $\mathbf{A}$  and  $(m \times n)$  matrix  $\mathbf{B} = (b_{ij})$ , the Kronecker product of  $\mathbf{A}$  and  $\mathbf{B}$  is defined as the  $(mt \times ns)$  matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{A}b_{11} & \mathbf{A}b_{12} & \cdot & \cdot & \cdot & \mathbf{A}b_{1n} \\ \mathbf{A}b_{21} & \mathbf{A}b_{22} & \cdot & \cdot & \cdot & \mathbf{A}b_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{A}b_{m1} & \mathbf{A}b_{m2} & \cdot & \cdot & \cdot & \mathbf{A}b_{mn} \end{bmatrix}.$$

We define *response* to mean the output of a simulation experiment, and *factors* as the non-random inputs. We assume that the simulation analyst controls the values of the factors without error, and that there are  $d$  such factors comprising the experiment. A value of factor  $i$  is called a factor level, and is denoted by  $\phi_i, i = 1, 2, \dots, d$ . A particular design point in an experiment is identified by the specific levels of the  $d$  experimental factors and represented as  $\varphi = (\phi_1, \phi_2, \dots, \phi_d)$ . For the first-order model we assume that there is a linear relationship between the response and the selected setting of  $\varphi$ , that is,

$$y = f(\varphi) + \varepsilon, \tag{2.1}$$

where  $f$ , the metamodel of the response variable is linear in the unknown parameters that relate the response to the factor settings,  $\varphi$ , and  $\varepsilon$  represents the inability of  $f$  to determine  $y$ .

We now define the random number streams used to drive the simulation model. A simulation model is usually driven by streams of random numbers ( $\mathbf{r}$ ). These streams are sequences of real numbers scaled to the interval  $[0,1]$  and constructed to appear random in nature. We assume that  $g$  random number streams are used to drive the

simulation model. We denote this by  $\mathbf{R}_i = (r_{i1}, r_{i2}, \dots, r_{ig})$ . We let  $\mathbf{R}_i (i = 1, 2, \dots, m)$  be the set of  $m$  such sets of random number streams used at the  $i$ th design point. Thus, we have

$$y_{ij}(\mathbf{R}_i) = f(\varphi_i) + \varepsilon_{ij}(\mathbf{R}_i), \quad \text{for } i = 1, 2, \dots, m \text{ and } j = 1, 2, \dots, r; \quad (2.2)$$

where  $y_{ij}$  is the response at the  $i$ th design point and the  $j$ th replicate,  $\varphi_i$  is the setting of the  $d$  factors at the  $i$ th design point, and  $\varepsilon_{ij}$  is the error at the  $i$ th design point and  $j$ th replicate. Typically  $f$  is unknown and one of the objectives of the simulation analysis is to estimate this function. The estimation process usually involves two steps: (a) hypothesize a functional approximation of  $f$ , and (b) estimate any unknown parameters in the hypothesized approximation (see p. 210 of Neter, Wasserman, and Kutner, 1989).

For example, under the assumption that  $f$  is first-order and linear in the unknown parameters, equation (2.2) can be written as

$$y_{ij}(\mathbf{R}_i) = \beta_0 + \sum_{l=1}^k \beta_l x_l(\varphi_i) + \varepsilon_{ij}(\mathbf{R}_i), \quad \text{for } i = 1, 2, \dots, m \text{ and } j = 1, 2, \dots, r; \quad (2.3)$$

where  $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_k)'$  is the  $((k + 1) \times 1)$  vector of unknown metamodel coefficients;  $x_l (l = 1, 2, \dots, k)$  represent known functions of the factor settings, and  $y_{ij}$ ,  $\varphi_i$ , and  $\varepsilon_{ij}$  are as defined above.

Equation (2.3) can be written in matrix notation as :

$$\mathbf{y}_j = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}_j, \quad \text{for } j = 1, 2, \dots, r; \quad (2.4)$$

where  $y_j = (y_{1j}, y_{2j}, \dots, y_{mj})'$ , is the  $(m \times 1)$  vector of responses at the  $j$ th replication,  $X$  is a  $(m \times (k + 1))$  matrix whose first column is all ones and whose  $(i, j + 1)$ th element is  $x_j(\varphi_i)$ , ( $i = 1, 2, \dots, m$  and  $j = 1, 2, \dots, k$ ),  $\beta$  is defined above, and  $\varepsilon_j = (\varepsilon_{1j}, \varepsilon_{2j}, \dots, \varepsilon_{mj})'$  is the  $(m \times 1)$  vector of random errors.

In addition to input factors, the response variable may also be related to another type of input variable called *concomitant variable*. While it is assumed that the factors are under the control of the experimenter, concomitant variables are merely observed during the course of the experiment. A concomitant variable is observed independently, typically, at each of the levels of the factors during the experiment and assumed to be correlated to the corresponding response (see p. 7 of Kwon, 1991). For instance, the experimenter may wish to compare the effects of different drugs on a patient by measuring a response variable of interest. It is assumed that the body weight of the patient is correlated with the response variable, but is independent of drug type. In this example, drug type is the factor and body weight of the patient is the concomitant variable (see pp. 279-281 of Seber, 1977). If the concomitant variables are strongly correlated with the response variable, then by subtracting an appropriate linear function of the concomitant variables from the response variable, the unknown error term of the response variable can be counteracted (the use of concomitant variables is discussed in detail in Section 2.2.1). Thus, a statistical model including the concomitant variables *may* describe the response more accurately than with the factors only.

Now, consider the first-order linear statistical model from equation (2.3) with both factors and concomitant variables for one replication only:

$$y_{ij} = \beta_0 + \sum_{l=1}^k \beta_l x_l(\phi_i) + \sum_{h=1}^s c_{ih} \alpha_h + \varepsilon_{ij}^*, \text{ for } i = 1, 2, \dots, m, j = 1, 2, \dots, r; \quad (2.5)$$

where  $y_{ij}$ ,  $\beta_i$ ,  $x_l(\phi_i)$  are defined in (2.3),  $c_{ih}$  ( $h = 1, 2, \dots, s$ ) is the  $h$ th concomitant variable at the  $i$ th design point,  $\alpha_h$  ( $h = 1, 2, \dots, s$ ) is the coefficient of  $c_{ih}$ , and  $\varepsilon_{ij}^*$  represents the inability of the postulated model to determine  $y_{ij}$ .

Equation (2.5) can be represented in the matrix notation as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{C}\mathbf{a} + \boldsymbol{\varepsilon}^*, \quad (2.6)$$

where  $\mathbf{y}$ ,  $\mathbf{X}$ , and  $\boldsymbol{\beta}$  are as defined in (2.4),  $\boldsymbol{\varepsilon}^* = (\varepsilon_1^*, \varepsilon_2^*, \dots, \varepsilon_m^*)'$ ,  $\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_s)'$ , and  $\mathbf{C}$  is a  $(m \times s)$  matrix whose  $i$ th row is comprised of the  $s$  control variates at the  $i$ th design point.

To estimate the linear regression metamodel given in (2.4) using simulation experiments, several variance reduction techniques have been used. These methods under certain conditions reduce the variances on the unknown coefficients  $\beta_i$ , or counteract the unknown error term of the response variable. Some such variance reduction techniques are discussed in the next section.

## 2.2 Variance Reduction Techniques

Variance reduction techniques (VRTs) reduce the variance of the estimator by replacing the original sampling procedure by a new procedure that yields the same expected value but with a smaller variance. Often, attention is restricted to the estimation of the mean. Hence VRTs can be regarded as methods that reduce the variance of the estimate of the

mean response. This response can be the waiting time of a customer in the steady state, the total profit of a firm over the planning period, etc. Some of the different VRTs used are stratified sampling, control variates, importance sampling, antithetic variates, common random numbers, and the joint application of antithetic variates and common random numbers (see pp. 110-200 of Kleijnen, 1974). (A detailed treatment on VRTs can be found in Chapter 11 of Law and Kelton, 1991, Chapters 2 and 8 of Bratley, Fox, and Schrage, 1983, and Chapter 3 of Kleijnen, 1974.) This section is sub-divided into two sub-sections. Sections 2.2.1 and 2.2.2 discuss variance reduction strategies for single population and multipopulation simulation experiments, respectively.

### **2.2.1 Single Population Experiments**

Experiments which are comprised of only one design point are referred to as single population experiments. To reduce the variance of statistics of interest in such experiments, the analyst often resorts to replications. As the number of replications increases, the sample size of the population increases, and hence the variance of the estimate of the statistic of interest decreases. In order to achieve significant reduction in variance, often the number of replications needed is quite large and thus, prohibitively expensive. Fortunately, there are effective ways to reduce the variance with a fewer number of replications. One such way is the method of control variates.

We first discuss the method of a single control variate used for a single population, single response experiments. Let  $y$  be the univariate response with mean denoted by  $\mu_y$ , and let  $c$  be a control variate corresponding to  $y$ . We assume a linear relationship between the control variate and the response variable, and that we know the value of  $\mu_c = E[c]$ .

As an example (see p. 635 of Law and Kelton, 1991),  $y$  can be the average delay in queue for the first 100 customers in a simple queueing system, and  $c$  could be the average service times of the first 99 customers, so that we would know its expectation since, typically, we generate the service times from a known distribution. It is thus natural to expect that larger values of  $c$  will yield larger values of  $y$ . That is,  $y$  is positively correlated with  $c$ . We can therefore *adjust*  $c$  in order to partially control  $y$ . This adjustment is expressed in terms of the deviation of  $c$  from its expectation, that is,  $c - \mu_c$ . Let  $\alpha$  be a constant which has the same sign as the correlation between  $y$  and  $c$ . We obtain a *controlled* response denoted by  $y(c)$  by scaling the deviation  $c - \mu_c$  as

$$y(c) = y - \alpha(c - \mu_c). \quad (2.7)$$

Since  $E[y] = \mu_y$  and  $E[c] = \mu_c$ , we have  $E[y(c)] = \mu_y$ . That is,  $y(c)$  has the same mean as the unadjusted response  $y$  and could have lower variance than  $y$ . From (2.6), we see that

$$\text{var}(y(c)) = \text{var}(y) + \alpha^2 \text{var}(c) - 2\alpha \text{cov}(c, y), \quad (2.8)$$

so that  $y(c)$  is less variable than  $y$  if and only if

$$2\alpha \text{cov}(c, y) > \alpha^2 \text{var}(c), \quad (2.9)$$

which may or may not be true in general.

The goal is to find the optimal value of  $\alpha$  from equation (2.8) which yields the minimum variance. This is obtained by differentiating (2.8) with respect to  $\alpha$  and setting this derivative to zero which yields an optimal value for  $\alpha$ , denoted by  $\alpha^*$  as

$$\alpha^* = \frac{\text{cov}(c, y)}{\text{var}(c)}. \quad (2.10)$$

Using  $\alpha^*$  in equation (2.8), we obtain the minimum-variance controlled response, say,  $y^*(c)$ , given by

$$\text{var}(y^*(c)) = \text{var}(y) - \frac{\text{cov}(y,c)^2}{\text{var}(c)} = (1 - \rho_{yc}^2)\text{var}(y), \quad (2.11)$$

where  $\rho_{yc}$  is the correlation coefficient between  $c$  and  $y$ . The optimal estimator of the response thus obtained can never be more variable than the uncontrolled estimator  $y$ .

We next discuss the method of multiple control variates used for single population (one design point), univariate response situations. A comprehensive discussion of control variates is given in Kwon (1991). Lavenberg and Welch (1981) studied the use of multiple control variates for a single population simulation experiment with a single response. Lavenberg, Moeller, and Welch (1982) developed some types of control variates and applied them to a closed queueing network problem. Later, Wilson and Pritsker (1984) developed procedures for using standardized control variates under replication and regenerative analyses.

Let  $y$  be a response from a single simulation run. Let  $\mathbf{c} = (c_1, c_2, \dots, c_q)'$  be a vector of  $q$  control variates having a known mean  $\boldsymbol{\mu}_c = E[\mathbf{c}]$ . The goal is then to predict and counteract the unknown deviation  $y - \mu_y$  by subtracting from  $y$  an appropriate linear transformation of the associated known deviation  $\mathbf{c} - \boldsymbol{\mu}_c$ . We denote the adjusted response  $y(\mathbf{c})$ , which is given by

$$y(\mathbf{c}) = y - \boldsymbol{\alpha}(\mathbf{c} - \boldsymbol{\mu}_c), \quad (2.12)$$

where  $\alpha$  is a  $(1 \times q)$  matrix of control coefficients. Let  $\Sigma_c$  denote the  $(q \times q)$  covariance matrix of  $\mathbf{c}$ , and  $\sigma_{yc}$  denote the  $(q \times 1)$  covariance vector between the response  $y$  and  $\mathbf{c}$ . Then the generalized variance of  $y(\mathbf{c})$  is given by

$$\text{var}(y(\mathbf{c})) = \sigma_y^2 - 2\sigma_{yc}\alpha + \alpha'\Sigma_c\alpha. \quad (2.13)$$

The value of  $\alpha$  that minimizes (2.13) is given by

$$\alpha' = \sigma_{yc}\Sigma_c^{-1}, \quad (2.14)$$

and the resulting minimum variance of  $y(\mathbf{c})$  is

$$\text{var}(y(\mathbf{c})) = \sigma_y^2 - \sigma_{yc}'\Sigma_c^{-1}\sigma_{yc} = (1 - R_{yc}^2)\sigma_y^2, \quad (2.15)$$

where  $R_{yc}^2 = \sigma_y^{-2} - \sigma_{yc}'\Sigma_c^{-1}\sigma_{yc}$  is the multiple correlation coefficient between  $y$  and  $\mathbf{c}$ . The stronger the correlation between a set of control variates  $\mathbf{c}$  and the response  $y$ , the greater the potential variance reduction of the estimator. Further, across  $r$  independent replications comprising the simulation experiment it is assumed that  $y_i$  and  $\mathbf{c}_i$  have a multivariate normal distribution ( $i = 1, 2, \dots, r$ ):

$$\begin{bmatrix} y_i \\ \mathbf{c}_i \end{bmatrix} \sim N_{1+q} \left[ \begin{bmatrix} y \\ \mu_c \end{bmatrix}, \begin{bmatrix} \sigma_y^2 & \sigma_{yc} \\ \sigma_{yc}' & \Sigma_c \end{bmatrix} \right]. \quad (2.16)$$

We can thus represent the response as

$$y_i = \mu_y + (\mathbf{c}_i - \mu_c)'\alpha + \varepsilon_i, \quad \text{for } i = 1, 2, \dots, r, \quad (2.17)$$

where  $y_i$  and  $\mathbf{c}_i$  are the  $i$ th observations of the response and the control variates respectively, and  $\varepsilon_i \sim IID N(0, \sigma^2)$ . In matrix form, (2.17) can be written as

$$\mathbf{y} = \mu_y \mathbf{1} + \mathbf{C}\boldsymbol{\alpha} + \boldsymbol{\varepsilon} \quad (2.18)$$

where  $\mathbf{y} = (y_1, y_2, \dots, y_r)'$ ,  $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_r)'$ , and  $\mathbf{C}$  is a  $(r \times q)$  matrix of control variates whose  $i$ th row is given by  $\mathbf{c}_i'$ .

We now consider the case where  $\Sigma_c$  and  $\sigma_{y_c}$  are known. In this case,  $y(\bar{\mathbf{c}})$  in (2.12) is unbiased and has minimum variance. Lavenberg, Moeller, and Welch (1982) defined the quantity  $(1 - R_{y_c}^2)$  to be the minimum variance ratio. In the case of  $r$  independent replicates, adjusted response  $\bar{y}(\bar{\mathbf{c}})$  is given by

$$\bar{y}(\bar{\mathbf{c}}) = \bar{y} - (\bar{\mathbf{c}} - \boldsymbol{\mu}_c)' \boldsymbol{\alpha}, \quad (2.19)$$

where  $\bar{y}$  is the sample mean of the response and  $\bar{\mathbf{c}}$  is a mean vector of the control variates taken across all  $r$  replicates.

We next consider the case where  $\Sigma_c$  and  $\sigma_{y_c}$  are unknown. In this case  $\hat{\boldsymbol{\alpha}}$ , the estimator of  $\boldsymbol{\alpha}$  is given by

$$\hat{\boldsymbol{\alpha}}' = S_{y_c} S_c^{-1} \quad (2.20)$$

where  $S_{y_c}$  and  $S_c$  are the sample estimators of  $\sigma_{y_c}$  and  $\Sigma_c$  respectively. Lavenberg and Welch (1981), and Lavenberg, Moeller, and Welch (1982) showed that the variance of  $\bar{y}(\bar{\mathbf{c}})$  is

$$\text{var}(\bar{y}(\bar{\mathbf{c}})) = \left( \frac{r-2}{r-q-2} \right) (1 - R_{y_c}^2) \frac{\sigma_y^2}{r}. \quad (2.21)$$

They defined the loss factor as the amount by which the variance is increased due to the use of  $\hat{\boldsymbol{\alpha}}$  in (2.20) instead of  $\boldsymbol{\alpha}$  in (2.14) as:

$$\text{loss factor} = \frac{r-2}{r-q-2}. \quad (2.22)$$

The effect of control variates is measured by the product form of the loss factor and the minimum variance ratio, which yields

$$\left(\frac{r-2}{r-q-2}\right)(1-R_{yc}^2). \quad (2.23)$$

Thus, the use of control variates is effective if  $\frac{q}{r-2} < R_{yc}^2$ . The loss factor becomes a critical problem if the number of control variates is large.

For a general class of closed queueing networks ( $q$  service stations,  $d$  different types of customers, and  $N$  customers of all types) which allow priorities and blocking, Lavenberg, Moeller, and Welch (1982) developed three types of control variates: (a) service time variable, (b) flow variable, and (c) work variable. The control variates are used to estimate the response of interest. After extensive experimentation, the authors concluded that (a) confidence intervals of the mean of the responses of interest using the control variates method were substantially reduced compared to those obtained without using control variates, (b) a loss factor appeared to inflate the minimum variance of the estimator correctly, (c) work control variates yielded the smallest variance of the estimator provided the loss factor was not too large, and (d) a regression based method of applying control variates produced confidence intervals having proper coverage.

Wilson and Pritsker (1984a, 1984b) used standardized control variates, and performed a set of simulation experiments on a variety of closed and mixed queueing networks. With the replication estimation scheme, their standardized control variates yielded variance reductions ranging from 20% to 90%, and confidence interval half-length reductions

between 10% and 70%. We next discuss variance reduction techniques for multipopulation simulation experiments.

## **2.2.2 Multipopulation Experiments**

Experiments which are comprised of multiple design points are referred to as multipopulation experiments. Simulation offers unusual opportunities for deliberately and advantageously inducing correlation, positive or negative, among observations. When comparing policies, the use of random number streams common to all of these policies offers a fairer comparison than would statistically independent streams since one source of variability has been removed by testing all policies under the same conditions (see p. 42 of Bratley, Fox, and Schrage, 1983). We next discuss the correlation-induction strategies used in simulation experiments.

Before starting a discussion on correlation-induction strategies, a note of caution is warranted. Unfortunately there is no general proof that correlation-induction strategies produce the desired variance reduction. In fact, there are examples of these techniques not yielding the desired results. For example, Nelson (1987) showed that the primary goal of variance reduction techniques, which is improving the point estimator performance, can actually deteriorate the interval estimator performance. Careful implementation of these techniques is therefore required. Common random numbers is one such correlation-induction technique, and is discussed next.

When the same set of random number streams is used at two design points, the two output responses tend to exhibit positive correlation (see p. 614 of Law and Kelton, 1991; p. 46 of Bratley, Fox, and Schrage, 1983; and, p. 200 of Kliejnen, 1974). If the

same streams of random numbers were used in two different simulations (common random numbers strategy), one producing a univariate output  $y_1$  and the other an univariate output  $y_2$ , then we expect  $\text{cov}(y_1, y_2) = \sigma^2\rho_+, > 0$ . The variance of  $(y_1 - y_2)$  is given by

$$\text{var}(y_1 - y_2) = \text{var}(y_1) + \text{var}(y_2) - 2\text{cov}(y_1, y_2).$$

Consequently, the statistic  $y_1 - y_2$  has a smaller variance than would occur with independent streams ( $\text{cov}(y_1, y_2) = 0$ ). Thus, common random numbers allows for detection of smaller differences between  $y_1$  and  $y_2$  than does the method of independent streams.

We see that the idea of the CRN strategy is to compare alternative systems under similar experimental conditions to improve confidence that observed differences in performance are due to differences in the system design rather than to differences in the experiment itself (see p. 614 of Law and Kelton, 1991). Next, we discuss the method of antithetic variates.

By contrast, the antithetic variates strategy induces negative correlations between responses (see p. 628 of Law and Kelton, 1991; p. 54 of Bratley, Fox, and Schrage, 1983; and, p. 186 of Kleijnen, 1974). This technique is implemented by generating one response from a set of random number streams  $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_g)$ , and the other response from a set of random number streams that are antithetic to those in  $\mathbf{R}$ , that is,  $\bar{\mathbf{R}} = (1 - \mathbf{r}_1, 1 - \mathbf{r}_2, \dots, 1 - \mathbf{r}_g)$ . If  $y_1$  and  $y_2$  are the corresponding univariate outputs, then we expect  $\text{cov}(y_1, y_2) = \sigma^2\rho_- < 0$ , so that  $\frac{(y_1 + y_2)}{2}$  has a smaller variance than occurs when independent streams are used.

Schruben and Margolin (1978) recommended a correlation induction strategy for a special class of multipopulation experiments. This strategy partitions the design matrix  $\mathbf{X}$  given in (2.4) into two orthogonal blocks. A set of common random number streams,

$\mathbf{R}$ , is applied to the first block, and a set of random number streams antithetic to the ones in the first block,  $\bar{\mathbf{R}}$ , is applied to the second block. Their strategy, under certain restrictions effectively combines the strategies of common random numbers and antithetic variates in the same experiment in order to reduce the variance of the metamodel coefficient,  $\beta$ , estimates.

We next briefly review the literature which deals with application of control variates to multipopulation simulation experiments. The development of the analytical results for achieving the desired variance reduction is similar to that presented in the Section 2.2.1.

Nozari, Arnold, and Pegden (1984) extended the results of the single population case considered by Lavenberg, Moeller, and Welch (1982) to a multipopulation case in a direction different from the work by Rubinstein and Marcus (1986). Kwon (1991), Tew and Kwon (1993), and Tew and Wilson (1992) developed combined correlation induction strategies. Kwon (1991) developed three combined methods utilizing antithetic variates and control variates for improving the estimation of the mean response in a single population model. He illustrated that the combined method using antithetic variates for non-control variate stochastic components, and independent streams for the control variates yields better results than applying either antithetic variates or control variates individually for several selected methods. He extended the application of this strategy to multipopulation simulation models. This combined method was further extended to incorporate the Schruben-Margolin method to estimate metamodel parameters of a multipopulation simulation model. Finally, he proposed a new approach utilizing control variates under Schruben-Margolin strategy for improved estimation in a first-order linear model.

Tew and Wilson (1992) offered another combined correlation-based variance reduction technique. Their procedure combines: (a) the Schruben-Margolin strategy for metamodel estimation, and (b) a metamodel estimation scheme based on the method of control variates. They showed their procedure to be superior to each of the conventional correlation-based VRT considered individually, under certain specified conditions on the dependency structure of the simulation outputs and with respect to  $D$ ,  $E$ , and  $A$ -optimality criteria.

In the next section we provide the statistical framework needed for simulation experiments conducted under independent streams (direct simulation).

### ***2.3 Metamodel Estimation in Simulation Experiments***

To attain a functional relationship between a univariate response and the levels of the factors of interest, the unknown coefficients  $\beta = (\beta_0, \beta_1, \dots, \beta_k)'$  in the models (2.3) and (2.4) need to be estimated. In order to estimate these parameters efficiently, at each design point the simulation experiment is often performed several times. This yields a good estimate of the error term variation. However, in simulation much work has been done on acquiring estimates of the error term variation with exactly *one* run performed at each design point (see Section 5 of Schruben and Margolin, 1978). Unlike all other forms of statistical experimentations, simulation experiments offer the researcher a high level of control over the variation in the output response. This control is attained by judicious choice of the random number streams used to drive the random components of the simulation model as seen in Section 2.2.2.

In order to estimate  $\beta$ , we proceed by assuming that  $\varepsilon_j$  ( $j = 1, 2, \dots, r$ ) in (2.4) has the following multivariate normal distribution:

$$\varepsilon_j \sim N_m(\mathbf{0}_m, \Sigma), \text{ for } j = 1, 2, \dots, r, \quad (2.24)$$

where  $\mathbf{0}_m$  is a  $(m \times 1)$  vector of zeros and  $\Sigma$  is a  $(m \times m)$  covariance matrix, such that the distribution of  $\varepsilon$  is nondegenerate. (Typically, in classical linear models  $\Sigma = \sigma^2 \mathbf{I}_m$ , that is, the error terms are uncorrelated across design points.)

From (2.4) and (2.24) we get that

$$y_j \sim N_m(\mathbf{X}\beta, \Sigma), \text{ for } j = 1, 2, \dots, r. \quad (2.25)$$

Under these assumptions, and for  $m > k + 1$  (see p. 210 of Neter, Wasserman, and Kutner, 1989), the ordinary least-squares estimate of  $\beta$ ,

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\bar{y}, \quad (2.26)$$

has the following distribution:

$$\hat{\beta} \sim N_{k+1}(\beta, (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\Sigma\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}), \quad (2.27)$$

where  $\bar{y} = \sum_{j=1}^r y_j/r$ . Thus, knowing the distribution of  $\hat{\beta}$ , the ordinary least-squares estimator of  $\beta$ , we can obtain  $100(1 - \alpha)\%$  confidence intervals and hypotheses tests on  $\beta$  and its elements (see p. 114 of Myers and Milton, 1991).

Such statistical analysis would not be valid if simulation experiments are performed under experimental designs which incorporate correlation-induction strategies due to the fact that responses obtained at different design points under such strategies are correlated. This presents the need to develop statistical analysis methods under such experimental settings. Nozari, Arnold, and Pegden (1987) developed methods for conducting

statistical analysis under the Schruben-Margolin strategy. Joshi and Tew (1993) developed similar methods under the CRN strategy which are discussed in detail in the next section.

## ***2.4. Statistical Analysis for the First-Order Metamodel***

### ***Under the Common Random Numbers Strategy***

In this Section we give a brief overview of the procedure to conduct statistical analysis for the first-order model under the CRN strategy. We also present theoretical results leading to the expected reduction in variance for the first order model under the CRN strategy over the independent streams case.

Recall from Section 2.1 that we defined  $\mathbf{y}$  to be a  $mr$ -dimensional vector of observations which has the multivariate normal distribution. The model under consideration is the one defined in equation (2.4) which can either be a first-order or a second-order model. Under the CRN technique, the same set of  $g$  random number streams  $\mathbf{R}_i = (r_{i1}, r_{i2}, \dots, r_{ig})$ , is applied to all  $m$  design points in the  $i$ th replication ( $i = 1, 2, \dots, r$ ).

When the same set of random number streams is used at two design points, we assume the following

1. A positive correlation,  $\rho_+$ , is induced between the two responses.
2.  $\rho_+$  is a constant, and does not depend on the specific set of seeds or the specific pair of design points. That is,  $\text{corr}(y_i, y_j) = \rho_+$  (for all  $i, j$ , and  $i \neq j$ ).
3. The variance is homogeneous over the entire design space, and is denoted by  $\sigma^2$ .

Under these assumptions, we have:

$$\text{cov} \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_r \end{bmatrix} = \Xi^{(CRM)} = \Sigma^{(CRM)} \otimes \mathbf{I}_r, \quad (2.28)$$

where

$$\Sigma^{(CRM)} = \sigma^2 \begin{bmatrix} 1 & \rho_+ & \cdot & \cdot & \cdot & \rho_+ \\ \rho_+ & 1 & \cdot & \cdot & \cdot & \rho_+ \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \rho_+ & \rho_+ & \cdot & \cdot & \cdot & 1 \end{bmatrix} \quad (2.29)$$

is a  $(m \times m)$  matrix of covariances between responses from the same replicate. Note that the covariance structure of  $\Sigma^{[CRM]}$  does not have the diagonal form, and hence, simple linear model theory cannot be applied directly to this model. Heikes, Montgomery, and Rardin (1976) proposed an approach to statistical analysis where they performed statistical inferences about multiple systems which have been simulated using common random numbers. They applied Graybill's method in the simulation environment. Kleijnen (1979) showed that the procedure proposed by Heikes, Montgomery, and Rardin (1976) can be replaced by a simple student t-statistic combined with a Bonferroni inequality. The method suggested by Joshi and Tew (1993) parallels the one developed by Nozari, Arnold, and Pegden (1987) for the Schruben-Margolin strategy, and is discussed next.

The method by Joshi and Tew (1993) uses an orthogonal transformation to transform the covariance matrix in (2.29) to that of a diagonal form, so that the theory of simple linear models can be used to estimate the unknown metamodel parameters, form confi-

dence intervals, and also perform the usual statistical tests on these parameters with the transformed responses. Once we know how these transformed responses can be used to perform these tasks, we can rewrite the procedures in terms of the original (untransformed) responses by taking the inverse transformation; this can be done, of course, only if the transformation is invertible (for obvious reasons we limit our discussion to such transformations).

The aim is to get a diagonal structure for  $\text{cov}(\mathbf{y}_i)$ ,  $i = 1, 2, \dots, m$ . This is done by applying the  $m \times m$  orthogonal transformation  $\Gamma^{(CRM)}$ , where

$$\Gamma^{(CRM)} = \begin{bmatrix} (m^{-1/2} \mathbf{1}'_m) \\ \mathbf{C}'_m \end{bmatrix}, \quad (2.30)$$

where  $\mathbf{C}_m$  is a  $m \times (m - 1)$  matrix of constants such that  $(m^{-1/2} \mathbf{1}_m | \mathbf{C}_m)$  is orthogonal. Note that the orthogonal matrix just described has  $\mathbf{1}_m$ , a  $m$ -dimensional column vector of ones, augmented by the matrix  $\mathbf{C}_m$ , and all the elements of this matrix are multiplied by the scalar  $m^{-1/2}$ . Define ( $i = 1, 2, \dots, r$ ):

$$\mathbf{y}_i^* = \begin{bmatrix} y_{1i}^* \\ y_{2i}^* \end{bmatrix} = \Gamma^{(CRM)} \mathbf{y}_i, \quad \mathbf{D}^* = [\mathbf{C}'_m \mathbf{D}], \quad (2.31)$$

$$\mathbf{X}^* = \Gamma^{(CRM)} \mathbf{X} = \begin{bmatrix} m^{(1/2)} \mathbf{0} \\ \mathbf{0} \quad \mathbf{D}^* \end{bmatrix}, \quad (2.32)$$

and

$$\Sigma^{(CRM)*} = \Gamma^{(CRM)} \Sigma^{(CRM)} \Gamma^{(CRM)'} = \begin{bmatrix} \lambda_1^2 & \mathbf{0} \\ \mathbf{0} & \lambda_2^2 \mathbf{I}_{(m-1)} \end{bmatrix}, \quad (2.33)$$

where  $y_{1i}^*$  is a scalar,  $\mathbf{y}_{2i}^*$  is  $(m-1) \times 1$ ,  $\lambda_1^2 = \sigma^2(1 + (m-1)\rho_+)$ , and  $\lambda_2^2 = \sigma^2(1 - \rho_+)$ .

Thus, by the definition of  $y_{1i}^*$ , and  $\mathbf{y}_{2i}^*$  in (2.31), we have,

$$y_{1i}^* = \left[ m^{-1/2} \sum_{j=1}^m y_{ij} \right] \quad (2.34)$$

and

$$\mathbf{y}_{2i}^* = [\mathbf{C}'_m \mathbf{y}_i], \quad (2.35)$$

Thus, we have:

$$\mathbf{y}_i \sim N_m(\mathbf{X}^* \boldsymbol{\beta}, \boldsymbol{\Sigma}^{(CRM)*}). \quad (2.36)$$

We see that by applying the transformation  $\boldsymbol{\Gamma}^{(CRM)}$  to the response vector  $\mathbf{y}$ , the transformed vector of responses,  $\mathbf{y}^*$ , is obtained whose covariance matrix is diagonal. Also, this transformation is invertible and does not involve any unknown parameters, so that any optimal procedure based on  $\mathbf{y}_i^*$  is also optimal among procedures based on  $\mathbf{y}_i = \boldsymbol{\Gamma}^{(CRM)} \mathbf{y}_i^*$ . Inspection of (2.33) clearly indicates that the model involving  $\mathbf{y}^*$  is really two separate ordinary linear models, one involving  $(y_{1i}^*, \beta_0, \lambda_1^2)$ , and one involving  $(\mathbf{y}_{2i}^*, \boldsymbol{\beta}_1, \lambda_2^2)$ .

We now state the results derived in Joshi and Tew (1993) for conducting statistical analysis under the CRN strategy. The results yield the following:

- an optimal (UMVU) estimate of  $\boldsymbol{\beta}$ ,
- optimal tests (UMP and invariant) on  $\boldsymbol{\beta}$ ,
- a confidence interval for  $\beta_0$ ,

- confidence intervals for  $\beta_1$ , and
- joint confidence intervals for  $\beta_0$  and  $\beta_1$ .

Let  $\hat{\beta}$  and  $\hat{\sigma}^2$  be the usual unbiased estimators for this model, that is :

$$\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1)' = (\mathbf{G}'\mathbf{G})^{-1}\mathbf{G}'\mathbf{y}, \quad (2.37)$$

and

$$\hat{\sigma}_1^2 = \frac{(\|\mathbf{y} - \mathbf{G}\hat{\beta}\|^2)}{mr - p}, \quad (2.38)$$

where

$$\mathbf{G} = \begin{bmatrix} \mathbf{X} \\ \mathbf{X} \\ \cdot \\ \cdot \\ \mathbf{X} \end{bmatrix}. \quad (2.39)$$

Note that  $\hat{\sigma}^2$ , the unbiased estimator of  $\sigma^2$  is model dependent. For this situation, Joshi and Tew (1993) proved each of the following results:

**Result 1:**  $\hat{\beta}$  in (2.37) is the optimal (UMVU) estimator of  $\beta$ .

**Result 2:**

$$\frac{[(mr)^{1/2}(\hat{\beta}_0 - \beta_0)]}{\hat{\lambda}_1} \sim t_{r-1},$$

where

$\hat{\lambda}^2$  is defined to be sample variation *between* replicates. Then,

$$\hat{\lambda}_1^2 = m \sum_{j=1}^r \frac{(\bar{y}_j - \bar{y}_{..})^2}{r}, \quad (2.40)$$

where  $y_{ij}$  denotes the  $i$ th observation of the  $j$ th replication ( $i = 1, 2, \dots, m$ , and  $j = 1, 2, \dots, r$ ).  $\bar{y}_j$  is defined as the mean of the observations across each replication, and  $\bar{y}_{..}$  is the overall mean of the observations. Hence,

$$\bar{y}_j = \sum_{i=1}^m \frac{y_{ij}}{m}, \quad \bar{y}_{..} = \sum_{j=1}^r \frac{\bar{y}_j}{r}. \quad (2.41)$$

**Result 3:** The optimal test for testing  $H_0 : \mathbf{H}\beta_1 = \mathbf{0}_h$  vs  $H_1 : \mathbf{H}\beta_1 \neq \mathbf{0}_h$ , where  $\mathbf{H}$  is a known  $(h \times k)$  matrix of rank  $h < k + 1$ , rejects  $H_0$  if

$$f^* = \frac{f\hat{\sigma}_1^2}{\hat{\lambda}_2^2} > F_{h, n-k-1-r}^\alpha, \quad (2.42)$$

where

$$\hat{\lambda}_2^2 = \frac{(mr - m)\hat{\sigma}_2^2 - m \sum_{i=1}^r (\bar{y}_i - \bar{y}_{..})^2}{r(m-1)}, \quad (2.43)$$

$\hat{\lambda}_1^2$  is the estimated variance, and  $\hat{\sigma}_2^2$  is given by

$$\hat{\sigma}_2^2 = [m(r-1)]^{-1} \sum_{j=1}^r \sum_{i=1}^m (y_{ij} - \bar{y}_{.j})^2. \quad (2.44)$$

The statistic  $f$  is the usual test statistic for testing  $H_0: \mathbf{H}\beta_1 = \mathbf{0}_h$  vs  $H_1: \mathbf{H}\beta_1 \neq \mathbf{0}_h$ , where  $\mathbf{H}$  is a known  $h \times (k + 1)$  matrix of rank  $h < k + 1$ , when  $\hat{\text{cov}}(\mathbf{y}_i) = \sigma^2 \mathbf{I}_m$ . That is,

$$f = \frac{r(\hat{\mathbf{H}}\hat{\beta}_1)'(\mathbf{H}(\mathbf{D}'\mathbf{D})^{-1}\mathbf{H}')^{-1}(\hat{\mathbf{H}}\hat{\beta}_1)}{h\hat{\sigma}_1^2}.$$

They further let  $HW_l$  denote the half-width of the  $100(1 - \alpha)\%$  simultaneous confidence intervals for the set of  $\mathbf{l}'\mathbf{H}\beta_1$ , for all  $\mathbf{l} \in R^k$ , when  $\hat{\text{cov}}(\mathbf{y}_i) = \sigma^2 \mathbf{I}_m$ . That is,

$$HW_l = \hat{\sigma}_1 \left[ h \frac{F_{h,n-k-1}^\alpha \mathbf{l}'\mathbf{H}(\mathbf{D}'\mathbf{D})^{-1}\mathbf{H}'\mathbf{l}}{r} \right]^{1/2}. \quad (2.45)$$

Note that  $\hat{\sigma}_1^2$  is another estimator of  $\sigma^2$ , and is model *independent*.

The  $100(1 - \alpha)\%$  simultaneous confidence intervals for the set of  $\mathbf{l}'\mathbf{H}\beta_1$  for all  $\mathbf{l} \in R^k$  when  $\Xi = \Sigma \otimes \mathbf{I}$ , are given by:

$$\mathbf{l}'\mathbf{H}\beta_1 \in \mathbf{l}'\mathbf{H}\hat{\beta}_1 \pm \frac{(\hat{\lambda}_2^2) F_{h,mr-k-1-r}^\alpha}{(\hat{\sigma}_1^2 F_{h,mr-k-1}^\alpha)^{(1/2)}} HW_l. \quad (2.46)$$

The next result involves inferences regarding both  $\beta_0$  and  $\beta_1$  simultaneously.

**Result 4:** A size  $\alpha$  procedure for testing  $H_0: \mathbf{K}\beta = \mathbf{0}_l$  vs  $H_1: \mathbf{K}\beta \neq \mathbf{0}_l$ , where  $\mathbf{K}$  is a  $(l \times k + 1)$  known matrix of rank  $l \leq k + 1$ , is to reject  $H_0$  if

$$\frac{r(r-l)}{l(r-1)} \hat{\beta}'\mathbf{K}'(\mathbf{K}\hat{\Delta}\mathbf{K}')^{-1}\mathbf{K}\hat{\beta} > F_{l,r-l}^\alpha \quad (2.47)$$

where  $\hat{\Delta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{S}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$ , and

$$\mathbf{S} = \frac{1}{r-1} \sum_{i=1}^r (y_i - \bar{y})(y_i - \bar{y})', \quad \text{where} \quad \bar{y} = \frac{1}{r} \sum_{i=1}^r y_i. \quad (2.48)$$

The set of  $100(1 - \alpha)\%$  simultaneous confidence intervals for  $\mathbf{l}'\mathbf{K}\boldsymbol{\beta}$  is then given by:

$$\mathbf{l}'\mathbf{K}\boldsymbol{\beta} \in \mathbf{l}'\mathbf{K}\boldsymbol{\beta} \pm \left[ l \frac{(r-1)}{r(r-l)} F_{l,r-l}^{\alpha} \mathbf{l}'\mathbf{K}\hat{\Delta}\mathbf{K}'\mathbf{l} \right]^{1/2} \quad (2.49)$$

for all  $\mathbf{l} \in R^k$ .

As shown in (2.33),  $\lambda^2_2 = \sigma^2_2(1 - \rho_+)$  is the variance of the components of  $\boldsymbol{\beta}_1$ . We note that these variances are reduced by a factor  $(1 - \rho_+)$  under the CRN strategy when compared to those under direct simulation. The variance of the components of  $\boldsymbol{\beta}_1$  can play an important role in search procedures used in RSM. This becomes evident after we review a standard RSM algorithm and its use as a simulation-optimization technique which is the theme of the next section.

## 2.5 Response Surface Methodology

RSM was first proposed by Box G. E. P., and Wilson K. B. in 1951 in the study of optimization problems in chemical process engineering. In general, RSM combines experimental design and regression analysis to find the relationship between the mean of the response variable  $E[y]$ , and the independent factors  $x_i$ s, given by say,

$$E[y] = f(x_1, x_2, \dots, x_k),$$

where  $f$  is some function. An overview of RSM is provided in Section 2.5.1. Section 2.5.2 discusses RSM used in the context of simulation experimentation. Nonlinear

programming (NLP) techniques applied to RSM are discussed in Section 2.5.3. We assume for the purpose of this discussion that the problem at hand is an unconstrained minimization problem, and that the minimum is known to lie within a given region of  $E^n$ . It is further assumed that the functional form of the problem is unknown, and that an estimate of  $E[y]$  at any given setting of the independent variables,  $x_1, x_2, \dots, x_k$ , is obtained through computer simulation.

### 2.5.1 Overview

Typically, to start the RSM procedure, an experiment is designed in a small sub-region of the factor space, and a low order polynomial (usually first-order) is used to represent the data obtained from the responses. This helps the practitioner to deduce which region should be studied next. The next region to be studied is reached by the method of steepest descent (for minimization problems). If the goal is to minimize the response, this method aims at climbing down the response surface rather than exploring the whole region, and its success depends on the assumption that the ultimate minimum can be reached by a rising path (see p. 503 of Davies O. L., 1956). That is, we assume that as the search moves down the surface (decreasing response), at some point there will be curvature evident, after which the response no longer improves or starts increasing. At this point another experiment is designed which will detect curvature and determine effects up to second order. Canonical analysis is performed to locate the stationary point of the fitted surface. Depending upon the results of the canonical analysis, further exploration is carried out. For example, if the stationary point is found to be a saddle point, then ridge analysis is performed. If the stationary point is found to be a mini-

mum, then the area around this point is explored a bit further and the optimum reported.

If the experimenter has a prior notion of the general vicinity of the location of the optimum, then Myers (1976) gives a stepwise procedure that can be generally described as follows (see p. 88 of Myers(1976)) :

Step 1: The experimenter fits a first-order response model in some restricted region of variables  $x_1, x_2, \dots, x_k$ .

Step 2: The information from Step 1 is used to locate a path of steepest descent.

Step 3: A series of experiments is conducted along the path until no additional decrease in response is evident.

Step 4: Steps 1, 2, and 3 are repeated, using the *new* region, the one which seems to be promising as indicated by Step 2.

Step 5: If curvature is evident and the experimenter is satisfied that he can obtain little or no additional information from the method, a more elaborate experiment is conducted.

In Step 1, typically, replications of a factorial or fractional factorial experiment are performed. The unknown parameters of the fitted linear model are then computed. Judicious selection of the experimental design is needed in order to obtain better estimates of the unknown parameters, like minimum variance, etc. The first-order linear model is represented as

$$y_{ij} = \beta_0 + \sum_{l=1}^k \beta_l x_{il} + \varepsilon_{ij}, \quad (2.50)$$

where  $y_{ij}$  is the response at the  $i$ th design point,  $x_1, x_2, \dots, x_k$  are the  $k$  factor variables,  $\beta_l$  are the unknown parameters of the linear model, and  $\varepsilon_{ij}$  is the error term at the  $l$ th design point ( $l = 1, 2, \dots, k, h = 1, 2, \dots, k - 1$ ).

If we use a full  $2^k$  factorial experiment in Step 1, then we can obtain estimators of the unknown parameters  $\beta_l$  in (2.50). We denote these estimators as  $b_l$ , (for  $l = 1, 2, \dots, k$ ). These estimators are used in Step 2 to locate the path of steepest descent, which is given by the vector  $\mathbf{d} = (-b_1, -b_2, \dots, -b_k)$ . Responses are observed along this path until there is no improvement. If curvature is evident, then a second order model is implemented. The second-order model is represented as

$$y_{ij} = \beta_0 + \sum_{l=1}^k \beta_l x_l + \sum_{h>l} \beta_{hl} x_h x_l + \sum_{l=1}^k \beta_{ll} x_l^2 + \varepsilon_{ij}, \quad (2.51)$$

where  $\beta_{hl}$  represent the coefficients of the interaction terms,  $\beta_{ll}$  represent the second-order coefficients ( $l = 1, 2, \dots, k, h = 1, 2, \dots, k - 1$ ), and all other terms are defined in (2.50). This model is used to check for the stationary point of the response surface. A canonical analysis is performed which involves among other things, locating the stationary point of the system, and determining the nature of this point.

To interpret the system more clearly, the estimated second-order response function in (2.51) is written in matrix notation as

$$\hat{y} = b_0 + \mathbf{x}'\mathbf{b} + \mathbf{x}'\mathbf{B}\mathbf{x}, \quad (2.52)$$

where,  $\hat{y}$  is the predicted response at a point,  $\mathbf{x}' = (x_1, x_2, \dots, x_k)$ , the vector  $\mathbf{b}$  is given by,  $\mathbf{b}' = (b_1, b_2, \dots, b_k)$ , and

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12}/2 & \cdot & \cdot & \cdot & b_{1k}/2 \\ b_{12}/2 & b_{22} & \cdot & \cdot & \cdot & b_{2k}/2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{1k}/2 & b_{2k}/2 & \cdot & \cdot & \cdot & b_{kk} \end{bmatrix}. \quad (2.53)$$

Note that the  $\mathbf{b}$  is the optimal (UMVU) estimator of the unknown coefficients of the metamodel (2.4) and is identical to  $\mathbf{b}$  used in Section 2.3. The stationary point for the second-order response function in (2.52) is denoted by  $\mathbf{x}_0$ , and is given by

$$\mathbf{x}_0 = -\mathbf{B}^{-1}\mathbf{b}/2 \quad (2.54)$$

Canonical analysis is the procedure of expressing the response surface in *canonical* form (see p.72 of Myers, 1976). This analysis translates the origin of the response function from  $(x_1 = 0, x_2 = 0, \dots, x_k = 0)$  to the stationary point  $\mathbf{x}_0$ . The response function is expressed in terms of new *canonical* variables  $\omega_1, \omega_2, \dots, \omega_k$ . The form of the function expressed in terms of these variables is called the *canonical* form and is of the form

$$\hat{y} = \hat{y}_0 + \theta_1\omega_1^2 + \theta_2\omega_2^2 + \dots + \theta_k\omega_k^2, \quad (2.55)$$

where  $\hat{y}_0$  is the estimated response at the stationary point,  $\mathbf{x}_0$ , and  $\theta_i$ 's are the characteristic roots, or eigenvalues of the matrix  $\mathbf{B}$ . The signs and magnitudes of these  $\theta$ 's help in determining the nature of the stationary point and the response system. If the eigenvalues of the matrix  $\mathbf{B}$  are all negative, then the stationary point found for this system is a minimum. However, if the ratio of any two eigenvalues is large, then it is an indication of a ridge system. The ratio of any two eigenvalues indicates the relative sensitivity of the two corresponding variables with respect to the response. How large

should the eigenvalue ratio be for the indication of a ridge system will be dependent on individual problems, and will be based on the experimenter's judgement and experience.

If the stationary point is found to be a minimum, the region around this point is explored and the optimum reported. If the stationary point is a saddle point, then it is an indication of a ridge system, and a ridge analysis is performed. We now briefly review ridge analysis.

In the presence of a ridge system, the ridge analysis helps the analyst to determine the best operating conditions. In this analysis, the  $k$ -variable optimization problem is reduced to a dual variable problem. Suppose the design center is assumed to have coordinates  $(0,0,\dots,0)$ , then the aim is to find a sequence of points at radii  $r = (\mathbf{x}'\mathbf{x})^{1/2}$ , from the design center which are the "best" (minimum response) design points among all other points at distances  $r$  from the design center. We only compute the "best" points along different radii from the design center which are inside the design region. In other words, the aim is to minimize the response variable defined in equation (2.52) subject to points being on specified radii. We can write it mathematically as

$$\begin{aligned} &\text{minimize } b_0 + \mathbf{x}'\mathbf{b} + \mathbf{x}'\mathbf{B}\mathbf{x} \\ &\text{subject to } r^2 = \mathbf{x}'\mathbf{x}. \end{aligned}$$

Using a Lagrange multiplier  $\mu$ , we can rewrite the above optimization problem as

$$\text{minimize } F = b_0 + \mathbf{x}'\mathbf{b} + \mathbf{x}'\mathbf{B}\mathbf{x} - \mu(\mathbf{x}'\mathbf{x} - r^2)$$

The minimum is found by differentiating  $F$  with respect to  $\mathbf{x}$ , and setting the derivative to 0, which yields the following equation that we need to solve for  $\mathbf{x}$ :

$$(\mathbf{B} - \mu\mathbf{I}_k)\mathbf{x} = \frac{-\mathbf{b}}{2}. \tag{2.56}$$

In the ridge analysis procedure, we first compute the eigenvalues of the matrix  $\mathbf{B}$ . We next choose the value of the Lagrange multiplier  $\mu$  such that it is smaller than the smallest eigenvalue of  $\mathbf{B}$ . That is, we follow the falling ridge along the "best" points on the path (a detailed explanation is provided on pp. 96-100 of Myers, 1976). The points along this ridge which do not lie inside the design region however may not be considered a display of reliable information (see p. 100 of Myers, 1976).

In general, if we observe responses at  $m$  design points, (2.50) and (2.51) can be represented in matrix form as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (2.57)$$

where  $\mathbf{y}$ ,  $\mathbf{X}$ ,  $\boldsymbol{\beta}$ , and  $\boldsymbol{\varepsilon}$  have similar interpretations as in (2.4). Also, as seen earlier, the matrix  $\mathbf{X}$  can be partitioned as  $\mathbf{X} = [\mathbf{1}_m | \mathbf{D}]$ , where  $\mathbf{D}$  is referred to as the design matrix. For example, for an optimization problem with two independent variables, the design matrix  $\mathbf{D}$ , for a full factorial experiment is represented as

$$\mathbf{D} = \begin{bmatrix} -1 & -1 \\ -1 & +1 \\ +1 & -1 \\ +1 & +1 \end{bmatrix}, \quad (2.58)$$

where elements of this matrix usually correspond to coded levels of the factor variables. For the same example, if the design used is an orthogonal central composite design (ccd), (second-order model), then  $\mathbf{D}$  will be represented as

$$\mathbf{D} = \begin{bmatrix} -1 & -1 & +1 & +1 & +1 \\ -1 & +1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & +1 \\ +1 & +1 & +1 & +1 & +1 \\ +\alpha & 0 & 0 & \alpha^2 & 0 \\ -\alpha & 0 & 0 & \alpha^2 & 0 \\ 0 & +\alpha & 0 & 0 & \alpha^2 \\ 0 & -\alpha & 0 & 0 & \alpha^2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \quad (2.59)$$

Several researchers have applied RSM to successfully solve unconstrained and constrained optimization problems.

Applications of RSM to engineering fields other than its origin in the chemical engineering processes are many. Wu (1962) applied the RSM technique to test tool lives. The RSM approach was illustrated in the improvement of traffic signal settings in a street network by Montgomery, Talvage, and Mullen (1972). Other applications include: (a) a production planning studied by Kleijnen (1987), (b) multicomputing environment studied by Biles and Ozmen (1987), and (c) stochastic networks by Bailey, Bauer, and Marsh (1989). A detailed presentation of RSM can be found in Myers (1976), Khuri and Cornell (1987), and Box and Draper (1987). An excellent review of the progress of RSM is discussed by Myers, Khuri, and Carter (1989). Rustagi (1981) discusses a wide class of optimizing techniques used in simulation experimentation which includes the technique of RSM.

All the literature that attempts to solve optimization problems using RSM use the same basic approach as outlined in the stepwise procedure above. The difference lies in the first-order and second-order designs used, or in the gradient search procedures used. For the first-order designs, more often than not, the  $2^k$  full factorial, or the  $2^{k-p}$ , fractional factorial experiments are conducted. However, there are various second-order designs employed depending on the focus of the experiment, and also on the number of design points that the experimenter can afford. A comparison of second-order designs is found in Montgomery and Evans (1972). They consider various second-order models such as: (a) the  $3^2$  factorial design, (b) a rotatable orthogonal central composite design (ccd), (c) a rotatable uniform precision ccd, (d) a rotatable minimum bias ccd, (e) a rotatable orthogonal hexagon design, and (f) a rotatable uniform precision hexagon design. These designs are briefly discussed next.

A design is said to be rotatable if the variance of the predicted response is a function of the distance from the design origin only and not of the direction. An orthogonal design provides regression coefficient estimates which are independent. A uniform-precision design requires that the variance of the predicted response, usually denoted by  $\hat{y}$ , at the design center is equal to the variance of  $\hat{y}$  at some arbitrary distance from the origin, but within the design region. Minimum-bias designs afford protection against the presence of third-order terms. The central composite designs consist of factorial ( $2^k$ ), or a fractional factorial ( $2^{k-p}$ ), portion plus an axial portion with  $2k$  axial points  $(\pm \alpha, 0, \dots, 0)$ ,  $(0, \pm \alpha, \dots, 0)$ , ...,  $(0, 0, \dots, \pm \alpha)$ , and  $n_0$  observations at the design center. The hexagon design belongs to the class of equiradial designs formed from the regular polygons with at least five vertices. The conditions for a second-order rotatable design which result in uniform precision, or, the conditions to achieve uniform precision, and

rotatable ccd's, and other properties of ccd's are discussed in Chapters 7 and 9 of Myers (1976).

## 2.5.2 RSM with Computer Simulation

In this sub-section, we give a review of the literature on applying RSM to the simulation context. Biles (1973), Biles (1975), and Biles and Swain (1979), Daughety and Turnquist (1979), Montgomery and Bettencourt (1977) discuss the RSM technique employed in the context of simulation. Their methods however do not exploit the variance reduction techniques that can be incorporated in the simulation context. In addition, their methods use nonlinear programming techniques other than the conventional method of steepest descent. Some of these methods are discussed in the next sub-section. In this sub-section we discuss the literature that deals with RSM used in conjunction with variance reduction techniques offered by simulation experiments. As seen in Section 2.2, through appropriate assignment of the random number streams to the design points in a carefully designed simulation experiment, the experimenter can purposefully induce correlations among responses that result in improved statistical inference on  $\beta$ . Cooley and Houck (1982) studied the application of the Schruben-Margolin strategy to RSM experiments for an inventory problem. Their results showed that for the first and second-order models, the use of the Schruben-Margolin strategy resulted in reduced error variances over the application of independent streams.

Hussey, Myers, and Houck (1987a) investigated the best pseudorandom number assignment in a simulation experiment for a first-order response surface model. They used factorial and fractional factorial designs, and compared three assignment strategies: (a)

independent streams, (b) CRN strategy, and (c) Schruben-Margolin strategy. They based their analysis on following variance criteria: generalized variance, prediction variance, integrated variance, and variance of slopes. Their results indicate that no one assignment is uniformly superior for all four criteria, although the Schruben-Margolin is the overall preferred design to be used. Their findings, in particular, show that for the variance of slopes criterion, the CRN strategy and the Schruben-Margolin strategy exhibit identical results. This criterion is the most important tool for the method of steepest descent. Hence the CRN strategy, which is much easier to employ seems to be the more viable one to use if the experimenter starts the RSM experiment remote from the optimum, and there is a good reason to believe that there is presence of curvature in the overall model. A similar study was performed by Hussey, Myers, and Houck (1987b) for second-order response surface models. The same four variance criteria and the same three assignment strategies were considered. The second-order designs considered were: (a) ccd and (b) Box-Behnken. Their results indicate that with priorities assigned to optimization and prediction, the Schruben-Margolin strategy has greater utility in the formulation of second-order simulation designs.

The factor of bias in the estimation of response surface models is studied by Donohue, Houck, and Myers (1992). They suggest experimental plans to be employed to protect the model against the presence of unfitted third-order terms. The RSM designs studied include: (a) ccd, (b) Box-Behnken, (c) three-level factorial, and (d) small ccDs. Each design is studied under three random number assignment strategies discussed earlier. They use the "fit-protect" experimental plan with the design criterion being the mean squared error which considers both the variance and the bias of errors associated with the estimated response. A detailed treatment of the same is found in Donohue (1988). The results indicate that the Schruben-Margolin strategy generally performs the best of the

three strategies, and the performance improves as the magnitudes of the induced correlations increase. The ccd and the Box-Behnken designs were found to perform the best of the four second-order design classes.

Incorporating variance reduction techniques in simulation-optimization algorithms has immense potential for increasing the efficiency of such algorithms. The number of simulation runs required by an algorithm can be reduced. Also, because of improvement in prediction capabilities due to reduction in variances, the predicted optimum can be obtained with more confidence. The next section discusses NLP techniques used in RSM.

### **2.5.3 NLP Techniques in RSM**

In a typical RSM study, if the system under study is being investigated for the first time, then starting conditions are often not close to the optimum conditions. Thus, the analyst needs to perform some preliminary investigation to identify the region where a second degree equation can usefully be employed. The analyst can then use a second-order experimental design to locate the optimum.

The preliminary investigation can generally be summarized as follows:

**Step 1:** Design a factorial or a fractional factorial experiment to obtain the fitted first-order model.

**Step 2:** Using the information from Step 1, observe responses along a path where improving responses are suspected to be obtained (that is, follow the path of steepest descent).

Step 3: Select the most favorable response along this path as the center of yet another design. If the first-order model for the new design seems to adequately represent the data, then go to Step 1. Otherwise, the preliminary investigation is complete.

Non-linear programming (NLP) techniques in RSM are used mostly in Step 2 of the preliminary investigation. NLP techniques can also be used after the preliminary investigation is complete. Instead of using second-order RSM designs, some quadratic programming technique can be used. NLP techniques can also be used *instead* of the preliminary investigation. Some NLP procedure can be used to get to the optimal point, and then a second-order model can be employed to improve the estimation of the optimal point. We next review the non-linear programming algorithms that have been implemented in the RSM context.

The problems studied are either: (a) single response, or (b) multiple response problems. The multiple response problems also include constrained optimization problems. This is due to the fact that in many multiple response situations, the most important response is taken as the primary response, and the others are viewed as constraints which need to take a particular value (equality constraints), or some threshold minimum or maximum values ("greater than" or "less than" constraints, respectively). For the purpose of this discussion, constrained response problems will be approached in the same manner as the multi-response problems.

We first discuss single response optimization problems. Single response optimization problems can be written in general as

$$\text{minimize } E[y] = f(\mathbf{x}) \text{ for } \mathbf{x} = (x_1, x_2, \dots, x_k)'. \quad (2.60)$$

For this class of problems, the most commonly used technique is the method of steepest descent. The method of searching one variable at a time can be used, but the path of steepest descent is usually more effective and economical. Box and Draper (pp. 194-197) discuss a single-response problem bounded by a plane. The path of steepest descent, say,  $\mathbf{d}$ , is pursued until this search "hits" the plane. They find: (a) the point at which it hits the plane, and (b) the direction of the modified descent vector once it hits the plane. That is, they project the gradient along the plane that "obstructs" the search, and continue the search along the new path.

We next discuss problems with multiple responses. Carroll (1961) offered the created response surface technique for constrained optimization problems. The problem under study was to find the settings of independent variables that maximize the effectiveness (response, or, objective function value) of the system subject to nonviolation of constraints at each stage of the optimization procedure. His method sets a barrier against leaving the region, and hence is called the barrier function method.

Myers and Carter (1973) studied the use of response surface technique to dual quadratic response surface systems. The problem is to maximize two response variables obtained from either the same experiment or externally. The approach they use is to find settings of the independent variables to maximize a quadratic "primary response" function subject to the condition that the second response, which is also quadratic, and which they call "constraint response", takes on some specified value. They used the method of Lagrange Multipliers (Lagrange Multipliers were also used earlier with response surfaces by Umland and Smith, 1959) to outline a method which can guarantee simple two dimensional plots to determine the conditions of constrained maximum primary response regardless of the number of independent variables in the system. Their method provides

a graphical display of the primary response for various values of the constraint response. The limitation of this method is that only two responses can be handled, although there could be any number of independent variables. Also, this method is restricted to second-order (quadratic) responses.

Heller and Staats (1973) used Zoutendijk's method of feasible directions to optimize the problem under study. At Step 2 of the preliminary investigation described earlier, for an unconstrained problem the gradient is computed and the path of steepest descent is then pursued for improving the response function. But for a constrained problem, Heller and Staats proposed Zoutendijk's method of feasible directions. The estimated gradient of  $f(x)$  with respect to the independent variables, and those constraints which are holding as equalities, are used to determine the direction of translation which provides the maximum rate of decrease in  $f(x)$  within the feasible region. Zoutendijk's method is described on p. 413 of Bazaraa, Sherali, and Shetty, 1993.

Biles (1975) used Rosen's gradient projection method to find conditions on a set of process design variables which optimize a primary response, subject to maintaining a set of secondary process responses within specified ranges. Biles concluded that the gradient projection method does not guarantee an optimal solution when employed in a purely computational procedure. He also notes that a comparison of the type of experimental design and the number of replicates at each design point are of considerable importance in finding the gradient-projection directions. He also suggests the need for further investigation in the placement or spacing of design points for multiple-response experimentation.

Montgomery and Bettencourt (1977) applied the Geoffrion-Dyer algorithm to the multiple response optimization problem. The approach to this problem is as follows. The

problem at hand is assumed to be a multiple response problem. The functional form of some (or maybe none) of the responses is known. RSM is used to obtain the functional form of the remaining responses near their respective optimal operating conditions. Once the functional forms of all the responses is established, the Geoffrion-Dyer vector maximal algorithm is applied to get the optimum of the utility function of the multiple responses. The interactive vector maximal algorithm is used in the context of the Frank-Wolfe method (see pp. 115-117 of Montgomery and Bettencourt, 1976, or pp. 358-361 of Geoffrion, Dyer, and Feinberg, 1972). In this approach, all solutions depend heavily on the decision-maker's approach to solving the given problem. Depending on the response surface, and the variance of the error terms, different methods are suitable. Another interactive algorithm for multiple objective optimization was suggested by Loganathan and Sherali (1987) which yields a best-compromise solution in situations with an implicitly defined utility function.

Biles and Swain (1979) studied direct search methods, first-order response surface methods, and second-order response surface methods. The direct search method studied was the complex search due to Box. They concluded that the effectiveness of simulation-optimization procedures depends on the randomness in the system and the topography of the true but unknown response surface. The first-order response surface method tends to seek a local stationary region and is inefficient in the presence of large error. The second-order surfaces on the other hand can also fail to adequately estimate optima for highly irregular surfaces. In these cases, the complex search works fairly well provided the points used at any iteration of the search are well dispersed over the experimental region. The next section introduces the notion of conjugacy and discusses quasi-Newton and conjugate gradient methods.

## 2.6 Quasi-Newton and Conjugate Gradient Methods

This section discusses the quasi-Newton and conjugate gradient direction methods which are *gradient deflection methods* and are based on the idea of conjugacy. In general, if  $\mathbf{H}$  is a  $k \times k$  symmetric matrix, then vectors  $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k$  are  $\mathbf{H}$ -conjugate if they are linearly independent, and if  $\mathbf{d}'_i \mathbf{H} \mathbf{d}_j = 0$  for  $i \neq j$ . In nonlinear optimization,  $\mathbf{H}$  usually denotes the Hessian of the objective function  $f(\mathbf{x})$  in say  $k$  variables, and  $\mathbf{d}_i$  ( $i = 1, 2, \dots, k$ ) are then called the conjugate directions. If  $f(\mathbf{x})$  is a quadratic function in  $k$  variables, then at most  $k$  steps are required to compute its minimum, provided the search is conducted along conjugate directions of  $\mathbf{H}$  (see Theorem 8.3.3 of Bazaraa, Sherali, and Shetty 1993).

This section is divided into three sub-sections. Section 2.6.1 reviews quasi-Newton methods, Section 2.6.2 discusses conjugate gradient methods and also a class of Quasi-Newton methods called *memoryless* quasi-Newton methods, and Section 2.6.3 presents an overview of some restarting criteria to enhance the performance of the conjugate gradient methods suggested in the literature.

### 2.6.1 Quasi-Newton Methods

Quasi-Newton methods use search directions which are of the form  $\mathbf{d}_j = -\mathbf{P}_j \mathbf{g}_j$  instead of using  $-\mathbf{H}_j^{-1} \mathbf{g}_j$ , as in Newton's method, where  $\mathbf{g}_j$  is the gradient of the function  $f(\mathbf{x})$  evaluated at the  $j$ th iterate,  $\mathbf{P}_j$  is a  $k \times k$  positive definite (PD) matrix that approximates the inverse of the Hessian matrix. That is, the gradient is deflected by premultiplying it

by  $-\mathbf{P}_j$  (hence these methods are called gradient deflection methods). Thus,  $\mathbf{d}_j$  is a descent direction if  $\mathbf{g}_j \neq 0$ , since then,  $\mathbf{d}'_j \mathbf{g}_j < 0$ .

Next define

$$\mathbf{p}_j = \mathbf{x}_j - \mathbf{x}_{j-1}, \quad (2.61)$$

and

$$\mathbf{q}_j = \mathbf{g}_j - \mathbf{g}_{j-1} = \mathbf{H}(\mathbf{x}_j - \mathbf{x}_{j-1}) = \mathbf{H}\mathbf{p}_j \quad (2.62)$$

Different variants of the quasi-Newton methods can be found in the literature. For example, Davidon-Fletcher-Powell used

$$\mathbf{P}_{j+1} = \mathbf{P}_j + \frac{\mathbf{p}_j \mathbf{p}'_j}{\mathbf{p}'_j \mathbf{q}_j} - \frac{\mathbf{P}_j \mathbf{q}_j \mathbf{q}'_j \mathbf{P}_j}{\mathbf{q}'_j \mathbf{P}_j \mathbf{q}_j}, \quad (2.63)$$

with  $\mathbf{P}_1$  being some PD matrix. If we let

$$\mathbf{C}_j^{DFP} = \frac{\mathbf{p}_j \mathbf{p}'_j}{\mathbf{p}'_j \mathbf{q}_j} - \frac{\mathbf{P}_j \mathbf{q}_j \mathbf{q}'_j \mathbf{P}_j}{\mathbf{q}'_j \mathbf{P}_j \mathbf{q}_j}, \quad (2.64)$$

(DFP for the authors) then, we can write (2.63) as

$$\mathbf{P}_{j+1} = \mathbf{P}_j + \mathbf{C}_j^{DFP}. \quad (2.65)$$

For quadratic functions,  $\mathbf{P}_j$  produces the exact representation of the Hessian inverse within  $k$  steps. For quasi-Newton methods, in general, we construct a matrix (see p.324 of Bazaraa, Sherali, and Shetty, 1993)

$$\mathbf{P}_{j+1} = \mathbf{P}_j + \mathbf{C}_j \quad (2.66)$$

where  $C_j$  is some symmetric matrix that ensures that  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_j$  are eigenvectors of  $\mathbf{P}_{j+1}\mathbf{H}$  with unit eigenvalues. Hence, we want  $\mathbf{P}_{j+1}\mathbf{H}\mathbf{p}_j = \mathbf{p}_j$ , or that,  $\mathbf{p}_j = \mathbf{P}_j\mathbf{q}_j + \mathbf{C}_j\mathbf{q}_j = \mathbf{P}_j\mathbf{H}\mathbf{p}_j + \mathbf{C}_j\mathbf{q}_j = \mathbf{p}_j + \mathbf{C}_j\mathbf{q}_j$ , which implies that we want

$$\mathbf{C}_j\mathbf{q}_j = 0 \text{ for } k = 1, 2, \dots, j-1. \quad (2.67)$$

$C_j$  is sometimes called a correction matrix, since in some ways it *corrects* the matrix  $\mathbf{P}_j$  to yield  $\mathbf{P}_{j+1}$ . For  $k \equiv j$ , we have the *quasi-Newton condition*

$$\mathbf{P}_{j+1}\mathbf{q}_j = \mathbf{p}_j. \quad (2.68)$$

This can be written as

$$\mathbf{C}_{j+1}\mathbf{q}_j = \mathbf{p}_j - \mathbf{P}_j\mathbf{q}_j. \quad (2.69)$$

Broyden, Fletcher, Goldfarb, and Shanno came up with a correction matrix which is referred to as the *BFGS update*, which has shown in many computational studies to dominate other updating schemes in its overall performance (see p. 326 of Bazaraa, Sherali, and Shetty).  $\mathbf{C}_j^{BFGS}$  is given by

$$\mathbf{C}_j^{BFGS} = \frac{\mathbf{p}_j\mathbf{p}'_j}{\mathbf{p}'_j\mathbf{q}_j} \left[ 1 + \frac{\mathbf{q}'_j\mathbf{P}_j\mathbf{q}_j}{\mathbf{p}'_j\mathbf{q}_j} \right] - \frac{\mathbf{P}_j\mathbf{q}_j\mathbf{p}'_j + \mathbf{p}_j\mathbf{q}'_j\mathbf{P}_j}{\mathbf{p}'_j\mathbf{q}_j}. \quad (2.70)$$

Thus, in quasi-Newton method, at each step  $j$ , we calculate the revised update  $\mathbf{P}_j$ , and follow the direction of descent given by  $-\mathbf{P}_j\mathbf{g}_j$ . Finally, within  $k$  iterations (for a  $k$ -variable quadratic minimization problem), the matrix  $\mathbf{P}_j$  (for some  $j \leq k$ ) exactly represents the Hessian inverse, and hence the descent direction is the Newton direction. In the next section we discuss the method of conjugate gradients and special class of quasi-Newton methods which are *memoryless*.

## 2.6.2 Conjugate Gradient Methods

Originally developed by Hestenes and Steifel (1952), conjugate direction methods were first applied to unconstrained minimization problems by Fletcher and Reeves (1964). These methods are devised to converge faster than the method of steepest descent and are efficient computationally in terms of storage requirements than the Newton's algorithm. They are also applied to non-quadratic problems, since smooth functions exhibit quadratic behavior in the vicinity of the optimum (see p. 360 of Sherali and Ulular, 1990). In the method of conjugate gradients, a sequence of points  $\mathbf{x}_{j+1}$ , and a sequence of directions  $\mathbf{d}_j$ , are generated iteratively as

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \lambda_j \mathbf{d}_j, \quad (2.71)$$

and

$$\mathbf{d}_j = -\mathbf{g}_j + \kappa_j \mathbf{d}_{j-1}, \quad (2.72)$$

where  $\kappa_j$  is a multiplier which scales the direction vector of the previous iteration to maintain conjugacy, and is calculated differently under different conjugate gradient methods,  $\mathbf{g}_j$  is the gradient of the objective function at the operating point  $\mathbf{x}_j$ ,  $\lambda_j$  is the step length along  $\mathbf{d}_j$  which minimizes  $f(\mathbf{x})$  along this direction. As seen from (2.72) a conjugate gradient direction at the  $k$ th iterate,  $\mathbf{d}_j$ , is formed from the linear combination of the negative gradient at  $k$ th iteration and the direction vector of the previous iterate. This implies a possible advantage of this method over the method of steepest descent. Now suppose  $f(\mathbf{x})$  is quadratic with a positive definite (PD) Hessian  $\mathbf{H}$ , and that  $\mathbf{d}_j$  and  $\mathbf{d}_{j-1}$  are  $\mathbf{H}$ -conjugate. Then we have,  $\mathbf{d}'_j \mathbf{H} \mathbf{d}_{j-1} = 0$ . From (2.63) and (2.72) we get,  $0 = \mathbf{d}'_j \mathbf{H} \mathbf{p}_j = \mathbf{d}'_j \mathbf{q}_j$ . Using this in (2.62) yields Hestenes and Stiefel's (1952) choice for

$\kappa_j$ , used even in nonquadratic situations by assuming a local quadratic behavior, as (see p. 329 of Bazaraa, Sherali, and Shetty, 1993)

$$\kappa_j^{HS} = \frac{\mathbf{g}'_j \mathbf{q}_j}{\mathbf{d}'_j \mathbf{q}_j} = \frac{\lambda_j \mathbf{g}'_j \mathbf{q}_j}{\mathbf{p}'_j \mathbf{q}_j}. \quad (2.73)$$

In a similar vein, various other choices for the multiplier  $\kappa_j$  were derived by Polak and Ribiere (1969) who obtained

$$\kappa_j^{PR} = \frac{\mathbf{g}'_j \mathbf{q}_j}{\|\mathbf{g}_j\|^2}, \quad (2.74)$$

and Fletcher and Reeves who derived

$$\kappa_j^{FR} = \frac{\|\mathbf{g}_j\|^2}{\|\mathbf{g}_{j-1}\|^2}, \quad (2.75)$$

etc. Another gradient deflection scheme where *conjugacy* is not necessarily maintained, although convergence is still guaranteed is provided by Sherali and Ulular (1990). It is called the average direction strategy (ADS), and yields the search direction on the  $j$ th iteration as

$$\mathbf{d}_j = -\mathbf{g}_j + \frac{\|\mathbf{g}_j\|}{\|\mathbf{d}_{j-1}\|} \mathbf{d}_{j-1}, \quad (2.76)$$

which essentially searches along the direction given by a weighted average of the gradient of the current iterate and the direction followed on the previous iterate.

We next discuss the relationship between conjugate gradient methods and a simplified version of the BFGS update in (2.70). Let us assume that in (2.70)  $\mathbf{P}_j = \mathbf{I}$ . Hence, we get

$$\mathbf{P}_{j+1} = \mathbf{I} + \frac{\mathbf{p}_j \mathbf{p}'_j}{\mathbf{p}'_j \mathbf{q}_j} \left[ 1 + \frac{\mathbf{q}'_j \mathbf{P}_j \mathbf{q}_j}{\mathbf{p}'_j \mathbf{q}_j} \right] - \frac{\mathbf{P}_j \mathbf{q}_j \mathbf{p}'_j + \mathbf{p}_j \mathbf{q}'_j \mathbf{P}_j}{\mathbf{p}'_j \mathbf{q}_j}. \quad (2.77)$$

The descent direction then conducts the search along

$$\mathbf{d}_{j+1} = -\mathbf{P}_{j+1} \mathbf{g}_{j+1}. \quad (2.78)$$

In this direction, the previous approximation  $\mathbf{P}_j$  is *forgotten*, and an identity matrix is used as for the first iteration of the quasi-Newton method. Hence, this method is called a *memoryless quasi-Newton method*.

If exact line searches are performed along the descent direction, then we have  $\mathbf{p}'_j \mathbf{g}_{j+1} = \lambda_j \mathbf{d}_j \mathbf{g}_{j+1} = 0$ . Thus, using (2.77) and (2.73) in (2.78) yields

$$\mathbf{d}_{j+1} = -\mathbf{g}_{j+1} + \frac{\mathbf{q}'_j \mathbf{g}_{j+1}}{\mathbf{p}'_j \mathbf{q}_j} \mathbf{p}_j = -\mathbf{g}_{j+1} + \kappa_j^{HS} \mathbf{d}_j. \quad (2.79)$$

Thus the memoryless BFGS update scheme is equivalent to the conjugate gradient method of Hestenes and Steifel (or Polak and Ribiere) when exact line searches are used.

Under the HS choice for  $\kappa_j^{HS}$ , the direction  $\mathbf{d}_j$  can be written as (see p. 1074 of Perry, 1978)

$$\mathbf{d}_j = - \left[ \mathbf{I} - \frac{\mathbf{p}_j \mathbf{q}'_j}{\mathbf{q}'_j \mathbf{p}_j} \right] \mathbf{g}_j \equiv -\mathbf{P}_j^{HS} \mathbf{g}_j. \quad (2.80)$$

Note that  $\mathbf{P}_j^{HS}$  is not symmetric and hence (2.80) is strictly speaking, not a memoryless quasi-Newton update. Also, if  $\mathbf{P}_j$  is some approximation of the Hessian inverse, the quasi-Newton condition requires that  $\mathbf{P}_j \mathbf{q}_j = \mathbf{p}_j$ , or under symmetry,

$$\mathbf{q}'_j \mathbf{P}_j = \mathbf{p}'_j. \quad (2.81)$$

Equation (2.80) yields  $\mathbf{q}'_j \mathbf{P}_j^{HS} = 0$ . Perry (1978) introduced  $\mathbf{P}_j^P$  which satisfies the quasi-Newton condition (2.81). Perry's choice for  $\kappa_j$  is given as

$$\kappa_j = \frac{\mathbf{q}'_j \mathbf{g}_j - \mathbf{p}'_j \mathbf{g}_j}{\mathbf{q}'_j \mathbf{d}_{j-1}}, \quad (2.82)$$

which yields

$$\mathbf{d}_j = - \left[ \mathbf{I} - \frac{\mathbf{p}_j \mathbf{q}'_j}{\mathbf{q}'_j \mathbf{p}_j} + \frac{\mathbf{p}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j} \right] \mathbf{g}_j \equiv - \mathbf{P}_j^P \mathbf{g}_j, \quad (2.83)$$

where

$$\mathbf{P}_j^P = \left[ \mathbf{I} - \frac{\mathbf{p}_j \mathbf{q}'_j}{\mathbf{q}'_j \mathbf{p}_j} + \frac{\mathbf{p}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j} \right].$$

Shanno (1978) noted that  $\mathbf{P}_j^P$  is not symmetric and hence the true quasi-Newton condition is not satisfied. He therefore proposed  $\mathbf{P}_j^{MBFGS}$ , given by

$$\mathbf{P}_j^{MBFGS} = \mathbf{I} - \frac{\mathbf{p}_j \mathbf{q}'_j + \mathbf{q}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j} + \left[ 1 + \frac{\mathbf{q}'_j \mathbf{q}_j}{\mathbf{q}'_j \mathbf{p}_j} \right] \frac{\mathbf{p}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j}. \quad (2.84)$$

Shanno observes that (2.84) corresponds to a memoryless BFGS update (hence the super-script MBFGS).

Perry's method is based on equating  $\mathbf{d}_j = -\mathbf{g}_j + \kappa_j \mathbf{d}_{j-1}$  to  $-\mathbf{P}_j \mathbf{g}_j$ , where  $\mathbf{P}_j$  is a non-symmetric matrix which is an approximation of the Hessian inverse. Sherali and Ulular (1990) note that if the Newton direction  $-\mathbf{H}_j^{-1} \mathbf{g}_j$  is contained in the cone spanned by  $-\mathbf{g}_j$  and  $\mathbf{d}_{j-1}$ , and is not coincident with  $-\mathbf{d}_{j-1}$ , then  $\kappa_j$  cannot alone guarantee the equality of  $\mathbf{d}_j \equiv -\mathbf{g}_j + \kappa_j \mathbf{d}_{j-1}$  and  $-\mathbf{H}_j^{-1} \mathbf{g}_j$ . This is due to the fact that under these assumptions  $\mathbf{d}_j$  and the Newton direction are only ensured to be collinear. They therefore developed a scale parameter  $s_j$  such that

$$s_j \mathbf{d}_j \equiv s_j [-\mathbf{g}_j + \kappa_j \mathbf{d}_{j-1}] = -\mathbf{H}_j^{-1} \mathbf{g}_j. \quad (2.85)$$

Solving for  $\kappa_j$ , Sherali and Ulular (1990) show that

$$\kappa_j = \frac{\mathbf{q}'_j \mathbf{g}_j - (1/s_j) \mathbf{p}'_j \mathbf{g}_j}{\mathbf{q}'_j \mathbf{d}_{j-1}}. \quad (2.86)$$

They further recommend the value of  $s_j$  to be  $\lambda_{j-1}$  which yields computationally better results than other choices considered for  $s_j$ . This scaled version of the quasi-Newton update for  $\kappa_j$  will be referred to as (P-SU).

We next discuss the scaled version of the MBFGS developed by Sherali and Ulular (1990). Using the (P-SU) choice for  $\kappa_j$  in (2.86), they obtain

$$\mathbf{d}_j = -\mathbf{P}_j^{P-SU} \mathbf{g}_j, \quad (2.87)$$

where

$$\mathbf{P}_j^{P-SU} = \mathbf{I} - \frac{\mathbf{p}_j \mathbf{q}'_j}{\mathbf{q}'_j \mathbf{p}_j} + \frac{\mathbf{p}_j \mathbf{p}'_j}{s_j \mathbf{q}'_j \mathbf{p}_j}. \quad (2.88)$$

Using Shanno's (1978) approach they further derive a modified BFGS update where they define

$$\mathbf{P}_j^{MBFGS-SU} = \mathbf{I} - \frac{\mathbf{p}_j \mathbf{q}'_j + \mathbf{q}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j} + \left[ \frac{1}{s_j} + \frac{\mathbf{q}'_j \mathbf{q}_j}{\mathbf{q}'_j \mathbf{p}_j} \right] \frac{\mathbf{p}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j}. \quad (2.89)$$

Note that when  $s_j = 1$ , (2.89) corresponds to the MBFGS update.

Conjugate gradient methods thus provide improvement over the steepest descent algorithms at a modest increase in storage requirements, and are therefore suited for large-scale applications. In this section we have reviewed some of the work related to the quasi-Newton, memoryless quasi-Newton, and conjugate gradient methods. The performance of these methods can be improved if proper restarting criteria are employed.

### 2.6.3 Restarting Criteria

Beale (1972) and Powell (1977) have offered various restarting criteria which improve the performance of the gradient deflection methods. In the context of RSM, restarting at any iteration would imply following the path of steepest descent at that iteration. This is not necessarily true in classical use of gradient deflection methods (not in the context of RSM). Since the focus of this study is on RSM, we only consider the ways of employing restarting in this context.

One restarting criterion, as used in the Fletcher and Reeves (1964) conjugate gradient method, is restarting every  $k$  iterations for a  $k$ -variable optimization problem. Since all directions in (2.72) are H-conjugate, Beale (1972) noted that there is no mathematical justification for using (2.72) for more than  $k$  iterations. Powell (1977) offered restarting

criteria which are additionally based on (a) checking of orthogonality is lost between  $\mathbf{g}_j$  and  $\mathbf{g}_{j+1}$ , and (b) checking if  $\mathbf{d}_j$  offers a sufficient descent, for every  $j$ th iteration. Powell's first condition mathematically implies to check if

$$\|\mathbf{g}_j' \mathbf{g}_{j+1}\| \geq \delta \|\mathbf{g}_j\|^2, \quad (2.90)$$

for  $\delta \in (0,1)$ . For definiteness the value 0.2 is suggested by Powell. The optimization is restarted if (2.90) holds. The test for sufficient descent along  $\mathbf{d}_j$  at the  $j$ th iteration is mathematically conducted as,

$$-1.2\|\mathbf{g}_j\|^2 \leq \mathbf{d}_j' \mathbf{g}_j \leq -0.8\|\mathbf{g}_j\|^2. \quad (2.91)$$

That is, restart the optimization procedure if (2.91) is violated. In conclusion, restart the optimization procedure for a  $k$ -variable problem at the  $j$ th iteration, that is set  $\mathbf{d}_j = -\mathbf{g}_j$ , if  $j = k$ , or if (2.90) holds, or if (2.91) is violated.

This chapter provides the necessary theoretical and notational framework to lead us towards (a) the discussion of an RSM algorithm under the CRN correlation-induction strategy and presentation of analytical results therein, which is the theme of the next chapter, and (b) a novel RSM algorithm which uses conjugate gradient and quasi-Newton methods instead of using the method of steepest descent *only*.

# CHAPTER III RSM Algorithm Under the CRN

## Strategy

The focus of this chapter is to develop analytical results quantifying the gains of the CRN strategy over direct simulation (IS strategy) at each stage of the RSM algorithm. Also, statistical analysis under the CRN strategy for the second-order model will be developed.

This chapter is organized as follows. Section 3.1 presents the RSM algorithm as it would be applied to the simulation-optimization problem. Also, all steps in the algorithm are explained in detail. Sections 3.2, 3.3, and 3.4, respectively, discuss the fitting of a first-order model, the gradient search procedure, and the fitting of a second-order model. Also included in these three sections are methodologies for implementing CRN, and theoretical results indicating the variance reduction of the estimates of statistics of interest using CRN over direct simulation. In Section 3.4 we also develop statistical procedures under the CRN strategy for the second-order model. Section 3.5 attempts to answer questions regarding the advantage of using CRN strategy over direct simulation while conducting the ridge analysis.

### ***3.1. RSM algorithm and discussion***

In the presentation of this algorithm we assume that the simulation analyst has no prior knowledge concerning the location of the optimum and that the set of  $k$  design variables

have lower and upper bounds which are set initially; that is, the design space for the  $k$  design variables is bounded. The problem to be solved is assumed to be an otherwise unconstrained minimization problem with only one response variable of interest.

The nine steps that comprise the RSM algorithm used for the simulation-optimization procedure are given below and closely follow those given by Myers (p. 88, 1976).

Step 1:

Select a starting point for the algorithm.

Step 2:

Construct a  $2^k$  full factorial design with 1 center point. For the first pass of the algorithm, the starting point selected in Step 1 is used as the center point for this design. Conduct  $r$  independent simulation runs at each of the  $2^k$  design points to obtain the responses  $y_{ij}$  ( $i = 1, 2, \dots, 2^k$ , and  $j = 1, 2, \dots, r$ ). Estimate the unknown regression coefficients for the first-order model using the ordinary least squares method. Note that due to the structure of the variance-covariance matrix of the responses in (2.29), we obtain sufficient conditions for equivalence of ordinary least squares (OLS) and weighted least squares (WLS) estimates (see p. 512 of Schruben and Margolin, 1978). Test for the lack-of-fit for the first-order model. That is, test,

$$\begin{aligned}
 H_0 : E[y] &= \beta_0 + \sum_{l=1}^k \beta_l x_l + \sum_{l>h} \beta_{lh} x_l x_h \\
 \text{versus} & \\
 H_1 : E[y] &\neq \beta_0 + \sum_{l=1}^k \beta_l x_l + \sum_{l>h} \beta_{lh} x_l x_h
 \end{aligned}
 \tag{3.1}$$

The lack-of-fit test based on the sums of squares is the usual  $F$ -test which is

$$F = \frac{SS_{LOF}/d_{LOF}}{SS_{PE}/d_{PE}},$$

where  $SS_{LOF}$  and  $d_{LOF}$  are the lack-of-fit sums of squares, and the corresponding degrees of freedom, respectively; and  $SS_{PE}$  and  $d_{PE}$  are the pure error sums of squares, and the corresponding degrees of freedom, respectively.

We reject  $H_0$  in (3.1) if

$$F > F_{d_{LOF}, d_{PE}}^{\alpha}$$

If the lack-of-fit test statistic is not significant at the  $\alpha = 0.1$  level, then go to Step 3. Otherwise, go to Step 4.

Step 3:

Determine the path of steepest descent given by  $\mathbf{d} = (-b_1, -b_2, \dots, -b_k)$ . Follow this path using a step length  $\lambda$  until there is no further improvement in the response values. Choose  $\lambda$  at each step such that none of the  $k$  variables  $x_1, x_2, \dots, x_k$  violate their higher or lower bounds. For any given step length  $\lambda$ , none of the decision variables can change by more than 1.5 times the design width along its axis (see discussion). If any one of the variables reaches its boundary, then fix that variable at its boundary, and continue the search along the other variables. If all variables reach their respective boundaries, and if the response is still improving (decreasing), then accept that extreme point (the most favorable point) as the optimum and go to Step 9. Otherwise, choose the design point with the best mean response as the center of a new design, and go to Step 2.

If responses along this direction are not improving, then select a design point at step length of  $\lambda/2$  from the current design center. If the mean response at this design point is still not an improvement, then accept the design point in the current design with the

most favorable mean response as the optimum. If the response improves, then use this design point as the center of a first-order design and go to Step 2.

Step 4:

Construct a central composite design (ccd) by augmenting the first-order design of Step 2 with axial points at distances  $\pm \alpha$  from the center point, with  $\alpha = 2^{k/4}$ . Test for the lack-of-fit for the second-order model. That is, test,

$$\begin{aligned}
 H_0 : E[y] &= \beta_0 + \sum_{l=1}^k \beta_l x_l + \sum_{l>h} \beta_{lh} x_l x_h + \sum_{l=1}^k \beta_{ll} x_l^2 \\
 \text{versus} & \\
 H_1 : E[y] &\neq \beta_0 + \sum_{l=1}^k \beta_l x_l + \sum_{l>h} \beta_{lh} x_l x_h + \sum_{l=1}^k \beta_{ll} x_l^2
 \end{aligned} \tag{3.2}$$

As before, the lack-of-fit test based on the sums of squares is the usual  $F$ -test with statistic

$$F = \frac{SS_{LOF}/d_{LOF}}{SS_{PE}/d_{PE}},$$

where  $SS_{LOF}$  and  $d_{LOF}$  are the lack-of-fit sums of squares, and the corresponding degrees of freedom, respectively; and  $SS_{PE}$  and  $d_{PE}$  are the pure error sums of squares, and the corresponding degrees of freedom, respectively.

Thus, we reject  $H_0$  in (3.2) at level  $\alpha$  if

$$F > F_{\alpha, d_{LOF}, d_{PE}}^*$$

If there is no lack-of-fit significant at  $\alpha = 0.1$  level, then go to Step 6. Otherwise, go to Step 5.

Step 5:

Expand the design space used in Step 2 by a suitable factor (see discussion). Go to Step 2.

Step 6:

Perform a canonical analysis and find the stationary point. If the stationary point does not exist, then return to Step 3. If the stationary point exists and is a minimum, then check the relative sensitivity of the response for each pair of the  $k$  decision variables. If the relative sensitivity between any two variables is greater than 6.0, then go to Step 7. Otherwise, go to Step 8. If the stationary point is a saddle point then go to Step 7. If the stationary point is a maximum, then return to Step 3.

Step 7:

Determine the path of steepest descent given by  $\mathbf{d} = (-b_1, -b_2, \dots, -b_k)$ . Follow this path using a step length  $\lambda$  until there is no further improvement in the response values. Accept the most favorable point along this path as the optimum. Go to Step 9. Choose  $\lambda$  at each step such that none of the  $k$  variables  $x_1, x_2, \dots, x_k$  violate their bounds. For any given step length  $\lambda$ , no decision variable is to change by more than 0.75 times the design width along its axis. If along this direction, any one of the variables reaches its boundary, then fix that variable at its respective boundary value and continue the search along the other variables. If all variables reach their upper or lower bounds, and if the response is still improving, then accept that extreme point as the optimum. Go to Step 9.

If the responses along the path of steepest descent  $\mathbf{d}$ , are not improving, then select a design point at a step length of  $\lambda/2$  from the current design center and observe the mean response at this design point. Accept the design point with the most favorable mean response as the optimum. Go to Step 9.

Step 8:

If the stationary point (found to be a minimum) lies inside the design region, then observe the response at that point, and select the optimum location as that design point in the current design which yields the most favorable response and go to Step 9. If the stationary point lies outside the design region, then go to Step 7.

Step 9:

Stop the algorithm.

We next discuss each step of the above algorithm in detail. In Step 1 we choose starting points that are selected randomly. In most real world situations, the selection of a starting point will almost never be random due to at least some prior knowledge of the process. However, for this study we use randomly selected starting points assuming no prior knowledge of the behavior of the system.

In Step 2 we choose a  $2^k$  factorial design with a single center run since we do not have any prior knowledge of the true optimum of the system under study. This design allows: (a) the first-order model to be fitted efficiently, (b) checks to be made to determine whether the first-order model is adequate, and (c) provides an estimate of the experimental error. Using the data from the  $2^k$  factorial experiment, we can calculate the estimates of  $\beta_i$ , and  $\beta_{ij}$  ( $i = 1, 2, \dots, k, j = 1, 2, \dots, k - 1, i > j$ ). The  $2^k$  factorial design is also the minimum variance design for estimating the unknown coefficients of the first-order model.

The generalized least-squares estimates of  $\beta_i$ , and  $\beta_{ij}$ , are respectively denoted by  $b_i$ , and  $b_{ij}$  ( $i = 1, 2, \dots, k, j = 1, 2, \dots, k - 1, i > j$ ).

In Step 3 we follow the path of steepest descent which is the gradient direction given by  $\mathbf{d} = (-b_1, -b_2, \dots, -b_k) = -\nabla f(\mathbf{x}_B)$ , where  $f(\mathbf{x}_B)$  is the value of the objective function at the design point  $\mathbf{x}_B$ , and  $\nabla f(\mathbf{x}_B)$  is the gradient of the objective function  $f(\mathbf{x})$  at the point  $\mathbf{x}_B$  whose true functional form is unknown. Let  $\mathbf{x}_B$  be the current base point, that is, the current design center. Then a point along the gradient direction, say,  $\mathbf{x}_g$  is calculated as  $\mathbf{x}_g = \mathbf{x}_B + \lambda(-\nabla f(\mathbf{x}_B))$ , where  $\lambda$  is the step length. In the algorithm, we do not let any variable violate its boundaries. To clarify the length of a step taken during the gradient search, consider the following example. Suppose the variable  $x_1$  has its uncoded values set to -2 and +2, then the width of the design along the  $x_1$ -axis is 4. In such a case, in one step along the gradient search, the step length  $\lambda$ , should be chosen such that the variable  $x_1$  does not change by more than 6 units (which is 1.5 times the uncoded design width along  $x_1$ ). If the responses along this direction are improving, then follow it by amounts specified by the step length given above until the responses cease to improve.

In Step 4 we fit a second-order design since the first-order fitted model cannot represent the response surface in this local region adequately. The second-order design chosen is a ccd which can be formed easily by augmenting the existing first-order model by axial design points. The reason for the choice of the distance of axial points being at  $\pm 2^{1/4}$  from the design center is to achieve a rotatable ccd. Obtain the estimates of the regression coefficients for the second-order model. That is, compute  $b_i, b_{ij}, b_{ii}(i = 1, 2, \dots, k, j = 1, 2, \dots, k - 1)$ , which are the estimates of the first-order, cross product, and second-order, unknown coefficients of the regression model, respectively.

In Step 5, expand (or contract) the design appropriately. For example, suppose at Step 2 of this algorithm, if the variable  $x_2$  has uncoded high and low levels at +2 and -2, respectively, then, in the expanded design its uncoded levels would be set at, say, +4 and -4 for high and low levels, respectively. These levels would depend upon the experimenter's judgement and experience in dealing with this problem. The levels of all other variables are selected in the expanded design likewise. The rationale for design expansion (or contraction) is as follows. At the levels currently set for the variables, neither the first-order, nor the second-order model can adequately represent the relation between the independent variables and the response. We therefore hope that exploring the design space over an expanded (or contracted) area would give us a better idea of the relation between independent variables and the response. Note that even if the linear first-order model with this expanded design is not a good fit, we still go to Step 3 and follow the path of steepest descent because we have no other alternative but to make an exploratory search in the quest for better responses.

The algorithm proceeds to Step 6 when curvature is evident. At this point, a more elaborate experiment is conducted, complete with canonical analysis which is described in detail in Section 2.5.1. If all the eigenvalues of the matrix  $\mathbf{B}$ , given in equation (2.53) are negative, then the stationary point found for this system is a minimum. However, if the ratio of any two eigenvalues is large, then it is an indication of a ridge system. Recall from Section 2.5.1 that the ratio of any two eigenvalues indicates the relative sensitivity of the two corresponding variables with respect to the response. How large should the eigenvalue ratio be for the indication of a ridge system will be dependent on individual problems, and will be based on the experimenter's judgement and experience. The yardstick set for this algorithm for the eigenvalue ratios, that is, the relative sensitivities between any pair of variables, is 6. We believe that this is a conservative value.

At the beginning of Step 7 in the algorithm, we have the following situation. The second-order model is a good fit. Also, from the canonical analysis we have that the stationary point is either a saddle point or the stationary point is a minimum, but the ratio of two eigenvalues is greater than 6, which is an indication of the existence of a ridge system. The only approach at this point is to move along the ridge with improving responses. The path of steepest descent would give a first-order approximation to track the falling (improving, for a minimization problem) ridge. A more conventional approach at this point is to perform a ridge analysis. But ridge analysis does not display reliable information outside the design region (see p.102 of Myers, 1976), and hence we follow the path of steepest descent. However, the step lengths taken are half the size of those taken for Step 3. The reason for this is that at this stage of our search we suspect the presence of some curvature (that is, we may be near the optimal point). Taking smaller step lengths will increase the probability of tracking the point of inflection rather than taking larger step lengths.

At Step 8, if the stationary point which is found to be a minimum lies inside the design region, then we explore the region around it and report the optimum. This is due to the confidence in the second-order design used, which is a rotatable ccd having good prediction capabilities inside the design region. However, if the predicted minimum lies outside the design region, then the prediction capabilities of this design are not dependable, and hence we do not draw conclusions about optimality of this stationary point or any other point in its vicinity. We next observe the response value at this stationary point. If the response at this point is better than any of the responses in the design, then we follow the path of steepest descent from this point and explore the response surface until we get no further improvement in responses. We then report the most favorable point as the optimum, and stop. Otherwise, we go to Step 7 which explores the response

surface along the path of steepest descent from the center of the current design, and reports the most favorable point as the optimum.

An important consideration before starting any RSM algorithm is deciding on the size of each design to be used at each stage. The size of the design affects the step lengths selected at Step 3 of the algorithm which could alter further decisions and results. Also, in determining the nature of the stationary point during canonical analysis, the size of the second-order design has serious implications on the decisions made there after. This issue has not been addressed adequately in the literature. Some suggestions based on some aspects of the response surface, like tolerances on the response, etc. could aid the practitioner of RSM decide on the size of his design. In this study, the design size has been fixed after conducting some pilot experiments.

Theoretical results quantifying the gains of using the CRN strategy over direct simulation are the topics of Sections 3.2 through 3.5 and are based on the algorithm presented above.

### ***3.2. The First-Order Model under CRN***

In this Section we summarize the results reviewed in Section 2.4 for conducting statistical analysis for the first-order model under the CRN strategy. We then present theoretical results leading to the expected reduction in variance in the estimates of the unknown parameters in the first-order model under the CRN strategy over the independent streams (IS) case. We further provide results pertaining to the joint confidence ellipsoids on the estimates of the unknown parameters of the first-order model. We show that under the CRN strategy, this ellipsoid has a lower volume than that under the IS strategy, thus illustrating the superiority of using CRN over the IS strategy for estimation of the unknown coefficients.

Under the CRN strategy, the same set of random number streams  $\mathbf{R}_{ij}$  ( $i = 1, 2, \dots, m; j = 1, 2, \dots, r$ ) is applied to all  $m$  design points in the  $j$ th replication. Hence, in Step 2 of the RSM algorithm, the same random number streams are used to drive the simulations at all design points of the first-order design (that is, the center point and the  $2^k$  factorial points).

Recall from Section 2.4 that since the responses under the CRN strategy are correlated, the theory of general linear models with diagonal covariance structures does not apply for the first-order metamodel given by (2.50). Thus, the need for the development of a statistical analysis under this strategy. Also recall that the covariance matrix between observations for the CRN strategy,  $\Sigma^{(CRM)}$ , is given by equation (2.29). To obtain uncorrelated observations, we have to apply the orthogonal transformation  $\Gamma^{(CRM)}$ , given by (2.30).

From this, we obtain the four results which yield

- optimal (UMVU) estimator of  $\beta$ ,
- optimal (UMP and invariant) tests on  $\beta$ .
- a confidence interval for  $\beta_0$ ,
- confidence intervals for  $\beta_1$ , and
- joint confidence intervals for  $\beta_0$  and  $\beta_1$ .

Note that the optimal estimator of  $\beta$  is used in Step 3 of the RSM algorithm to determine the path of steepest descent.

Using the orthogonal transformation,  $\Gamma^{(CRM)}$ , yields the covariance matrix for the transformed responses,  $\Sigma$ , given in (2.33). Now,  $\lambda_2^2$  is the variance of the estimators of the

components of  $\beta_1$ ; that is, variance of estimates of each  $\beta_i$  ( $i = 1, 2, \dots, k$ ) under the CRN strategy. For direct simulation (independent strategy), the variance on  $b_i$  ( $i = 1, 2, \dots, k$ ) is  $\sigma^2$ . The  $b_i$ s are used to determine the path of steepest descent. We notice that the variance of each component of the estimator of  $\beta_1$  is reduced by a factor  $(1 - \rho_+)$  under the CRN strategy when compared to the variance under independent streams, which is  $\sigma^2$ . We therefore conclude that the percentage reduction in variance of the  $b_i$  ( $i = 1, 2, \dots, k$ ) which are used to determine the path of steepest descent is  $100\rho_+$  along each coordinate axis for the independent variables. Thus, if the magnitude of the induced correlation between responses is high, then we have a significant improvement in terms of lower variances for the estimators of the components of the search direction.

We next obtain results on joint confidence ellipsoids on  $\beta$ . We know that  $\hat{\beta}_1$ , the UMVU estimate of  $\beta_1$ , is normally distributed:

$$\hat{\beta}_1 \sim N_k(\beta_1, \lambda_2^2 \mathbf{I}_k), \quad (3.3)$$

A joint  $100(1 - \alpha)\%$  confidence region for *all* the coefficients  $\beta_1$  is given by (see p. 94 of Draper and Smith, 1981)

$$(\beta_1 - \hat{\beta}_1)' \lambda_2^2 \mathbf{I}_{(k-1)} (\beta_1 - \hat{\beta}_1) \leq ks^2 F_{p, v}^\alpha \quad (3.4)$$

where  $s^2$  is an unbiased estimate of  $\sigma^2$ ,  $F_{p, v}^\alpha$  is the  $1 - \alpha$  critical point of the  $F(p, v)$  distribution. The inequality above provides the equation of an elliptically shaped contour in a space which has as many dimensions,  $k$ , as there are parameters in  $\beta_1$  (see p. 94 of Draper and Smith, 1981). Elongation along any of the  $k$  axes of the ellipsoid is directly proportional to the relative variances of the components of  $\hat{\beta}_1$ . Large variances on the components of  $\hat{\beta}_1$  increase the volume of the joint confidence ellipsoid, which in turn

reflects a deterioration in the joint estimation of the components of  $\hat{\beta}_1$ . From (3.3) we see that the variance of components of  $\hat{\beta}_1$  are equal and uncorrelated. Hence, the ellipsoid reduces to a spheroid. Let  $V^{(IS)}$  denote the volume of the spheroid under the IS strategy, and let  $V^{(CRM)}$  denote the volume of the spheroid under the CRM strategy. Now, the radius of such a spheroid is equal to the variance of each component of  $\hat{\beta}_1$ . Thus for the IS strategy, the radius of the joint confidence spheroid is  $\sigma^2$ , and it is  $\sigma^2(1 - \rho_+)$  for the CRM strategy. If we denote the radius of the spheroid under the IS strategy by  $a^{(IS)}$ , and the radius of the spheroid under the CRM strategy by  $a^{(CRM)}$ , then,

$$a^{(IS)} = \sigma^2, \quad (3.5)$$

and

$$a^{(CRM)} = \sigma^2(1 - \rho_+) \quad (3.6)$$

where  $\rho_+$  is the induced correlation, such that  $0 \leq \rho_+ \leq 1$ . The volume of any spheroid in  $k$  dimensions is proportional to  $a^k$ , if  $a$  denotes the radius of that spheroid. Thus we see that the volumes of the spheroids under the IS and CRM strategies for  $k$  even, are respectively (see p. 136 of Sommerville, 1958),

$$V^{(IS)} = \frac{\pi^{k/2} (a^{(IS)})^k}{(k/2)!} = \frac{\pi^{k/2} \sigma^{2k}}{(k/2)!}, \quad (3.7)$$

and

$$V^{(CRM)} = \frac{\pi^{k/2} (a^{(CRM)})^k}{(k/2)!} = \frac{\pi^{k/2} \sigma^{2k} (1 - \rho_+)^k}{(k/2)!}. \quad (3.8)$$

For odd values of  $k$ , (3.7) and (3.8) are given by

$$V^{(IS)} = \frac{\pi^{\frac{k-1}{2}} \left(\frac{k-1}{2}\right)! (2a^{(IS)})^k}{(k!)} = \frac{\pi^{\frac{k-1}{2}} \left(\frac{k-1}{2}\right)! 2\sigma^{2k}}{(k!)} , \quad (3.9)$$

and

$$V^{(CRN)} = \frac{\pi^{\frac{k-1}{2}} \left(\frac{k-1}{2}\right)! (2a^{(CRN)})^k}{(k!)} = \frac{\pi^{\frac{k-1}{2}} \left(\frac{k-1}{2}\right)! 2\sigma^{2k} (1 - \rho_+)^k}{(k!)} . \quad (3.10)$$

Thus, we see that

$$V^{(CRN)} = V^{(IS)} (1 - \rho_+)^k , \quad (3.11)$$

which illustrates the uniform superiority of the CRN strategy over the IS strategy in evaluation of joint confidence regions of  $\hat{\beta}_1$ . It also is evident from (3.11) that as the number of independent variables increase, the effectiveness of the CRN strategy increases exponentially.

### 3.3 Gradient Search Procedure

In this section we discuss the application of the CRN strategy to the gradient search procedure, and develop an expression for gain achieved using this strategy instead of the IS strategy. The power of the tests for testing the hypothesis of stopping at the correct step along the gradient under IS and CRN strategies are compared.

Recall from Section 3.1 that the gradient search procedure is used at Step 3 of the RSM algorithm. As we perform the gradient search procedure, that is, as we follow the path of steepest descent, the statistic of interest is the difference between the responses along the path. If we denote  $\bar{y}_0$ , as the mean response at the center of the current design, and  $\bar{y}_j$  ( $j = 1, 2, \dots, w$ ) as the  $w$  mean responses along the gradient search path, then our

statistics of interest are  $\bar{y}_j - \bar{y}_{j-1}$  ( $j = 1, 2, \dots, w$ ). Under the CRN scheme, we apply the same random number streams to drive the stochastic components to obtain simulation responses  $\bar{y}_j$  ( $j = 0, 1, \dots, w$ ). By the first two assumptions in Section 2.4, we have positive correlations  $\rho_+$  induced among all these responses.

Under the independent strategy case, we have ( $j = 1, 2, \dots, w$ ):

$$\text{var}(\bar{y}_j - \bar{y}_{j-1}) = \text{var}(\bar{y}_j) + \text{var}(\bar{y}_{j-1}) = \frac{2\sigma^2}{r}, \quad (3.12)$$

and under the CRN strategy,

$$\text{var}(\bar{y}_j - \bar{y}_{j-1}) = \text{var}(\bar{y}_j) + \text{var}(\bar{y}_{j-1}) - 2\text{cov}(\bar{y}_j, \bar{y}_{j-1}) = \frac{2\sigma^2(1 - \rho_+)}{r}, \quad (3.13)$$

where  $r$  is the number of replications performed at each design point, and  $\sigma^2$  is the homogeneous variance of the response at each design point.

From (3.12) and (3.13) we see that as  $\rho_+ \rightarrow 1$ , the variance on the statistic of interest in the gradient search procedure goes to zero. That is, if we can induce high covariances (or correlations) between responses, then we have a larger reduction in variance of the statistics of interest under the gradient search method. The expected reduction in variance being the magnitude of the induced correlation.

The advantage of using CRN for the gradient search can be characterized in another way. In the gradient search, assume that once the step length  $\lambda$  is fixed, we need to take *exactly*  $w$  steps before the response starts increasing. That is, after fixing the step length, we should get responses to improve (decrease) exactly until the  $(w - 1)$ st step, and on the  $w$ th step, the response should increase forcing us to stop the gradient search procedure. However, due to natural variation in  $\bar{y}$ , the variance in the mean responses may be so

large that this could result in the search taking something other than *exactly*  $w$  steps along the gradient. We would like to increase the probability that the experimenter stops after *exactly*  $w$  steps.

In other words, the gradient search procedure is similar to performing sequential tests of the form ( $j = 1, 2, \dots, w$ ):

$$\mathbf{H}_0 : \bar{y}_j - \bar{y}_{j-1} \leq 0, \text{ vs, } \mathbf{H}_1 : \bar{y}_j - \bar{y}_{j-1} > 0.$$

If the search is performed accurately, then the experimenter should fail to reject the first  $w - 1$  tests, and reject the  $w$ th test. For this purpose, let us first define  $d_i = \bar{y}_i - \bar{y}_{i-1}$  ( $i = 1, 2, \dots, w - 1$ ), and  $d_w = \bar{y}_{w-1} - \bar{y}_w$ . Maximizing the probability that the experimenter stops after exactly  $w$  steps, is therefore equivalent to the probability that

$$Pr(d_1 > 0, d_2 > 0, \dots, d_w > 0). \quad (3.14)$$

We now prove that the above probability is higher under the CRN strategy than under the IS strategy. Proving this would also indicate that the power of the test under the CRN strategy is greater than that under the IS strategy for the following hypothesis test:

$$\begin{aligned} &\mathbf{H}_0 : d_1 > 0, d_2 > 0, \dots, d_w > 0 \\ &\text{versus} \\ &\mathbf{H}_1 : \text{Any above condition violated.} \end{aligned} \quad (3.15)$$

We know that the  $d_i$ 's are not independent, and so the tests on the  $d_i$ 's are also not independent. Now,

$$var(d_i) = var(\bar{y}_i - \bar{y}_{i-1}) = var(\bar{y}_i) + var(\bar{y}_{i-1}) - 2cov(\bar{y}_i, \bar{y}_{i-1}). \quad (3.16)$$

We assume under the IS strategy,  $cov(\bar{y}_i, \bar{y}_{i-1}) = 0$ , and that under the CRN strategy,  $cov(\bar{y}_i, \bar{y}_{i-1}) = \rho_+ \sigma^2$ . Thus, we get for the IS strategy,

$$var(d_i) = \frac{2\sigma^2}{r}, \quad (3.17)$$

and for the CRN strategy,

$$var(d_i) = \frac{2\sigma^2(1 - \rho_+)}{r}. \quad (3.18)$$

Also (see p. 11 of Seber, 1977), under the IS strategy,

$$cov(d_i, d_{i-1}) = cov((\bar{y}_i - \bar{y}_{i-1}), (\bar{y}_{i+1} - \bar{y}_i)) = \frac{-\sigma^2}{r}, \quad (3.19)$$

and under the CRN strategy,

$$cov(d_i, d_{i-1}) = cov((\bar{y}_i - \bar{y}_{i-1}), (\bar{y}_{i+1} - \bar{y}_i)) = \frac{\sigma^2(1 - \rho_+)}{r}. \quad (3.20)$$

The variance-covariance matrix of  $d$ 's is thus a tridiagonal matrix denoted by  $\Phi$ . Note that the structure of this matrix is identical under the two simulation strategies, and is of the form

$$\Phi = \begin{bmatrix} \gamma & \delta & \cdot & \cdot & 0 & 0 \\ \delta & \gamma & \cdot & \cdot & 0 & 0 \\ 0 & \delta & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \gamma & \delta \\ 0 & 0 & \cdot & \cdot & \delta & \gamma \end{bmatrix}, \quad (3.21)$$

where  $\gamma = 2\sigma^2/r$  under the IS strategy, and  $\gamma = 2\sigma^2(1 - \rho_+)/r$  under the CRN strategy. Also,  $\delta = -\sigma^2/r$  under the IS strategy, and  $\delta = -\sigma^2(1 - \rho_+)/r$  under the CRN strategy.

To obtain independent tests on  $d_i$ 's, we use an orthogonal transformation denoted by  $\mathbf{M}$ , such that

$$\Phi^* = \mathbf{M}\Phi\mathbf{M}' \quad (3.22)$$

is a diagonal matrix. Note that  $\mathbf{M}$  is an orthogonal matrix whose columns comprise of the eigenvectors of  $\Phi$ . The transformation  $\mathbf{M}$  will be different under IS and CRN strategies, but will have the same matrix *structure*. Thus, if we transform  $\mathbf{d}' = (d_1, d_2, \dots, d_w)$ , to  $\mathbf{d}^*$ , such that

$$\mathbf{d}^* = \mathbf{M}\mathbf{d}, \quad (3.23)$$

and let  $\mathbf{d}^{*'} = (d_1^*, d_2^*, \dots, d_w^*)$ , then the tests performed on  $d_i^*$  are independent. The goal then becomes to maximize the following probability:

$$Pr(d_1^* > 0, d_2^* > 0, \dots, d_w^* > 0) = Pr(d_1^* > 0) Pr(d_2^* > 0) \dots Pr(d_w^* > 0). \quad (3.24)$$

Since  $\mathbf{d}^* = \mathbf{M}\mathbf{d}$ , (3.24) becomes

$$Pr(\mathbf{M}\mathbf{d} > \mathbf{0}), \quad (3.25)$$

which is equivalent to

$$Pr(d_1 > 0) Pr(d_2 > 0) \dots Pr(d_w > 0). \quad (3.26)$$

Using  $\phi$  as the standard normal cumulative distribution function, we can standardize the individual  $d_i$ 's and obtain probabilities as (see p. 931 of Kreyszig) ( $i = 1, 2, \dots, w$ ):

$$Pr(d_i > 0) = \phi\left(\frac{\bar{d}_i}{\sigma_d/w}\right), \quad (3.27)$$

where  $\bar{d}_i$  is the mean  $d_i$  ( $i = 1, 2, \dots, w$ ),  $\sigma_d^2$  is the homogeneous variance of all  $d_i$ , and hence the homogeneous variance of all  $\bar{d}_i$ 's, is  $\sigma_d^2/w$ . The goal in (3.24) is now equivalent to minimizing  $\sigma_d$ . Equations (3.17) and (3.18) give the expressions for  $\sigma_d^2$  under the IS and CRN strategy, respectively. We notice that under the CRN strategy,  $\sigma_d$  is less than that under IS strategy by a factor of  $\sqrt{1 - \rho_+}$ , thus showing the superiority of the CRN strategy over the IS strategy.

We notice that the CRN strategy performs better than the IS strategy especially for values of  $\bar{d}_i$  close to zero, for which we need more sensitive tests, since these are the values at which large variance in  $d_i$ 's can result in wrong conclusions along the gradient search.

In this section we have illustrated the superiority of using the CRN strategy over the IS strategy for the gradient search procedure. Sections 3.2 and 3.3 have thus showed the gains under the CRN strategy over the IS strategy for the first three steps in the RSM algorithm. In the next section we offer a procedure to conduct statistical analysis under the CRN strategy for the second-order model, and show similar gains of using the CRN strategy for the second-order model which comprises Step 4 of the RSM algorithm.

### ***3.4. The Second-Order model***

In this section we discuss methods to conduct statistical analysis under the CRN strategy for the second-order model defined in (2.51). We also present theoretical results to

show the gains of using the CRN strategy over direct simulation for the second-order model.

Recall that the second-order model in equation (2.51) was written as

$$y_{ij} = \beta_0 + \sum_{l=1}^k \beta_l x_l + \sum_{h>l} \beta_{hl} x_h x_l + \sum_{l=1}^k \beta_{ll} x_l^2 + \varepsilon_{ij}, \quad (3.28)$$

for  $l = 1, 2, \dots, k-1, j = 1, 2, \dots, r, i = 1, 2, \dots, r$ , and all terms have the same interpretation as in (2.51). The above can be expressed in matrix notation identical to the one in (2.4), only that vector  $\beta_1$  for this model is represented by  $\beta_1' = (\beta_1, \beta_2, \dots, \beta_k, \beta_{12}, \dots, \beta_{k,k-1}, \beta_{11}, \dots, \beta_{kk})$ . Let the vector  $\beta_1$  be partitioned as  $\beta_1 = (\beta_f' | \beta_s')'$ , where  $\beta_f$  is the vector of coefficients for the first-order and cross product terms; that is  $\beta_f' = (\beta_1, \beta_2, \dots, \beta_k, \beta_{12}, \dots, \beta_{k,k-1})$ ; and  $\beta_s$  is the vector for the pure second-order (quadratic) terms; that is,  $\beta_s' = (\beta_{11}, \beta_{22}, \dots, \beta_{kk})$ .

Statistical analysis for the second-order model parallels the one described for the first-order model in Section 2.4 if the second-order design selected is orthogonal. However, an orthogonal design does not always satisfy design requirements. For example, if prediction variances are required to be under certain limits, then sometimes a rotatable ccd is preferred over an orthogonal ccd. The drawback of a non-orthogonal second-order design is that the estimated second-order coefficients are usually correlated, and hence the statistical analysis needs to be modified. In the RSM algorithm described in Section 3.1, we use a rotatable ccd, and hence a modified statistical analysis for this design is provided. We illustrate the problem and its analysis for two variables, and then present results for a general case.

Consider a rotatable ccd in a two variable situation. The design matrix  $\mathbf{D}$  for a rotatable ccd is given as:

$$\mathbf{D} = \begin{bmatrix} -1 & -1 & +1 & +1 & +1 \\ -1 & +1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & +1 \\ +1 & +1 & +1 & +1 & +1 \\ +1.414 & 0 & 0 & 2 & 0 \\ -1.414 & 0 & 0 & 2 & 0 \\ 0 & +1.414 & 0 & 0 & 2 \\ 0 & -1.414 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.29)$$

We notice that this design is not orthogonal. The variance-covariance matrix of  $\hat{\beta}_1$  under the CRN strategy is given as:

$$\text{var}(\hat{\beta}_1) = (\mathbf{D}'\mathbf{D})^{-1} \text{var}(\mathbf{y}) = \sigma^2(1 - \rho_+) \begin{bmatrix} 0.125 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.125 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.250 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.937 & -0.03 \\ 0.000 & 0.000 & 0.000 & -0.03 & 0.937 \end{bmatrix}. \quad (3.30)$$

Notice that under the IS strategy, the structure of  $\text{var}(\hat{\beta}_1)$  would remain the same as that in (3.30), with the only difference being that  $\sigma^2$  will replace by  $\sigma^2(1 - \rho_+)$ . Since the components of  $\hat{\beta}_1$  that are correlated with each other (and independent of other coefficient estimates) are the pure second-order coefficient estimates, we develop a modified statistical analysis to draw inferences regarding these coefficients. For this purpose, we transform the vector  $\beta$ , using an orthogonal transformation. Let  $\Gamma^{(SO)}$  be the orthogonal transformation whose columns comprise of the eigenvectors of  $\text{var}(\hat{\beta}_1)$ . Then, the transformed vector of second-order coefficients is given by:

$$\beta_s^* = \Gamma^{(SO)}\beta_s.$$

For a two variable design under the CRN strategy, we have:

$$\text{var}(\hat{\beta}_s) = \sigma^2(1 - \rho_+) \begin{bmatrix} 0.937 & -0.03 \\ -0.03 & 0.937 \end{bmatrix}. \quad (3.31)$$

For this design, we will use the orthogonal transformation:

$$\Gamma^{(SO)} = \begin{bmatrix} 0.707 & 0.707 \\ 0.707 & -0.707 \end{bmatrix}, \quad (3.32)$$

which yields

$$\text{var}(\hat{\beta}_s^*) = \sigma^2(1 - \rho_+) \begin{bmatrix} 0.907 & 0.000 \\ 0.000 & 0.967 \end{bmatrix}. \quad (3.33)$$

Therefore, we have

$$\hat{\beta}_s^* \sim N_k(\beta_s^*, \text{var}(\hat{\beta}_s^*)). \quad (3.34)$$

Inspection of (3.34) indicates that the model involving  $\hat{\beta}_s$  is an ordinary linear model and we can now form confidence intervals and perform the usual statistical tests on these unknown parameters. Once we know how these transformed parameters can be used to perform these tasks, we can rewrite the procedures in terms of the untransformed parameters (by taking the inverse transformation), and hence in terms of the original responses. Hence, this statistical analysis is model independent, and all inferences drawn on  $\beta_s$  parallel the ones in Results 2 and 3 for  $\beta_1$  described in Section 2.4.

To conduct statistical analysis for the  $k$ -variable problem, we can find a similar orthogonal transformation and proceed in an identical manner to the one described for the two variable problem above. Note that the variance-covariance matrix of  $\hat{\beta}$ , is always symmetric, and hence we can find an orthogonal transformation which is the matrix whose columns comprise of the eigenvectors of  $\hat{\beta}$ .

From results proved earlier, recall that we achieve a reduction of  $100\rho_+$  percent on the variance of all elements of the vector  $\beta_1$  under CRN strategy over direct simulation. These parameters are used to fit a second-order response surface model. The second-order response surface model is used to locate the stationary point of the system. By using the CRN strategy, the stationary point is better estimated than by using direct simulation. This is due to the fact that the stationary point is a function of the vector  $\hat{\beta}_1$  (see (2.54)). Since we compute  $\hat{\beta}_1$  with reduced variance using CRN, it is evident that we estimate the stationary point better under CRN strategy than we would using direct simulation. The precision of estimating the components of matrix  $\mathbf{B}$  directly affects in determining the nature of the stationary points as explained in Section 2.5.1 (immediately following equation (2.55)), which determines the nature of the response surface (convex, or concave, etc). This is another advantage of CRN strategy over direct simulation. Results similar to the ones developed for the first-order model in Section 3.2, which showed the reduction in volume of joint confidence ellipsoids of the elements of  $\hat{\beta}_1$  can also be applied for this model. The second-order model also plays an important role in the ridge analysis, which is focus of the next section.

In this section we discussed the statistical analysis methods for the second-order model under the CRN strategy. From the discussion on the gains of using the CRN strategy over direct simulation above, and also in Section 3.2, it is evident that if the magnitude

of the induced correlation between responses is high, then we have improved estimators of the second-order model in terms of their variances. These estimators play an important role in the canonical and ridge analyses in our quest to locate the optimum. In the next section we discuss the advantage of the CRN strategy over direct simulation in the context of ridge analysis.

### 3.5. Ridge Analysis

In this section we discuss the advantages of using the CRN strategy over direct simulation under ridge analysis. Ridge analysis is discussed in Section 2.5.1.

Recall from Section 2.5.1 that in the method of ridge analysis, we follow the falling ridge along points which are the "best" (minimum response) points on their respective radii given by  $(\mathbf{x}'\mathbf{x})^{1/2}$ . To obtain such design points, we solve the system of equations (given in (2.56)):

$$(\mathbf{B} - \mu\mathbf{I}_k) \mathbf{x} = \frac{-\mathbf{b}}{2}. \quad (3.35)$$

Since  $\det(\mathbf{B} - \mu\mathbf{I}_k) = 0$  if  $\mu$  is an eigenvalue of  $\mathbf{B}$ , and  $\mu$  is selected such that it is smaller than the smallest eigenvalue, we have that  $(\mathbf{B} - \mu\mathbf{I}_k)$  is non-singular. Hence, we get,

$$\mathbf{x} = (\mathbf{B} - \mu\mathbf{I}_k)^{-1} \frac{-\mathbf{b}}{2}, \quad (3.36)$$

and the radius  $r$  is given as

$$r = (\mathbf{x}'\mathbf{x})^{1/2} = \left[ \frac{1}{4} \mathbf{b}'(\mathbf{B} - \mu\mathbf{I}_k)^{-1}(\mathbf{B} - \mu\mathbf{I}_k)^{-1}\mathbf{b} \right]^{1/2}. \quad (3.37)$$

Since  $(\mathbf{B} - \mu\mathbf{I}_k)^{-1}$  is symmetric, (3.32) can be written as

$$r = \left[ \frac{1}{4} \mathbf{b}'(\mathbf{B} - \mu \mathbf{I}_k)^{-2} \mathbf{b} \right]^{1/2}. \quad (3.38)$$

From (3.38) we see that  $r$  is a function of the matrix  $\mathbf{B}$  and the vector  $\mathbf{b}$ , which are as defined in (2.52) and (2.53) respectively. Also, results in Sections 3.2 and 3.4 indicate that all the elements of the matrix  $\mathbf{B}$  and vector  $\mathbf{b}$  are better estimated under the CRN strategy than under direct simulation. Hence it is evident that the radii of the “best” (minimum response) design points while conducting ridge analysis are estimated better under the CRN strategy than using direct simulation. Recall that under the CRN strategy all elements of the matrix  $\mathbf{B}$ , and all the elements of the vector  $\mathbf{b}$  have their variances reduced by  $100\rho_+$  percent. Thus if the induced variance is large, then the variance reduction in each of those elements is significant, and this would result in substantial improvement in the performance of ridge analysis.

To obtain analytical results that display the advantages of using the CRN strategy instead of direct simulation, we tried to obtain the distribution of the radius,  $r$ , which is discussed next. Since all elements of  $\mathbf{B}$  are normally distributed, so are those of the matrix  $(\mathbf{B} - \mu \mathbf{I}_k)$ , since  $\mu$  is a constant. Hence the matrix  $(\mathbf{B} - \mu \mathbf{I}_k)$  has a matrix normal distribution (see pp. 310-313 of Arnold, 1981). If the rows in matrix  $\mathbf{B}$  had been independent, then we could have shown that the matrix  $(\mathbf{B} - \mu \mathbf{I}_k)^2$  has a non-central Wishart distribution, and the statistic  $r^2$  given by  $\frac{1}{4} \mathbf{b}'(\mathbf{B} - \mu \mathbf{I}_k)^{-2} \mathbf{b}$ , would then have a doubly non-central  $F$  distribution (see p.190 of Johnson and Kotz, 1970, and, p.319 of Arnold, 1981). Unfortunately, the rows of matrix  $\mathbf{B}$  are not independent, and hence we cannot find the distribution of  $r$ . Since no analytical solution is attainable, we plan to conduct an empirical study to explore the advantages, if any, of using the CRN strategy.

In this chapter we presented a slightly modified RSM algorithm rather than using a more conventional approach. Analytical results were also presented which prove the validity of using the CRN approach over direct simulation (IS strategy). The gains of using CRN were seen for the first-order model, the gradient search, the second-order model, and the ridge analysis. We noted that an empirical study needs to be done to study the use of CRN strategy over direct simulation for the ridge analysis which is a part of the proposed work addressed in Chapter 5. In the next Chapter we apply the RSM algorithm discussed in Section 3.1 to solve an optimization problem associated with a job-shop model. The model formulation and computational results under the two simulation strategies form the core of the next chapter.

## CHAPTER IV Example

This chapter describes an example that was used to compare the relative effectiveness of CRN and IS strategies in the context of the RSM algorithm described in Section 3.1. To justify the validity of these quantitative results, fifty different randomly selected starting points were chosen. Random selection of starting points provides an opportunity to make probabilistic assertions about the relative performance of the two simulation strategies. The relative performance of the two strategies is quantified with computational results using numerous performance measures that focus on different algorithm characteristics across all fifty searches. The computational results are categorized into two classes which characterize: (a) *accuracy*, and (b) *speed*, of the RSM algorithm. An important statistic of interest which relates to both, the speed and the accuracy of the algorithm is the average gain in response values per simulation run (or design point). All these computational results are presented in Section 4.2. Section 4.1 details the problem statement.

### *4.1 Problem Statement*

The problem under study is described on p. 460 of Pritsker (1986). The SLAM II code for the same is presented in Appendix 1. A job shop consists of six machines with each machine performing a different operation. The estimated processing time for each machine follows an exponential distribution (processing times are rounded off to integer

values with no value being less than 1). Actual processing time is equal to the estimated processing time plus a random component which is normally distributed with a mean of zero and standard deviation equal to three-tenths of the estimated processing time (the random component is white (or Gaussian) noise). Note that the job shop is balanced so that each machine has the same average processing time.

Jobs arrive to the shop with interarrival times being exponentially distributed. In our study, this continuous variable is denoted by  $\xi_1$ , which has mean,  $x_1$ . The interarrival times are integerized and must be greater than or equal to 1. Each job consists of a set of operations to be performed on the machine in the job shop. The number of operations per job is normally distributed with a mean of 4 and a standard deviation of 1. However, no job can require less than 3 operations nor more than 6 operations. The routing of the job through the machines is determined by random assignment. The dispatching rule included in this example is the  $SI^x$  rule which processes jobs at a machine in the order of the shortest estimated processing time for the job that is in a high priority class. The continuous variable, processing time is denoted by  $\xi_2$ , which has mean,  $x_2$ . Priority jobs are defined as those jobs whose float is negative. Float is defined for a job as the due date minus the current time minus the estimated time remaining to perform operations on the job minus a safety factor. The  $SI^x$  rule divides jobs in front of a given machine into two classes and within each class the jobs are ordered based on shortest estimated processing time.

The design space for the two variables in our study is

$$50 \leq x_1 \leq 110, \text{ and}$$

$$15 \leq x_2 \leq 40.$$

Denote the sojourn time in the system for each job to be  $ST$ . The computer model is simulated until five hundred jobs (observations) finished processing. The mean sojourn time of these five hundred jobs is recorded as the sojourn time,  $ST$  for further calculations. Since the mean sojourn times are considered as primary statistics of interest, the normality assumption for further statistical analysis seems to be reasonable. Also, define  $Z$  as:

$$Z = ST + 0.5x_1 - 5x_2 - 0.02x_1x_2$$

The response of interest, denoted by  $y$ , has its mean value evaluated as follows:

$$E[y] = \begin{cases} Z & \text{for } 50 \leq x_1 < 81, \text{ and } 15 \leq x_2 < 31. \\ Z - 10(x_1 - 81)^2 - 20(x_2 - 31)^2 & \text{for } 81 \leq x_1 \leq 100, \text{ and } 31 \leq x_2 \leq 35 \\ Z - 3930 + 2(x_1 - 81)^2 + 2(x_2 - 31)^2 & \text{for } x_1 \geq 100, \text{ and } x_2 \geq 31 \\ Z - 3930 + 2(x_1 - 81)^2 + 2(x_2 - 31)^2 & \text{for } x_1 \geq 81, \text{ and } x_2 \geq 35 \end{cases}$$

For this Monte-Carlo study, it is assumed that the problem under study is a minimization problem and that the analyst has no prior knowledge of the functional form of the response function which is defined above.

As the RSM search seeks the mean optimal response, appropriate models which represent the effects of the independent variables on the mean response adequately, need to be employed in a specific region of experimentation. The functional form of the objective function  $y = f(\xi_1, \xi_2)$ , is unknown over the entire design space of the independent variables. It is assumed that at any stage of the investigation the model can be represented as a first-order model, or a second-order model in  $\xi_1$  and  $\xi_2$ . The first-order metamodel is:

$$y = \beta_0 + \beta_1\xi_1 + \beta_2\xi_2 + \beta_3\xi_1\xi_2 + \varepsilon, \quad (4.1)$$

where  $\beta_i$  ( $i = 0, 1, 2, 3$ ) are the unknown regression coefficients, and  $\varepsilon$  is the error term. The uncoded high and low levels respectively for the first variable, mean time between arrival are set at +2 and -2, and those for the second dependent variable, mean service time for the machines are set at +1 and -1 around the design center. The uncoded values for the two decision variables are different since the design space for  $\xi_1$  is larger than that of  $\xi_2$ , and also the response is less sensitive to changes in  $\xi_1$  than to changes in  $\xi_2$ . For the expanded design, the uncoded high and low levels respectively for  $\xi_1$  are set at +5 and -5, and those for  $\xi_2$  are set at +2 and -2. The design matrix,  $\mathbf{D}$ , defined in Section 2.1 for the first-order model is given by

$$\mathbf{D} = \begin{bmatrix} -1 & -1 \\ -1 & +1 \\ +1 & -1 \\ +1 & +1 \\ 0 & 0 \end{bmatrix}. \quad (4.2)$$

For the first-order design, simulations at the five design points (which comprise of the five rows of (4.2)) are performed using the same random number seeds to drive the stochastic components under the CRN strategy, and using independent random number seeds under the IS strategy. Further, at each design point, two independent replications are performed. Ten responses are thus obtained from the first-order model. The responses from these simulations are as described earlier in this section. The second-order design is constructed by augmenting the first-order design with axial points at coded distances of  $\sqrt{2}$  from the design center along each direction of both variable axes. Simulation runs are conducted at these axial points using the same random number

seeds under the CRN strategy, and using independent random number seeds under the IS strategy (the random number seeds under the IS strategy are selected from *A Million Random Digits with 100,000 Normal Deviates*, published by the Rand Corporation). In addition, two independent simulation runs are performed at each design point throughout the entire search.

The second-order model is given by:

$$y = \beta_0 + \beta_1\xi_1 + \beta_2\xi_2 + \beta_{11}\xi_1^2 + \beta_{22}\xi_2^2 + \beta_3\xi_1\xi_2 + \varepsilon, \quad (4.3)$$

where  $\beta_{11}$  and  $\beta_{22}$  are the coefficients of  $\xi_1^2$  and  $\xi_2^2$  respectively, and all other terms are as defined in (4.1). The design matrix for the second-order model, which is a rotatable ccd, is defined by:

$$\mathbf{D} = \begin{bmatrix} -1 & -1 & +1 & +1 & +1 \\ -1 & +1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & +1 \\ +1 & +1 & +1 & +1 & +1 \\ +1.414 & 0 & 0 & 2 & 0 \\ -1.414 & 0 & 0 & 2 & 0 \\ 0 & +1.414 & 0 & 0 & 2 \\ 0 & -1.414 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4.4)$$

The optimization procedure using the RSM algorithm from Section 3.1 on a sample search problem is described next. For this purpose, consider search number 21 conducted under the CRN strategy. For this problem, the same random number seeds are used to drive the stochastic components and two independent simulation runs are performed at all design points.

Using Step 1, the location of the starting point is selected to be (81.65,32.3) which corresponds to coordinates  $(x_1, x_2)$ , and the coded coordinates being (0,0), respectively. Two independent simulation runs are performed, and the responses, recorded. In Step 2, a first-order  $2^2$  full factorial model was built around the selected starting point. That is, at coded design points  $(-1, -1)$ ,  $(-1, +1)$ ,  $(+1, -1)$ ,  $(+1, +1)$  which correspond to uncoded values (79.65,31.3), (79.65,33.3), (83.65,31.3), (83.65,33.3), respectively, simulation experiments are performed under the CRN strategy. This first-order model is not found to represent the data adequately (or, was not a good fit), and hence this design is augmented with axial points to accommodate a rotatable ccd. That is, we proceeded with Step 4 of the RSM algorithm. The augmented coded levels are  $(\sqrt{2}, 0)$ ,  $(-\sqrt{2}, 0)$ ,  $(0, \sqrt{2})$ ,  $(0, -\sqrt{2})$ , which correspond to uncoded coordinates, (84.48,32.3), (78.82,32.3), (81.65,33.71), (81.65,30.89), respectively, at which simulation experiments are performed under the CRN strategy. The second-order model is also not found to be an adequate fit, and hence this necessitates design expansion. Thus, Step 5 of the algorithm is undertaken. Under design expansion, the coded values of  $(-1, -1)$ ,  $(-1, +1)$ ,  $(+1, -1)$ ,  $(+1, +1)$  are made to correspond to the respective uncoded values of (76.65,30.3), (76.65,34.3), (86.65,30.3), (86.65,34.3). Using this model the direction of steepest descent is computed. The direction computed is (690.0,268.0). The best response along this gradient is observed at the design point (96.35,38.13) which is then selected as the center of a new first-order design as prescribed in Step 3 of the RSM algorithm. The search procedure until this stage is summarized in Figure 1. The rest of the search until the optimal is reached, is illustrated in Figure 2.

Locations of the first-order design points are (94.35,37.13), (94.35,39.13), (98.35,37.13), (98.35,39.13). The first-order design con-

Starting Point Location: (81.65, 32.3)  
 Mean Response: (-33.45), Variance: (0.0014)

**First-Order Model:**

X1	X2	Mean Response	Variance
79.65	31.3	2.55	4.95
79.65	33.3	5.82	3.38
83.65	31.3	71.96	3.22
83.65	33.3	-175.09	10.15

First-order model is not a good fit.  
 Augment the design to build a second-order model.

**Augmented (Axial) Design Points**

X1	X2	Mean Response	Variance
84.88	32.3	-155.0	7.38
78.82	32.3	3.81	1.82
81.65	33.71	-146.07	3.89
81.65	30.89	-1.3	3.37

Second-order model is not a good fit.

**Expanded Design:**

X1	X2	Mean Response	Variance
76.65	30.3	5.03	3.1
76.65	34.3	5.95	1.13
86.65	30.3	-2.41	5.36
86.65	34.3	-538.5	14.27

Direction of steepest descent: (690.0, 280.0)  
 Location of the best point: (96.35, 38.13)  
 Mean response value: - 3346.42, Variance : 19.87

Figure 1. Search Procedure for Starting Point # 21 under the CRN strategy.

Center of a new design at location: (96.35, 38.13)

**First-Order Model:**

X1	X2	Mean Response	Variance
94.35	37.13	-3487.66	13.62
94.35	39.13	-3433.47	0.19
98.35	37.13	-3241.92	0.51
98.35	39.132	-3183.55	0.51

First-Order Model is not a good fit.  
Augment the design to build a second-order model.

**Augmented (Axial) Design Points:**

X1	X2	Mean Response	Variance
99.18	38.13	-3154.33	14.11
93.52	38.13	-3504.81	1.09
96.35	39.54	-3301.14	3.56
96.35	36.72	-3382.37	7.12

Second-order model is a good fit.

Stationary point location: (82.13, 33.13)

Mean Response: -100.73

Stationary point outside design region.

Direction of steepest descent: (-248.8, -28.4)

Location of best response: (81.65, 36.44)

Mean Response: -3866.5, Variance 7.93

Location of optimal point: (81.65, 36.44)

Figure 2. Search Procedure continued.

structured around this design point is found to represent the responses inadequately, and hence a second-order design was constructed. Locations of the additional axial points to complete the second-order design are (93.52,38.13), (99.18,38.13), (96.35,36.72), (96.35,39.54). The second-order estimated model is accepted as representing the responses adequately. Canonical analysis is performed and the stationary point is found to be a minimum. The optimum design point however is located at coordinates (82.13,33.13), which is outside the design region. Therefore follow Step 8 and then consequently Step 7 of the algorithm to follow the path of steepest descent from the center of the current design ((96.35,38.13)) in quest for the search for better responses. The direction is computed to be (-247.8,-28.43). The best mean response along this gradient is observed at location (81.65,36.44) and is reported as the optimum.

Detailed computational results for the fifty searches under the two simulation strategies of CRN and IS are presented in the next section.

## ***4.2 Computational Results***

In this section a quantitative assessment of the efficiency of CRN over the IS strategy is provided. As mentioned earlier, the problem under study is solved under the two simulation strategies starting from fifty different starting points under each strategy. That is, there were fifty optimization searches conducted under the CRN strategy, and fifty optimization searches under the IS strategy.

The settings of the independent variables that yield local and global optima for the problem under study were known *a priori*, and were located at (81.01,35.01) and (100.0,35.0), respectively. A scatter plot of the fifty starting points that are selected for

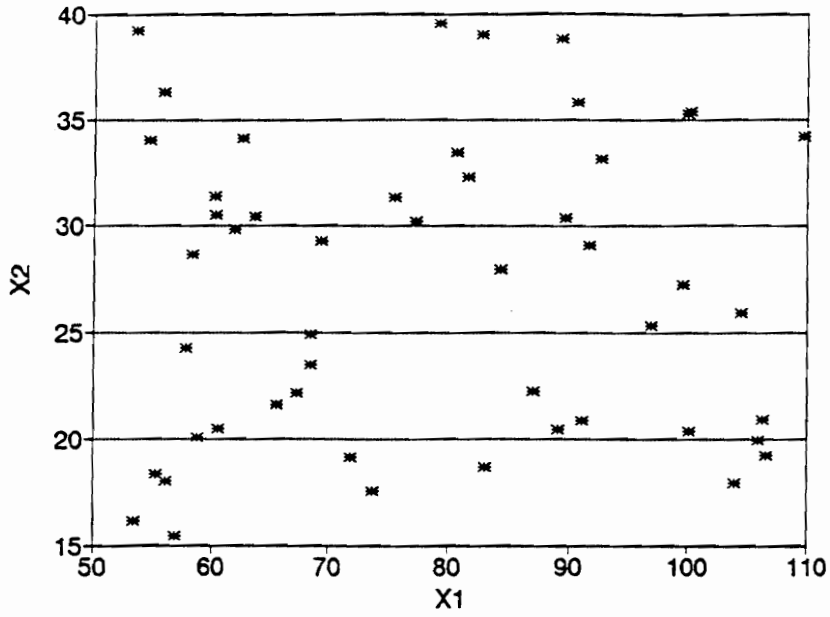


Figure 3. Scatter Plot for Locations of Starting Points.

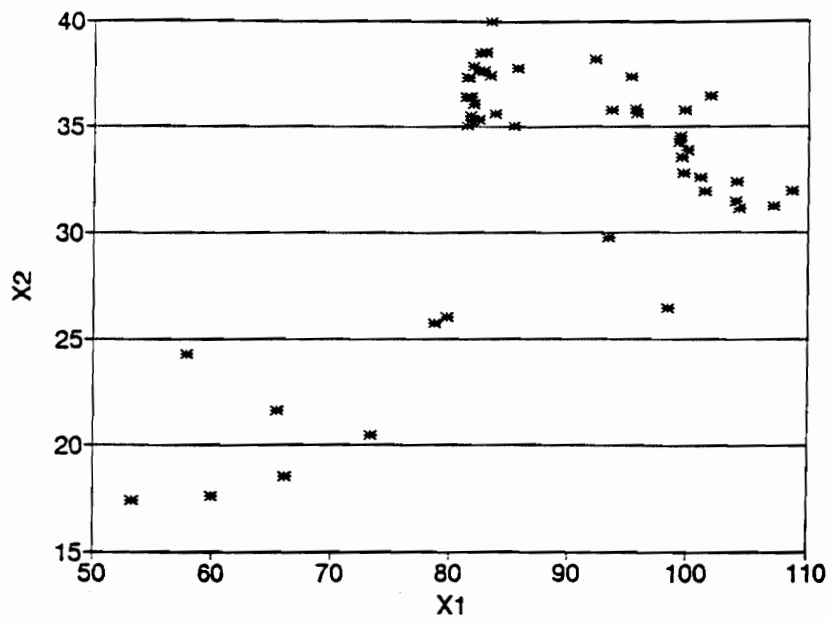


Figure 4. Scatter Plot for Locations of Stopping Points Under the CRN Strategy.



this study is presented in Figure 3. This sub-section describes the performance measures of the algorithm which are categorized into two categories: (a) speed and (b) accuracy of the algorithm.

The performance measures that characterize the accuracy of the algorithm are discussed first. The performance measures that quantify accuracy are: (a) the number of searches that fail to reach a minimum specified level of the mean response (NFML), (b) the number that reach the local optimum (NLO), (c) the number that reach the global optimum (NGO), (d) the average difference between the mean response at the stopping point of the search and the mean global optimal response (DRGO), (e) average of the difference in the mean response values from the starting point and stopping point of a search (DIMR), (f) the average distance from any point along the search from the local optimum (ADLO), (g) the average distance of the stopping point of the search from the global optimum (ADGO), and (h) best and worst mean responses recorded (BERE and WORE, respectively).

For the performance measure NFML, a minimum specified level of response is set at -1000. Across the fifty searches, if any mean optimal response falls above this value, then that search is considered to have failed to reach either the global or the local optimum. If the location of the optimal is found in the vicinity of (81.01, 35.01), and if the search has not failed, then the search is assumed to have stopped at the local optimal. If the location of the optimal is found in the vicinity of (100.0,35.0), and if the search has not failed, then it is assumed that the global optimal is reached.

DRGO, the average difference between the mean response at the stopping point using RSM and the mean global optimal response (already known to be -3900), is computed as:

$$DRGO = \frac{\sum_{i=1}^{50} [\bar{y}_{optimal,i} + 3900]}{50} , \quad (4.5)$$

where  $\bar{y}_{optimal,i}$  ( $i = 1, 2, \dots, 50$ ) is the mean optimal response for the  $i$ th search. Small values of DRGO are naturally preferred.

Let  $\bar{y}_{start,i}$  ( $i = 1, 2, \dots, 50$ ) be the mean response at the  $i$ th starting point, then the average difference between the mean response at starting and stopping points of the search (DIMR) is calculated as

$$DIMR = \frac{\sum_{i=1}^{50} [\bar{y}_{optimal,i} - \bar{y}_{start,i}]}{50} . \quad (4.6)$$

The simulation strategy with smaller DIMR is preferred, since lower values of DIMR indicate higher average improvement.

Prior knowledge of the location of the local and global optima enables the analyst to comment on the quality of any search conducted. For example, if a search started at location (65,23), and stopped near the local optimal at coordinates (82,36), then the search could have either proceeded in a smooth direction towards the local optimal, or it could have taken a longer path. The former search would hence have performed qualitatively better than the latter. That is, the average distance from the true optimum of each location along a search can be an indication of the accuracy of that search, so that, a large average distance would imply lower accuracy. Hence, this is another important performance measure characterizing accuracy of the algorithm. If

$n_i$  ( $i = 1, 2, \dots, j$ ) represents the number of design points (locations) needed for the  $i$ th search to reach the local optimal, and that  $j$  out of the fifty searches reach the local optimal, then the average distance of each design point (location) along the search from the local optimal (ADLO), is defined as a paired statistic and is calculated as:

$$ADLO = \frac{\sum_{i=1}^J \sum_{l=1}^{n_i} [(x1,x2)_l - (81.01,35.01)]}{jn_i}, \quad (4.7)$$

where  $(x1,x2)_l$  ( $l = 1, 2, \dots, n_i$ ) are the coordinates at the  $l$ th location of the  $i$ th search ( $i = 1, 2, \dots, j$ ), and all other terms are defined above. Note that ADLO is a paired statistic in which the elements represent the average of the difference between that elements and the corresponding coordinates of the optimal location.

In a similar vein, define another paired statistic which is the average distance of each design point (location) along the search from the global optimal (ADGO) as:

$$ADGO = \frac{\sum_{i=1}^J \sum_{l=1}^{n_i} [(x1,x2)_l - (100.0,35.0)]}{jn_i}, \quad (4.8)$$

where  $(x1,x2)_l$  ( $l = 1, 2, \dots, n_i$ ) are the coordinates of the  $l$  location of the  $i$ th search ( $i = 1, 2, \dots, j$ ),  $n_i$  is the number of design points (locations) needed for the  $i$ th search to reach the global optimal, and in this case,  $j$  searches reach the global optimal.

Accuracy of an algorithm also can be measured using the best and the worst mean optimal responses achieved which are denoted respectively by BERE and WORE. The

performance measures of the algorithm that characterize the speed of the algorithm are discussed next.

The speed of the algorithm is quantified by: (a) the average number of runs required to perform a search (ANR), (b) the number of simulation runs needed in a search, given that the search reaches the local optimum (ANRL), and (c) the number of simulation runs needed in a search, given that the search reaches the global optimum (ANRG).

In many situations, simulation experiments are either prohibitively expensive, or that they take too long to be feasibly conducted at more than some specified number of design points. Under such circumstances, the average number of simulation runs required to conduct a search (ANR) evidently becomes an important measure of performance for the algorithm. If  $n_i$  ( $i = 1, 2, \dots, 50$ ) represents the number of design points (locations) needed to stop the  $i$ th search, then ANR is calculated as:

$$ANR = \frac{\sum_{i=1}^{50} n_i}{50} . \quad (4.9)$$

The statistic ANR in certain situations can however be misleading since it takes into account *only* the number of design points needed to *stop* the search, and not reach either optima. A fair assessment of the speed of the algorithm would therefore be to compare the number of locations required under different strategies conditioned on whether the search reaches the global or the local optima. Consider the performance measure denoted by ANRL, which is the average number of design points needed for the  $i$ th search given that the search stops at the local optimum, and is calculated as:

$$ANRL = \frac{\sum_{i=1}^j n_i}{50}, \quad (4.10)$$

where  $j$  is the number of searches that reach the local optimal. Similarly define the performance measure denoted by ANRG, which is the average number of design points needed for the  $i$ th search given that the search stops at the global optimum. It is then calculated as:

$$ANRG = \frac{\sum_{i=1}^J n_i}{50}, \quad (4.11)$$

where  $j$  is the number of searches that reach the global optimal.

An important statistic of interest which relates to both, the speed and the accuracy of the algorithm is the average gain in response values per simulation run (or design point), denoted by GPSR, and given by:

$$GPSR = \frac{\sum_{i=1}^{50} [\bar{y}_{optimal,i} - \bar{y}_{start,i}]}{\sum_{i=1}^{50} n_i}, \quad (4.12)$$

where all quantities are defined before. Notice from (4.6), (4.9), and (4.12) that GPSR can be written in terms of DIMR and ANR as

$$GPSR = \frac{DIMR}{ANR}. \quad (4.13)$$

Since DIMR is a performance measure characterizing accuracy, and ANR is one that characterizes speed, from (4.13) it is evident that GPSR relates to both, speed and accuracy.

Table 1 provides a summary of all the statistics computed for the example under study.

From the first three statistics in Table 1, notice that under the IS strategy thirty-one searches failed to reach either the local or the global optima compared to only ten under the CRN strategy. Also, under the IS strategy fifteen searches resulted in reaching close to the local optimal compared to twenty-one under the CRN strategy. Finally, under the IS strategy only four searches of the total fifty reached close to the global optimal compared to nineteen under the CRN strategy.

Scatter plots of the stopping points of the algorithm under the CRN and IS strategies are provided in Figures 4 and 5 respectively. Notice from these scatter plots that a large number of searches under the CRN strategy stopped close to either the local or global optima as compared to those under the IS strategy. These performance measures are also illustrated by condensing them in the form of pie-charts as seen in Figure 6.

Another way to view these performance measures is by observing the histogram of the optimal objective function values attained by the searches for all fifty runs under the two strategies. These histograms are presented in Figures 7 and 8 for the CRN and IS strategies respectively. Observe that for the minimization problem considered in this example, the number of searches that yield response values around -3600 is much higher under the CRN strategy than under the IS strategy. Also, the number of searches that yield high response values around 0 is much lower under the CRN strategy than under

Table 1. Performance Measures Characterizing Accuracy.

Performance Measures	CRN Strategy	IS Strategy
NFML	10	31
NLO	21	15
NGO	19	4
DGO	1495.74	2911.66
DIMR	-2434.27	-1018.34
ADLO	(14.1, 4.1)	(21.0, 7.61)
ADGO	(15.8, 6.98)	(21.3, 8.04)
BERE	-3893.81	- 3897.18
WORE	11.32	37.28

NFML = Number of searches that fail to reach a minimum specified level;

NLO = Number of searches that reach the local optimal;

NGO = Number of searches that reach the global optimal;

DGO = Average distance of the mean response value at stopping points and the true mean global response;

DIMR = Difference in response values between the starting and stopping points;

ADLO = Average distance from coordinates of any point along the search from coordinates of the local optimal;

ADGO = Average distance from coordinates of any point along the search from coordinates of the global optimal;

BERE = Best mean optimal response attained;

WORE = Worst mean optimal response attained;

Table 2. Performance Measures Characterizing Speed.

Performance Measure	CRN Strategy	IS Strategy
ANR	26.2	19.4
ANRL	38.28	32.88
ANRG	23.5	20.75

ANR = Average number of design points required to stop the search;

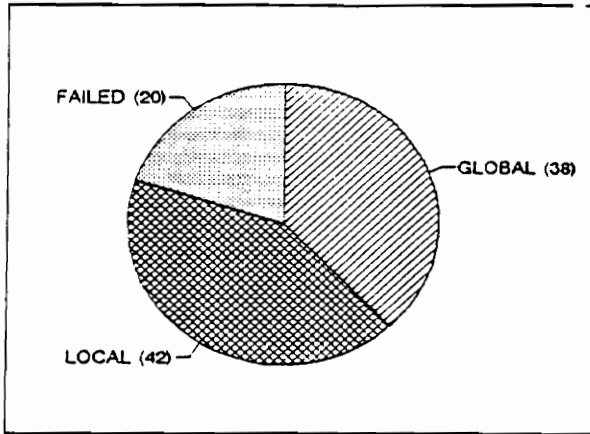
ANRL = Average number of design points required to stop the search in the vicinity of the local optimal;

ANRG = Average number of design points required to stop the search in the vicinity of the global optimal;

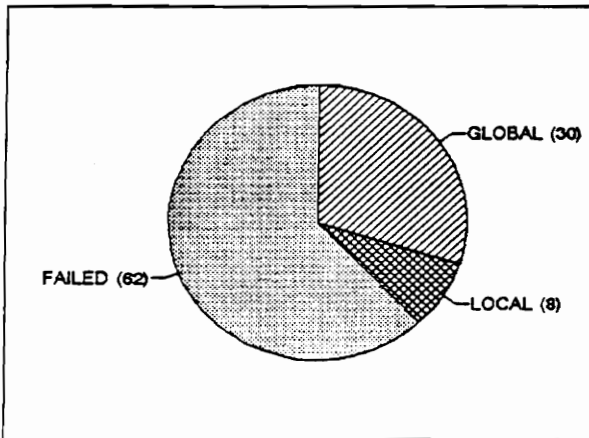
Performance Measures characterizing Speed and Accuracy:

Performance measure	CRN Strategy	IS Strategy
GPSR	-102.55	-32.65

GPSR = Gain in response value per simulation run.



CRN STRATEGY



IS STRATEGY

Figure 6. Pie-Charts to compare CRN and IS strategies under RSM.

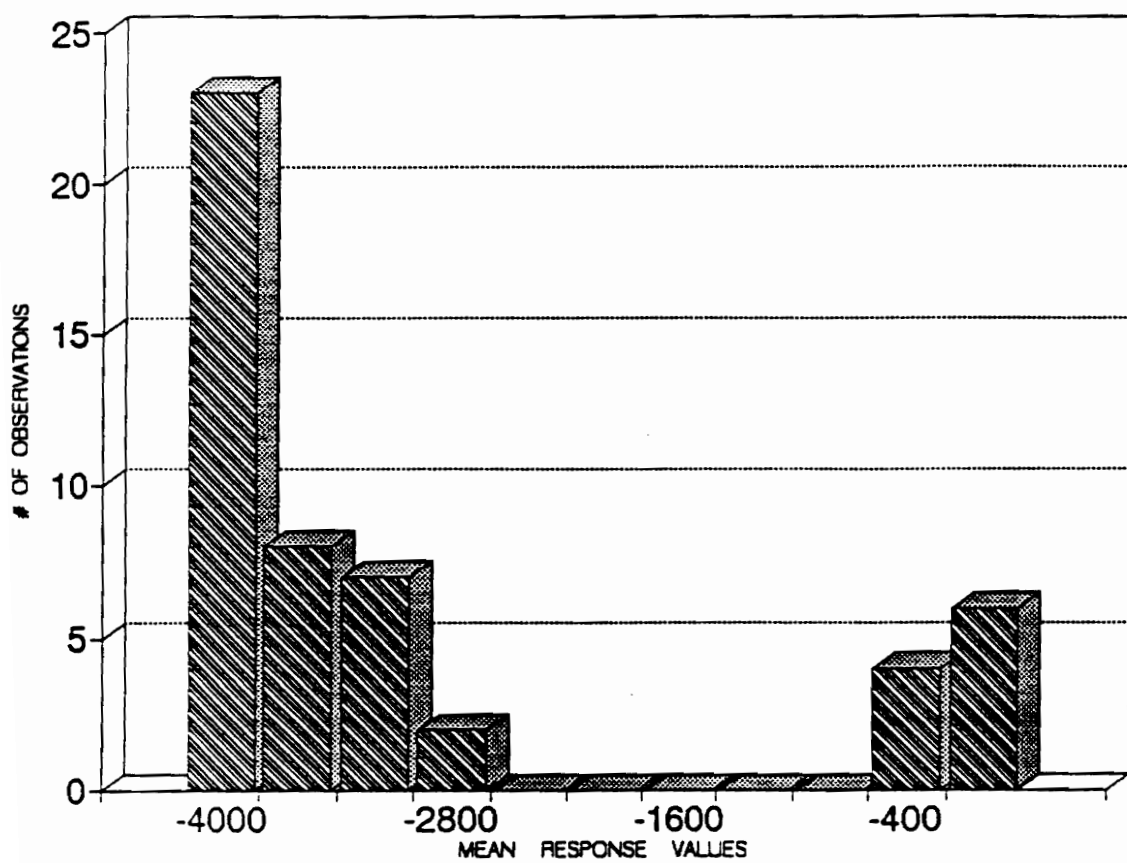


Figure 7. Histogram of mean responses observed under the CRN strategy.

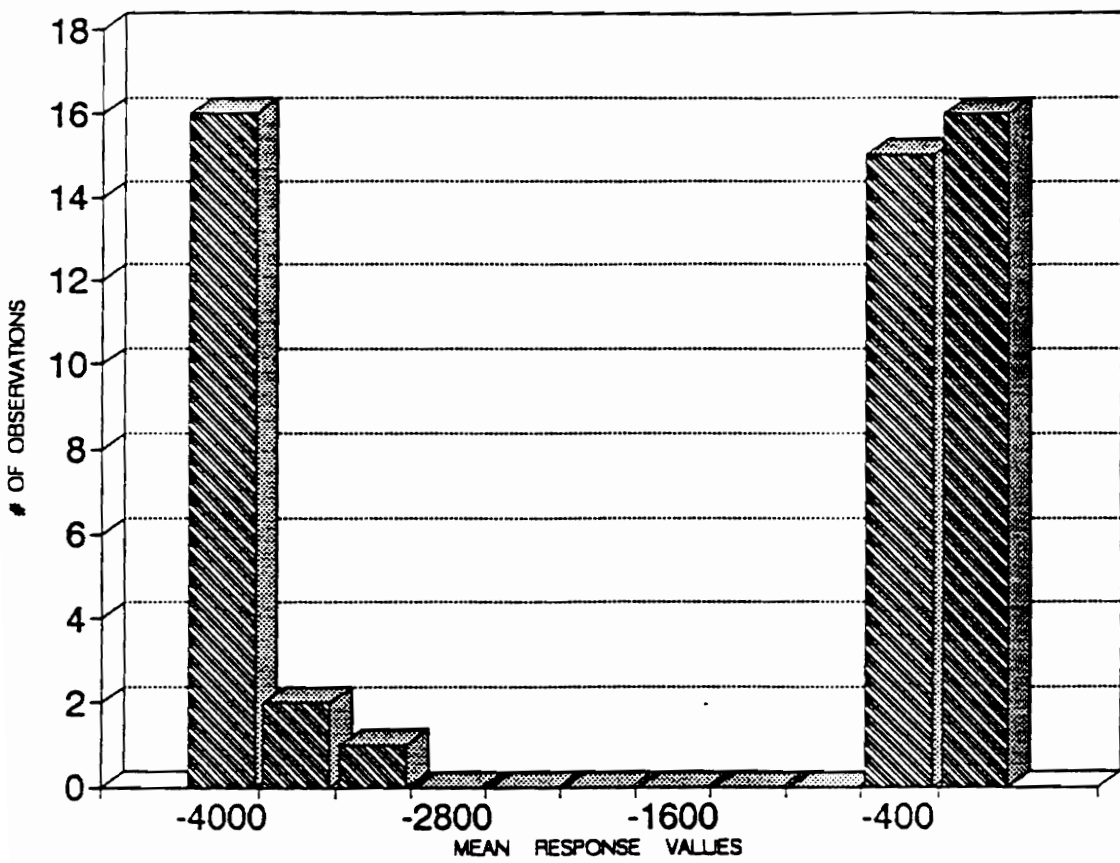


Figure 8. Histogram of mean responses observed under the IS strategy.

the IS strategy. These histograms clearly ascertain the advantage of using the CRN strategy.

Recall that DRGO is a performance measure that quantifies the accuracy of a search algorithm by considering the average of the difference between the mean response value at the stopping point of the algorithm, and the mean response at the global optimum. Under CRN this average value was found to be 1495.74, while the average value under direct simulation was computed to be 2911.66, which illustrates that searches under the CRN strategy reached closer to the best attainable mean response value than the searches under the IS strategy.

The average difference in the response values between the responses at the starting point and at the end of the search across all fifty searches denoted by DIMR is computed from (4.6) to be -2434.27 under CRN, versus -1018.34 under the IS strategy, thus exhibiting overwhelming gains achieved by using the CRN strategy.

The standard deviation on DIMR under the CRN and IS strategy respectively were 1618.77 and 1626.95.

The average distance of the optimal point found using the RSM search method from the local optimum across all fifty searches is expressed as a paired statistic, with the first element being the distance along  $x_1$ , and the second being the distance along  $x_2$ . This statistic was (14.1,4.1) under the CRN strategy compared to (21.0,7.61) under direct simulation. Lower magnitudes on the elements of ADLO under the CRN strategy compared to those for the IS strategy indicates that on an average, the searches under the CRN strategy reached closer to the local optimum than under the IS strategy.

A similar paired statistic is the average distance of any point selected along the search procedure from the global optimum. This was computed to be (15.8,6.98) under CRN, and (21.3,8.04) under direct simulation, further illustrating consistent improvement in the performance of the algorithm under the CRN strategy over direct simulation.

The best response under CRN strategy was -3893.81, and that under IS strategy was -3897.18. These values were very close, and the difference can be attributed to the randomness in the system. The worst response under CRN strategy was 11.32, and that under IS strategy was 37.28.

Computational results quantifying the speed of the algorithm under the two simulation strategies are discussed next.

The performance measure, ANR, which is defined as the average number of design points (locations) required to stop the search procedure, was computed to be 26.2 under the CRN strategy, and 19.4 under the IS strategy. This is understandable since relatively less number of searches failed to reach a minimum specified level under the CRN strategy, than under the IS strategy, which in turn added to the number of functional evaluations under the former strategy.

The performance measure, ANRG, which is the number of design points needed to stop the search in the vicinity of the global optimum is computed to be 23.5 under the CRN strategy, as compared to 20.5 under the IS strategy. These results are not detracting to the CRN strategy since searches from certain starting points that reached the global optimal under the CRN strategy and failed to reach a minimum specified level under the IS strategy. These types of searches under the CRN strategy required more number of simulation runs to reach the global optimum, than the searches conducted under the

CRN strategy from starting points which reached the global optimum under direct simulation as well. This resulted in increasing the average number of runs under CRN strategy to reach the global optimum compared those under direct simulation.

The statistic ANRL is calculated to be 38.28 under the CRN strategy, as compared to 32.88 under the IS strategy. Interpretation for this performance measure is identical to the one provided for ANRG.

One of the most significant and important statistics of interest that relates to the speed as well as the accuracy of the algorithm, and also reflects the effectiveness of one simulation strategy over another is the average gain per design point (that is, per two simulation runs), GPSR, which is recorded to be -102.55 under the CRN strategy, and -32.65 under the IS strategy. That is, for each design point used under the CRN strategy, responses decreased by 102.55, versus 32.65 under the IS strategy. This shows a three-fold improvement in response values per design point of the CRN strategy over direct simulation. This performance measure can be critical in situations where simulation experiments are prohibitively expensive.

Recall from Section 3.5 that analytical results could not be presented to qualitatively assess the gains of using the CRN strategy over the IS strategy when a ridge system is encountered by the RSM search. The next section presents an empirical study to quantitatively assess their use of CRN over IS strategy.

### ***4.3 Empirical Study for a Ridge System***

This section presents the results of a Monte-Carlo study to analyze the use of the CRN over the IS strategy when the RSM search encounters a ridge system. For this purpose,

a situation of an interesting ridge system in one of the fifty searches of the study in Section 4.2 is selected. In this situation, the optimum cannot be *pinpointed*, and hence further analysis and experimentation is needed. Experiments are therefore conducted along the falling ridge in the quest for the optimum.

To locate design points along the falling ridge, (2.56) is employed by setting the Lagrange multiplier  $\mu$  equal to zero, and solving for  $x$ . This is clearly a deviation from the classical ridge analysis, but uses the same search direction as in the ridge analysis. This search however investigates design points outside the design region unlike in ridge analysis, where observations are taken only inside the design region. The direction followed is then the *Newton* direction (from (2.56) and p.308 of Bazaraa, Sherali, and Shetty, 1993), and the most favorable design point along this direction is reported as the optimum.

For the purpose of this Monte-Carlo analysis, the center of a second-order model selected, is at the location (102.29,36.37). The mean response at this point is recorded as -2978.3. Canonical analysis indicates the presence of a ridge system. Twenty-five independent replications under the CRN and IS strategies are performed to compare relative performance of the two simulation strategies. The average optimal response function value is reported to be -3638.26 under the CRN strategy and -3451.9 under the IS strategy. This clearly illustrates the improvement of the algorithm under the CRN strategy relative to the IS strategy. It is also observed that *all* searches under the CRN strategy resulted in an improvement from the starting point (center of the design), and reached close to the global optimal. Under the IS strategy, four searches failed to track any improving direction, thirteen searches reached close to the global optimal, and eight reached close to the local optimal.

The computational results thus quantify in several ways, the superiority of CRN over the IS strategy for the RSM algorithm presented in Section 3.1. Other ways to improve upon the existing RSM algorithm using gradient deflection methods are the focus of the next chapter.

# CHAPTER V A Novel RSM Algorithm

This chapter presents a novel RSM algorithm which incorporates certain gradient deflection methods discussed in Section 2.6 instead of using the method of steepest descent only. This new RSM algorithm attempts to improve upon the search direction described in the RSM algorithm in Section 3.1 in addition to presenting additional modifications. This forms the core of Section 5.1. Section 5.2 discusses the particular gradient deflection methods that are incorporated into the existing RSM algorithm, and Section 5.3 specifies the restarting criteria used in this study.

## *5.1 Modified RSM Algorithm*

This section discusses the novel RSM algorithm. Before the algorithm is presented, some preliminary observations about the problem are warranted. It is assumed that the problem at hand is a minimization problem in  $k$  variables. All variables are assumed to have upper and lower bounds which cannot be violated. If along a search direction, certain variables reach their lower or upper bounds, then those variables are set at their respective bounds, and the search is continued. In a design however, we can set the variables above or below their bounds (for example, when the center of a design lies on the boundary of any variable). The algorithm is described as follows.

### Step 1.

Select a starting point.

### Step 2.

Construct a  $2^k$  full factorial experiment. For the first pass of the algorithm, use the starting point as the center of this design.

If the first-order model is a good fit, then go to Step 3.

Otherwise, go to Step 4.

### Step 3.

For the first pass of the algorithm, follow the path of steepest descent. For further passes, follow the appropriate "gradient deflection" direction. The first design point along this direction is selected such that some variable reaches its upper or lower bound (note that if any variable was already at its upper or lower bound, and if the search direction followed must violate the bound, then that variable is fixed at its boundary value, and the search is continued along other variables in the same prescribed direction). Next a line search is conducted between this design point and the starting point for this search, and the most favorable point in this interval is selected as the center of a new design. Step 2 is then repeated, unless if the most favorable point is any corner point of this interval. If the best response is observed at the starting point, then the algorithm stops with the most favorable point in the current design reported as the optimum. Go to Step 9. If the most favorable point is the end-point of this interval, then this point becomes the starting point of a similar search described in this step above, and the search continues. If all the variables have reached their upper or lower bounds, and if the response is still improving, then the search is stopped, and this point is reported as the optimum. Go to Step 9.

### Step 4.

Construct a second-order design (ccd) 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.

If the second-order model is not a good fit, then go to Step 5.

#### Step 5.

Expand (or contract) the design appropriately and using the current design center as the center of a new design, go to Step 2.

#### Step 6.

Perform a canonical analysis and determine the nature of the stationary point. Let the stationary point be denoted by  $\bar{\mathbf{x}}$ . Let  $\mathbf{H}(\mathbf{x})$  denote the Hessian of the predicted quadratic response function. Let  $\lambda_1, \lambda_2, \dots, \lambda_k$  be the  $k$  eigenvalues of the Hessian. Let  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$  be the  $k$  normalized eigenvectors corresponding to the eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_k$ . Also, let  $\mathbf{Q} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$ . Let  $\Lambda = \text{diag}.[\lambda_1, \lambda_2, \dots, \lambda_k]$ . Finally, let  $J_+ = \{j: \lambda_j > 0\}$ .

If the Hessian is negative definite (ND) or negative semi-definite (NSD), go to Step 3.

If the Hessian is positive definite (PD) or positive semi-definite (PSD), go to Step 7.

If the Hessian is indefinite (ID), go to Step 8.

#### Step 7.

If the Hessian is PD or PSD (that is, the stationary point is a minimum), then follow Step 8 if the following two conditions hold: (a) if the stationary point lies outside the design region, or (b) if the condition number of the Hessian (ratio of maximum to the minimum eigenvalue of the Hessian) is greater than a prespecified number (determined by the modeler).

If the stationary point is a minimum and lies inside the design region, and the condition number of the Hessian is less than a prespecified number, then use the following procedure.

If  $\mathbf{H}(\mathbf{x})$  is PD then the descent direction  $\mathbf{d}$  is given by

$$\mathbf{d} = -\mathbf{Q}\Lambda^{-1}\mathbf{Q}'\nabla f(\bar{\mathbf{x}}) = -\sum_{j=1}^k (v_j) \left[ \frac{v_j' \nabla f(\bar{\mathbf{x}})}{\lambda_j} \right] = -\mathbf{H}^{-1} \nabla f(\bar{\mathbf{x}}). \quad (5.1)$$

Note that

$$\nabla f(\bar{\mathbf{x}})' \mathbf{d} = -\sum_{j=1}^k \frac{(\nabla f(\bar{\mathbf{x}})' v_j)^2}{\lambda_j} < 0 \text{ if } \nabla f(\bar{\mathbf{x}}) \neq 0. \quad (5.2)$$

Accept the most favorable point along this search as the minimum. If no improvement is observed along this search, then report the design point with the most favorable response value in the current design as the optimum. Go to Step 9.

If the Hessian is PSD, then for  $J_+ = \{j: \lambda_j > 0\}$ , we define

$$\mathbf{C} = \sum_{j \in J_+} \frac{v_j v_j'}{\lambda_j}, \quad (5.3)$$

so that

$$\mathbf{d} = -\mathbf{C} \nabla f(\bar{\mathbf{x}}), \quad (5.4)$$

and

$$\mathbf{d}'\nabla f(\bar{\mathbf{x}}) = -\nabla f(\bar{\mathbf{x}})' \mathbf{C} \nabla f(\bar{\mathbf{x}}) \leq 0, \text{ since } \mathbf{C} \text{ is PD or PSD.} \quad (5.5)$$

If  $\mathbf{d}'\nabla f(\bar{\mathbf{x}}) = 0$  in (5.5), then let  $\mathbf{d} = -\nabla f(\bar{\mathbf{x}})$ .

Accept the most favorable point along the search with direction  $\mathbf{d}$  as the minimum. If no improvement is observed along this search, then report the design point with the most favorable response value in the current design as the optimum. Go to Step 9.

Step 8.

For  $J_+ = \{j: \lambda_j > 0\} \neq \phi$ , we define

$$\mathbf{E} = \sum_{j \in J_+} \frac{\mathbf{v}_j \mathbf{v}_j'}{\lambda_j}, \quad (5.6)$$

so that

$$\mathbf{d} = -\mathbf{E} \nabla f(\bar{\mathbf{x}}), \quad (5.7)$$

and

$$\mathbf{d}'\nabla f(\bar{\mathbf{x}}) = -\nabla f(\bar{\mathbf{x}})' \mathbf{E} \nabla f(\bar{\mathbf{x}}) \leq 0, \text{ since } \mathbf{E} \text{ is PSD or PD.} \quad (5.8)$$

If  $\nabla f(\bar{\mathbf{x}})' \mathbf{d} < 0$ , then  $\mathbf{d}$  is the descent direction. Else, if  $J_+ = \phi$ , or if  $\mathbf{d}'\nabla f(\bar{\mathbf{x}}) = 0$  in (5.8), then let  $\mathbf{d} = -\nabla f(\bar{\mathbf{x}})$ .

Explore the response surface along the descent direction described above. If this search yields a better response that lies outside the current design region, then select the most favorable design point as the center of a new design, and go to Step 2. If responses

along this direction are non-improving, then accept the best response as optimal and stop the algorithm. Go to Step 9.

#### Step 9.

Stop the algorithm.

Apart from the direction of search followed by this algorithm, there is a key difference in this modified algorithm from the one presented in Section 3.1. The difference is in the decision made after the second-order model is found to adequately represent the response function. The original algorithm stops at this step after reporting the most favorable design point obtained using the second-order model, whereas the modified algorithm allows the search to be continued using more designs, if needed. The gains achieved using this aspect of the modification are discussed in the next chapter using a numerical example.

In Step 3 of the algorithm above, it is prescribed that the analyst follow an appropriate gradient deflection direction after the first pass of that step. The gradient deflection directions tested for the purpose of this work are discussed next.

## ***5.2 Gradient Deflection Methods***

The idea of conjugacy and gradient deflection techniques are discussed in Section 2.6. In this section the gradient deflection methods that are used in Step 3 of the modified algorithm for the purpose of this study are specified.

Notice that just before Step 3 of the algorithm, the gradient of the objective function at the  $j$ th ( $j = 2, 3, \dots$ ) iterate is estimated. For the first pass of the algorithm, the search direction followed is the path of steepest descent. From the second pass onwards

through Step 3, instead of following the negative gradient (for minimization problems) as in the path of steepest descent, this gradient is deflected and the deflected direction is used to conduct the search for better responses. This work compares the use of steepest descent to four different gradient deflection methods in the context of RSM. Empirical studies on these different techniques may help to identify their relative performance. This is one of the goals of this study. The four gradient deflection strategies are briefly presented next.

Recall from Section 2.6 that in gradient deflection methods, or, the method of conjugate gradients, a sequence of points  $\mathbf{x}_{j+1}$ , and a sequence of directions  $\mathbf{d}_j$ , are generated iteratively as

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \lambda_j \mathbf{d}_j, \quad (5.9)$$

and

$$\mathbf{d}_j = -\mathbf{g}_j + \kappa_j \mathbf{d}_{j-1}, \quad (5.10)$$

where  $\kappa_j$  is a multiplier which scales the direction vector of the previous iteration, and is calculated differently under different conjugate gradient methods,  $\mathbf{g}_j$  is the gradient of the objective function at the operating point  $\mathbf{x}_j$ ,  $\lambda_j$  is the step length along  $\mathbf{d}_j$  which minimizes  $f(\mathbf{x})$  along this direction. As seen from (2.72) a conjugate gradient direction at the  $k$ th iterate,  $\mathbf{d}_k$ , is formed from the linear combination of the negative gradient at  $k$ th iteration and the direction vector of the previous iterate. This implies a possible advantage of this method over the method of steepest descent.

The first two gradient deflection methods use different values of  $\kappa_j$  ( $j = 2, 3, \dots$ ) at the  $j$ th iterate. In particular, the first gradient deflection method proposed by Sherali and

Ulular (1990) uses the deflected direction given in (2.76). We denote it as GD1. Therefore, under GD1 the deflected direction at the  $j$ th iterate is given by

$$\mathbf{d}_j = -\mathbf{g}_j + \frac{\|\mathbf{g}_j\|}{\|\mathbf{d}_{j-1}\|} \mathbf{d}_{j-1}, \quad (5.11)$$

where  $\mathbf{d}_{j-1}$  is the search direction followed at the  $(j-1)$ st iterate, with  $\mathbf{d}_0 = -\mathbf{g}_0$ .

The second method used in this study is the one provided by Sherali-Ulular (1990) which uses  $\kappa_j^{SV}$  defined in (2.86). This method is denoted by GD2, and the deflected direction under GD2 at the  $j$ th iterate is given by

$$\mathbf{d}_j = -\mathbf{g}_j + \frac{\mathbf{q}'_j \mathbf{g}_j - (1/s_j) \mathbf{p}'_j \mathbf{g}_j}{\mathbf{q}'_j \mathbf{d}_{j-1}} \mathbf{d}_{j-1}, \quad (5.12)$$

where  $s_j$ ,  $\mathbf{q}_j$ , and  $\mathbf{p}_j$  are defined in (2.85), (2.61), and (2.62) respectively.

The third gradient deflection method uses the memoryless BFGS update given by (2.84), so that the direction at the  $j$ th ( $j = 2, 3, \dots$ ) iterate is given by

$$\mathbf{d}_j = -\left[ \mathbf{I} - \frac{\mathbf{p}_j \mathbf{q}'_j + \mathbf{q}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j} + \left[ 1 + \frac{\mathbf{q}'_j \mathbf{q}_j}{\mathbf{q}'_j \mathbf{p}_j} \right] \frac{\mathbf{p}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j} \right] \mathbf{g}_j. \quad (5.13)$$

Finally, the fourth gradient deflection method is the modification of the memoryless BFGS update provided by Sherali-Ulular (1990), and given in (2.89).

$$\mathbf{d}_j = -\left[ \mathbf{I} - \frac{\mathbf{p}_j \mathbf{q}'_j + \mathbf{q}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j} + \left[ \frac{1}{s_j} + \frac{\mathbf{q}'_j \mathbf{q}_j}{\mathbf{q}'_j \mathbf{p}_j} \right] \frac{\mathbf{p}_j \mathbf{p}'_j}{\mathbf{q}'_j \mathbf{p}_j} \right] \mathbf{g}_j. \quad (5.14)$$

The four gradient deflection methods will be used to perform the empirical study. As suggested in the literature, restarting employed in conjunction with gradient deflection methods enhance the performance of these methods. We next specify the restarting procedures employed for the purpose of this study.

Restarting procedures are discussed in Section 2.6.3. Two restarting criteria will be examined under the four gradient deflection methods. An effective choice of restarting criteria for the gradient deflection methods can then be made. The two restarting criteria are denoted by RSA and RSB.

Under criterion RSA, for a  $k$ -variable problem, the optimization procedure is restarted at the  $j$ th iteration, that is,  $\mathbf{d}_j = -\mathbf{g}_j$ , if any of the following three conditions are satisfied:

1. if  $j = k$ , or
2. if  $\|\mathbf{g}'_j \mathbf{g}_{j+1}\| \geq 0.2\|\mathbf{g}_j\|^2$  holds, or
3. if  $-1.2\|\mathbf{g}_j\|^2 \leq \mathbf{d}'_j \mathbf{g}_j \leq -0.8\|\mathbf{g}_j\|^2$  is violated.

The rationale for these conditions is provided in Section 2.6.2.

Under criterion RSB, for a  $k$ -variable problem, the optimization procedure is restarted at the  $j$ th iteration if any of the following two conditions are satisfied:

1. if  $j = k$ , or
2. if  $\mathbf{d}'_j \mathbf{g}_j \geq -0.8\|\mathbf{g}_j\|^2$ .

Conditions in RSB are a subset of conditions in RSA. These restarting criteria are empirically tested in the next chapter.

This chapter presented a novel RSM algorithm and discussed the different gradient deflection methods that add novelty to the algorithm. The next chapter provides a quantitative assessment of the new RSM algorithm, and attempts to draw conclusions regarding relative performance of the various gradient deflection methods in the context of RSM. Recommendations to allocate various restarting policies to the different gradient deflection methods will also be suggested.

## CHAPTER VI Example

This chapter provides computational results of the application of the modified RSM algorithm to some standard test problems in the literature. Relative performance of the various gradient deflection methods (gradient deflection methods are discussed in Section 2.6) in the context of RSM is also measured. This study was started with ten test functions compiled in Sherali and Ulular (1990). The test functions with their starting solutions denoted by  $\mathbf{x}^0$  are as follows:

1. Witte and Holst's Strait Function:

$$f(x) = (x_2 - x_1^2)^2 + 100(1 - x_1)^2, \quad \mathbf{x}^0 = (-1.2, 1.0).$$

2. Witte and Holst's Cube Function:

$$f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2, \quad \mathbf{x}^0 = (-1.2, 1.0).$$

3. C. F. Wood's Function:

$$\begin{aligned} f(x) = & 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1(x_2 - 1)^2 + \\ & 10.1(x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1), \\ \mathbf{x}^0 = & (-3.0, -1.0, -3.0, -1.0). \end{aligned}$$

4. Powell's Function:

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - 2x_4)^4,$$

$$\mathbf{x}^0 = (-3.0, -1.0, 0.0, 1.0).$$

5. Witte and Holst's Shallow Function:

$$f(x) = (x_2 - x_1^2)^2 + (x_1 - 1)^2, \quad \mathbf{x}^0 = (-3.0, -1.0).$$

6. Rosenbrock's Function:

$$f(x) = \sum_{i=2}^n 100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2, \quad \mathbf{x}^0 = (-1.2, 1.0, -1.2, 1.0).$$

7. Oren's Power Function:

$$f(x) = (\mathbf{x}'\mathbf{A}\mathbf{x})^2, \quad \mathbf{A} = \text{diag}(1, 2, \dots, n), \quad \mathbf{x}^0 = (1, 1, \dots, 1).$$

8. Fletcher and Powell's Trigonometric Function:

$$f(x) = 100\{[x_3 - 10\theta(x_1, x_2)]^2 + [r(x_1, x_2) - 1]^2\} + x_3,$$

where

$$2\pi\theta(x_1, x_2) = \begin{cases} \arctan\left(\frac{x_2}{x_1}\right) & \text{if } x_1 > 0 \\ \pi + \arctan\left(\frac{x_2}{x_1}\right) & \text{if } x_1 < 0 \end{cases}$$

and

$$r(x_1, x_2) = (x_1^2 + x_2^2)^{1/2}, \quad \mathbf{x}^0 = (-1, 0, 0)$$

9. Watson's Function:

$$f(x) = \sum_{i=1}^{30} \left\{ \sum_{j=1}^n (j-1)x_j y_i^{j-2} - \left( \sum_{j=1}^n x_j y_i^{j-1} \right)^2 - 1 \right\}^2 + x_1^2,$$

where

$$y_i = \frac{i-1}{29}, \mathbf{x}^0 = (0,0,\dots,0)$$

10. Mancino's Function:

$$f(x) = \sum_{i=1}^n f_i^2,$$

where

$$f_i = \sum_{j=1}^n \left[ \left( x_j^2 + \frac{i}{j} \right)^{1/2} \left( \sin^\alpha \log \left( x_j^2 + \frac{i}{j} \right)^{1/2} + \cos^\alpha \log \left( x_j^2 + \frac{i}{j} \right)^{1/2} \right) \right] + \beta n x_i + \left( i - \frac{n}{2} \right)^\gamma,$$

and

$$\alpha = 5, \beta = 14, \gamma = 3, \mathbf{x}^0 = (af_1(0), \dots, af_n(0)),$$

with

$$a = \frac{\beta n}{\beta^2 n^2 - (\alpha + 1)^2 (n - 1)^2}$$

The computational results for the above ten response functions showing the best solutions obtained using the different search strategies in the context of RSM along with the number of simulation runs needed to reach these solutions are provided in Table 3. Interesting results are exhibited by solutions to test functions 3, 6, and 8. Observe from Table 4, that for function # 3, the best optimal objective function values obtained using the path of steepest descent, GD1, GD2, GD3, and GD4 are 16.93, 2.55, 128.76, 2.76, and 1.61 respectively. This illustrates better performance of the RSM algorithm using GD1, GD3, and GD4 than using the path of steepest descent. RSM with GD2 has performed worse than RSM with the path of steepest descent. Similar results are exhibited by Function # 6 (Table 5). For Function # 8 (Table 6), RSM with all gradient deflection methods have performed worse than that with the path of steepest descent. For the rest of the test functions, on an average, the RSM algorithm using gradient deflection methods did not perform any better or worse than that using the path of steepest descent. The focus for the remainder of this empirical study is therefore restricted to test functions 3, 6, and 8.

To improve the performance of the RSM algorithm using gradient deflection methods, two different restarting criteria, RSA, and RSB, as discussed in Section 5.2 were employed. The results for these two criteria along with the results when no restarting was used, are tabulated in Tables 4, 5, and 6 for response functions 3, 6, and 8 respectively. These tables present the best solution obtained under each method for the three test functions, the number of simulation runs needed to achieve it, and also the number of first and second order models employed along the search. From these results we observe that GD2 performs better with RSA and GD1, GD3, and GD4 work well with RSB. This allocation of the restarting criteria to gradient deflection methods is therefore followed for the remainder of this study.

Table 3. Performance of test functions under different methods of RSM.

Optimal objective function value for all functions is 0.0

Function #	P.O.S.D	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)	0.001 (56)	0.005 (55)	0.77 (26)	0.75 (26)
6	6.47 (102)	3.73 (174)	130.04 (63)	5.47 (87)	3.44 (87)
7	0.0014 (101)	0.0011 (123)	0.0001 (102)	0.002 (83)	0.002 (83)
8	0.06 (156)	18.38 (86)	2.25 (131)	2.64 (90)	2.63 (119)
9	2.94 (40)	2.5 (68)	2.41 (68)	2.94 (40)	2.94 (40)
10	3.0 (19)	3.0 (19)	3.0 (19)	3.0 (19)	3.0 (19)

Figures represent the best solutions obtained.

Figures in parentheses represent the number of simulation runs required.

Table 4. Performance of test function # 3 under different restarting conditions.

	Restarting Conditions			
		RSA	RSB	No Restarting
GD1	Best Solution	4.95	2.55	2.55
	# of runs	94	100	100
	# of 1st order models	3	3	3
	# of 2nd order models	1	1	1
GD2	Best Solution	35.09	128.76	128.78
	# of runs	90	69	69
	# of 1st order models	4	3	3
	# of 2nd order models	0	0	0
GD3	Best Solution	35.29	7.58	2.76
	# of runs	91	190	200
	# of 1st order models	4	9	8
	# of 2nd order models	0	0	1
GD4	Best Solution	35.04	11.93	9.59
	# of runs	91	157	165
	# of 1st order models	4	6	7
	# of 2nd order models	0	1	1

Table 5. Performance of test function # 6 under different restarting conditions.

	Restarting Conditions			
		RSA	RSB	No Restarting
GD1	Best Solution	2.99	5.68	3.73
	# of runs	171	179	174
	# of 1st order models	7	9	8
	# of 2nd order models	1	0	1
GD2	Best Solution	2.97	130.04	130.04
	# of runs	129	63	63
	# of 1st order models	5	4	4
	# of 2nd order models	1	0	0
GD3	Best Solution	2.99	2.62	5.47
	# of runs	131	132	106
	# of 1st order models	4	4	5
	# of 2nd order models	1	1	0
GD4	Best Solution	2.99	2.62	3.44
	# of runs	131	132	105
	# of 1st order models	4	4	5
	# of 2nd order models	1	1	0

Table 6. Performance of test function # 8 under different restarting conditions.

	Restarting Conditions			
		RSA	RSB	No Restarting
GD1	Best Solution	7.51	8.61	18.38
	# of runs	70	90	86
	# of 1st order models	4	5	6
	# of 2nd order models	1	1	0
GD2	Best Solution	0.06	18.77	3.25
	# of runs	137	82	131
	# of 1st order models	3	6	5
	# of 2nd order models	5	0	1
GD3	Best Solution	0.03	0.02	0.04
	# of runs	136	173	90
	# of 1st order models	4	4	2
	# of 2nd order models	4	6	3
GD4	Best Solution	0.06	0.019	2.63
	# of runs	140	191	118
	# of 1st order models	3	4	4
	# of 2nd order models	5	7	1

A factor of interest for function # 8 is the number of second-order models employed under the modified RSM algorithm. Recall from Section 3.1, that the original 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 which allows the search to be continued using more designs, if needed, *after* the data has being adequately represented by a second-order model at any stage of the algorithm, improves the mean response of the best design point reported. For example, GD3 uses six second-order models to reach the best response function value of 0.02, with the true optimal objective function value being 0. If the original algorithm were used in conjunction with GD3, and only one second-order model employed, then the best response function value obtained would be 6.78, thus illustrating the improvement of the modified RSM algorithm over the standard existing one.

The three test functions are next subject to randomness and the RSM algorithm is implemented under the path of steepest descent and four gradient deflection methods. At each design, the response functions are subjected to a random error term which is normally distributed with mean 0, and variance equal to 10 percent of the *true* response at the center of the design. Two independent replications are performed at each design point and the average of these two responses is recorded as the mean response at that design point. Independent replications are also performed *across* design points. Along the gradient search, the variance on the error term of the responses is the same as that used for the first-order design. For all the test functions, the uncoded values of the variables for the first-order model are set at +0.1 and -0.1 as the high and low levels respectively. The axial points for the second-order model are selected so as to make the design rotatable ( see Section 2.5 for a discussion on rotatable designs). Functions 3 and

6 were chosen to have dimension, four. Computational results are presented for the three test functions under randomness in Tables 7, 8, and 9 for test functions 3, 6, and 8 respectively.

For Function # 3, GD1 performs considerably better than any of the other methods. Also, GD2, GD3, and GD4 do not perform as well as the path of steepest descent. For Function # 6, GD1 is again the winner among all the methods employed. GD4 and the path of steepest descent also perform competitively. Notice that GD2 fails to reach close to the optimal for replication # 2, and GD3 does the same for replication # 3. For Function # 8, all the methods seem to perform reasonably well in terms of the best solution obtained. GD1 however requires more simulation runs than any of the other methods. From Tables 3 and 9, it is observed that results that were clearly in favor of the path of steepest descent for function # 8 in the deterministic case are however not visible when randomness is incorporated into these functions.

It can thus be concluded that randomness does affect the use of gradient deflection methods to a certain degree. Tables 7, 8, and 9 also suggest that under randomness, GD1 seems to be performing the best among all methods. Its use is therefore suggested for future use in RSM optimization problems subjected to randomness.

This chapter has illustrated the use of the modified RSM algorithm. The improvement using certain gradient deflection methods over the conventional method of steepest descent in the context of RSM are evident. This empirical study also conjectures on the use of GD1 over other gradient deflection methods in the context of RSM when response functions are subject to randomness. The novel RSM algorithm presented in Chapter 5 and its numerical illustrations via Chapter 6 therefore add a new dimension to the search procedures currently being used in RSM, and hence seems to be a useful

Table 7. Performance of test function # 3 under randomness.

Replication		POSD	GD1	GD2	GD3	GD4
1	Best solution	34.98	2.04	113	113	113
	# of runs	70	99	48	48	48
2	Best solution	33.11	7.88	33.11	33.11	33.11
	# of runs	59	90	59	59	59
3	Best solution	31.85	2.52	31.85	31.85	31.85
	# of runs	62	85	62	62	62
Average	Best solution	33.31	4.15	59.32	59.32	59.32
	# of runs	64	91.3	56.3	56.3	56.3

Table 8. Performance of test function # 6 under randomness.

Replication		POSD	GD1	GD2	GD3	GD4
1	Best solution	3.23	1.16	2.9	2.58	2.68
	# of runs	134	131	151	112	142
2	Best solution	4.64	1.76	144	2.52	6.62
	# of runs	141	132	41	105	107
3	Best solution	3.97	2.5	3.62	40.07	1.71
	# of runs	141	130	153	90	106
Average	Best solution	3.95	1.81	50.17	15.06	3.67
	# of runs	138.67	131	115	102.3	118.3

Table 9. Performance of test function # 8 under randomness.

Replication		POSD	GD1	GD2	GD3	GD4
1	Best solution	6.44	6.94	5.84	5.69	7.3
	# of runs	77	120	94	88	64
2	Best solution	6.4	6.25	6.25	5.0	-0.5
	# of runs	79	98	79	89	89
3	Best solution	5.71	6.72	5.71	5.71	5.71
	# of runs	59	90	59	59	59
Average	Best solution	6.18	6.64	5.93	5.47	4.17
	# of runs	71.67	102.67	77.3	78.67	70.67

contribution to the existing literature. The next chapter provides a summary of this dissertation and suggests avenues and recommendations for future research.

## CHAPTER VII Conclusions and Future Research

This chapter summarizes the research conducted in this dissertation and provides recommendations and avenues for future research.

This dissertation provides the following: (a) an RSM algorithm which uses the CRN correlation-induction strategy for sequential designs, (b) a novel RSM algorithm, and (c) empirical studies to show the gains of using (a) and (b) over the existing procedures. Chapter 3 describes the application of the CRN strategy to a general RSM algorithm. In addition, it develops analytical results showing the gains of employing the CRN strategy at each stage of the RSM algorithm. Chapter 4 quantifies this study using a job-shop example. Overwhelming gains are observed using the CRN strategy over the IS strategy which is illustrated by numerous performance measures. In particular, the improvement in response value per simulation run for the problem under study is shown to have improved three-fold using CRN strategy over the IS strategy.

Chapter 5 offers a novel RSM algorithm which incorporates the gradient deflection methods with restarting employed, instead of using the path of steepest descent *only*. Chapter 6 illustrates the use of the new algorithm over the old one, by means of a numerical example. For the test functions considered in this work, the average direction strategy (denoted by GD1 in this dissertation) proposed by Sherali and Ulular (1990) is shown to have performed the best and is therefore proposed for use in future studies.

The novel RSM algorithm with the average direction strategy used as the gradient search procedure is thus shown to be superior to the existing implementations of RSM algorithms.

The improvements in the existing implementations of the RSM algorithm achieved using efficient simulation strategies and improved gradient search procedures exhibit encouraging results. The research conducted as a part of this dissertation not only contributes to the existing literature, but also opens many avenues for future research. These are as follows:

1. The RSM algorithms presented in Sections 3.1 and 5.1 assume only first and second order models. Bias considerations under third or higher order models could be considered and incorporated into the study.
2. This work assumes homogeneous variances for the mean responses in a design and also for design points along the gradient. Effect of heterogeneous variances can be incorporated into this study.
3. Certain other simulation-optimization techniques like the Nelder-Mead, or Rosenbrock, etc. can be compared to the RSM algorithm to compare relative efficiencies of the different techniques.
4. The modified algorithm presented in Section 5.1 can be tested under some variance reduction techniques to improve the performance of the algorithm.

## References

1. Arnold, S. F. 1981. *The Theory of Linear Models and Multivariate Analysis*. John Wiley & Sons, New York.
2. Bailey, T. G., K. Bauer, and A. B. Marsh. 1989. Response Surface Analysis of Stochastic Network Performance. *Proceedings of the Winter Simulation Conference: E. A. MacNair, K. J. Musselman, P. Heidelberger (eds.)*, 437-443.
3. Bazaraa, M.S., H. D. Sherali, and C. M. Shetty. 1993. *Nonlinear Programming*. 2nd. Ed. John Wiley & Sons, New York.
4. Beale, E. M. L. 1972. A Derivation of Conjugate Gradients, in *Numerical Methods for Nonlinear Optimization*, F. A. Lootsma, ed., Academic Press, London, pp. 39-43.
5. Biles, W. E. 1973. Constrained Sequential-Block Search in Simulation Experimentation. *Proceedings of the Winter Simulation Conference*, 277-241.
6. Biles, W. E. 1975. A Response Surface Method for Experimental Optimization of Multi-Response Processes. *Industrial Engineering Chemistry, Process Design and Development*,14, 152-158.
7. Biles, W. E. and H. T. Ozmen 1987. Optimization of Simulation Responses in a Multicomputing Environment. *Proceedings of the Winter Simulation Conference: A. Thesen*,
8. Biles, W. E. and J. J. Swain 1979. *Mathematical Programming and the Optimization of Computer Simulations. Mathematical Programming Study*. North Holland Publishing Company , 189-207. H. Grant, W. David Kelton (eds.), 402-408.

9. Box, G. E. P. and N. R. Draper 1987. *Empirical Model-Building and Response Surfaces*. John Wiley & Sons, New York.
10. Box, G. E. P. and K. B. Wilson 1951. On the Experimental Attainment of Optimum Conditions. *Jour. of the Royal Stat. Soc.*, **B13**, 1-38, discussion 38-45.
11. Bratley, P., B. L. Fox, and L. E. Schrage 1983. *A Guide to Simulation*. Springer-Verlag, New York.
12. Carroll, C. W. 1961. The Created Response Surface Technique for Optimizing Nonlinear Restrained Systems. *Operations Research*, **9**, 169-184.
13. Cooley, B. J. and E. C. Houck 1982. A Variance-Reduction Strategy for RSM Simulation Studies. *Decision Sciences*, **13**, 482-492.
14. Daughety, A. F., and M. A. Turnquist 1981. Budget Constrained Optimization of Simulation Models via Estimation of Their Response Surfaces. *Operations Research*, **29**, 485-500.
15. Davies, O. L. 1956. *Design and Analysis of Industrial Experiments*, 2nd ed., Hafner Publishing Co., Inc., New York.
16. Donohue, J. M., E. C. Houck, and R. H. Myers 1992. Simulation Designs for the Estimation of Quadratic Response Surface Functions in the Presence of Model Misspecification. *Management Science*, to appear.
17. Donohue, J. M. 1988. *Use of Correlated Simulation Experiments in Response Surface Optimization*. Ph.D dissertation, Department Of Management Science, Virginia Polytechnic Institute and State University.
18. Draper, N. R., and H. Smith 1981. *Applied Regression Analysis*, John Wiley and Sons.
19. Fletcher, R., and C. M. Reeves 1964. Function Minimization by Conjugate Gradients. *Computer Journal*, **7**, 149-154.

20. Geoffrion, A. M., J. M. Dyer, and A. Feinberg 1972. An Interactive Approach for Multi-Criterion Optimization with an Application to the Operation of an Academic Department. *Management Science* , **19**, 357-368.
21. Graybill, F. A. 1976. *Theory and Application of the Linear Model*. Duxury Press, North Scituate, Massachusetts.
22. Haddock, J., and G. Bengu 1987. Application of a Simulation Optimization System for a Continuous Review Inventory Model. *Proceedings of the Winter Simulation Conference: A. Thesen, H. Grant, W. David Kelton (eds.)*, 382-390.
23. Heller, N. B. and G. E. Staats 1973. Response Surface Optimization when Experimental Factors are Subject to Costs and Constraints. *Technometrics*, **15**, 113-123.
24. Hestenes, M. R. 1980. *Conjugate Direction Methods in Optimization*, Springer-Verlag.
25. Hestenes, M. R. and E. Stiefel 1952. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, **48**, 409-436.
26. Heikes, R. G., D. C. Montgomery, and R. L. Rardin 1976. Using Common Random Numbers in Simulation Experiments - An Approach to Statistical Analysis. *Simulation*, **25**, 81-85.
27. Hooke, R. and T. A. Jeeves 1961. Direct Search Solution of Numerical and Statistical Problems. *Jour. Ass. Compu. Mach.*, **8**, 212-229.
28. Hussey, J. R., R. H. Myers, and E. C. Houck 1987a. Correlated Simulation Experiments in First-Order Response Surface Designs. *Operations Research*, **35**, 744-758.
29. Hussey, J. R., R. H. Myers, and E. C. Houck 1987b. Pseudorandom Number Assignment in Quadratic Response Surface Designs. *IIE Transactions*, **19**, 395-403.
30. Johnson, N. L. and S. Kotz 1970. *Distributions in Statistics: Continuous Univariate Distributions-2*, John Wiley and Sons, New York.

31. Joshi, S. S., and J. D. Tew 1993. Statistical Analysis and Validation Procedures Under the Common Random Number Correlation Induction Strategy for Multi-population Simulation Experiments. *European Journal of Operational Research*, to appear.
32. Khuri, A. I., and J. A. Cornell 1987. *Response Surfaces*, Marcel Dekker, New York.
33. Kleijnen, J. P. C. 1974. *Statistical Techniques in Simulation, Part I*. Marcel Dekker, New York.
34. Kleijnen, J. P. C. 1979. Analysis of Simulation with Common Random Numbers : A Note on Heikes et al. (1976). *Simuletter*, 11, 7-13.
35. Kreyszig, Erwin. 1991. *Advanced Engineering Mathematics*. 5th Ed., Wiley Eastern Limited.
36. Kwon, C. 1991. Combined Correlation Induction Strategies for Designed Simulation Experiments. *Ph.D Dissertation. Virginia Polytechnic Institute and State University, Blacksburg, VA* .
37. Kwon, C. and J. D. Tew. 1993. Strategies for Combining Antithetic Variates and Control Variates in Designed Simulation Experiments. *Management Science*, to appear.
38. Kwon, C. and J. D. Tew. 1993. Combined Correlation Methods for Multipopulation Metamodels. *Journal of Statistical Computer Simulation*, to appear.
39. Lavenberg, S. S., T. L. Moeller, and P. D. Welch 1982. Statistical Results on Control Variables with Application to Queueing Network Simulation. *Operations Research*, 182-202.
40. Lavenberg, S. S. and P. D. Welch 1981. A Perspective on the Use of Control Variables to Increase the Efficiency of Monte Carlo Simulations. *Management Science*, 27, 322-335.
41. Law, A. M. and W. D. Kelton 1991. *Simulation Modeling and Analysis*., 2nd Ed.,

McGraw-Hill, New York.

42. Loganathan, G. V., and H. D. Sherali 1987. A Convergent Interactive Cutting-Plane Algorithm for Multiobjective Optimization. *Operations Research* , **35**, 365-377.
43. Lootsma, F. A. 1971. *Numerical Methods fo Nonlinear Optimization*. Academic Press.
44. Luenberger, D. G. 1984. *Introduction to Linear and Nonlinear Programming*. 2nd Ed., Addison-Wesley, Amsterdam.
45. Montgomery, D. C. and D. M. Evans 1972. Second-Order Response Surface Designs in Computer Simulation. *Simulation*, 169-178.
46. Montgomery, D. C., J. J. Talavage, and C. J. Mullen 1972. A Response Surface Approach to Improving Traffic Signal Settings in a Street Network. *Transportation Research*, **6**, 69-80.
47. Montgomery, D. C. and V. M. Bettencourt 1977. Multiple Response Surface Methods in Computer Simulation. *Simulation*, 113-121.
48. Morrison, D. F. 1976. *Multivariate Statistical Methods*. McGraw-Hill, Inc.
49. Myers, R. H. and J. S. Milton 1991. *A First Course in the Theory of Linear Statistical Models*. The Duxbury Advanced Series in Statistics and Decision Sciences. PWS-Kent Publishing Company, Boston.
50. Myers, R. H., A. I. Khuri, and W. H. Carter 1989. Response Surface Methodology: 1966-1988. *Technometrics*, **31**, 137-157.
51. Myers, R. H. 1976. *Response Surface Methodology* .
52. Myers, R. H. and W. H. Carter 1973. Response Surface Techniques for Dual Response Systems. *Technometrics*, **15**, 301-317.
53. Nelson, B. L. 1987. Some Properties of Simulation Interval Estimators under Dependence Induction. *Operations Research Letters*, **6**, 169-176.

54. Neter, J., W. Wasserman, and M. Kutner 1989. *Applied Linear Regression*. Irwin - Homewood, Ill. .
55. Nozari, A., S. F. Arnold, and C. D. Pegden 1987. Statistical Analysis for use with the Schruben-Margolin correlation induction strategy. *Operations Research* ,**35**, 127-138.
56. Nozari, A., S. F. Arnold, and C. D. Pegden 1984. Control Variates for Multipopulation Simulation Experiments. *IIE Transactions*, 159-169.
57. Perry, A. 1978. A Modified Conjugate Gradient Algorithm. *Operations Research* , **26**, 1073-1078.
58. Powell, M. J. D. 1977. Restart Procedures for the Conjugate Gradient Method. *Mathematical Programming*, **12**, 241-254.
59. Powell, M. J. D. 1964. An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives. *Computer Journal*, **7**, 155-162.
60. Pritsker, A. A. B. 1986. *Introduction to Simulation and SLAM II*, 3rd. Ed., Halsted Press, New York.
61. Rand Corporation. 1955. *A Million Random Digits with 100,000 Normal Deviates*. The Free Press, Publishers, Glencoe, Illinois.
62. Rubinstein, R. Y. and R. Marcus 1985. Efficiency of Multivariate Control Variates in Monte Carlo Simulation. *Operations Research*, **33**, 661-677.
63. Rustagi, J. S. 1981. Optimizing Methods in Simulation. *Technical Report No.239*, Dept. of Statistics, Ohio State University.
64. Schruben, L. W. and B. H. Margolin, B. H. 1978. Pseudorandom Number Assignment in Statistically Designed Simulation and Distribution Sampling Experiments. *Journal of the American Statistical Association* **73**, 504-525.
65. Seber, G. A. F. 1977. *Linear Regression Analysis.*, John Wiley & Sons, New York.

66. Shanno, D. F. 1978. Conjugate Gradient Methods with Inexact Searches. *Math. of Oper. Res.* , 3, 244-256.
67. Sherali, H. D., and O. Ulular 1990. Conjugate Gradient Methods Using Quasi-Newton Updates with Inexact Line Searches. *Journal of Mathematical Analysis and Applications*, 150, 359-377.
68. Sommerville, D. M. Y. 1958. *An Introduction to the Geometry of N Dimensions*, Dover Publications, Inc., New York.
69. Tew, J. D. 1989. Correlation Induction Techniques for Fitting Second-order Metamodels in Simulation Experiments. *Proceedings of the Winter Simulation Conference*, 538-546.
70. Tew, J. D. and J. R. Wilson 1992. Validation Of Simulation Analysis Methods for the Schruben-Margolin correlation-induction strategy, *Operations Research*, 40, 87-103.
71. Wilson, J. R. 1984. Variance Reduction Techniques for Digital Simulation. *American Jour. of Math. and Mgt. Sciences* , 4, 277-312.
72. Wilson, J. R. and A. A. B. Pritsker 1984. Variance Reduction in Queueing Simulation Using Generalized Concomitant Variables. *Journal of Statistics and Computer Simulation*, 19, 129-153.
73. Wilson, J. R. and A. A. B. Pritsker 1984. Experimental Evaluation of Variance Reduction Techniques for Queueing Simulation Using Generalized Concomitant Variables. *Management Science*, 12, 1459-1472.
74. Wu, S. M. 1964. Tool-Life Testing by Response Surface Methodology. *Transactions of the ASME*. 105-109.

# Appendix 1

## SLAM II Code for Job-Shop Simulation Model

```
PROGRAM MAIN
DIMENSION NSET(10000)
COMMON QSET(10000)
COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
*MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
*SSL(100),TNEXT,TNOW,XX(100),TBA,PT,INDEXX(6)
EQUIVALENCE (NSET(1),QSET(1))
NNSET = 10000
NCRDR = 5
NPRNT = 6
NTAPE = 7
CALL SLAM
STOP
END
```

```
SUBROUTINE EVENT(I)
GO TO (1,2),I
```

```
C
1 CALL ARRIV
RETURN
```

```
C
2 CALL ENDSV
RETURN
END
```

```
SUBROUTINE INTLC
COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
*MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
*SSL(100),TNEXT,TNOW,XX(100),TBA,PT,INDEXX(6)
COMMON/UCOM1/ XJINF(3000),AAJOB(15),JOBF,NMG,NATT,SAFET,NLATE,
*NUMJOB
```

```
C
C***** XX(MACH) IS BUSY STATUS OF MACHINE MACH
```

```
C
XX(7) = 0.0
```

```

XX(8) = 0.0
XX(9) = 0.0
NMG = 6
NATT = 2*NMG + 3
SAFET = 50.
DO 10 MACH = 1,NMG
10 XX(MACH) = 0.0
C
C***** SET UP THE AUXILLARY ATTRIBUTE ARRAY AND SCHEDULE THE
C***** FIRST ARRIVAL
C
  CALL SETAA(NATT,XJINF,JOBF,3000)
  CALL SCHDL(1,0.0,ATRI)
  NUMJOB = 0
  NLATE = 0
  RETURN
  END
C
C*****
C
  SUBROUTINE ARRIV
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
  *MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
  *SSL(100),TNEXT,TNOW,XX(100),TBA,PT,INDEXX(6)
  COMMON/UCOM1/ XJINF(3000),AAJOB(15),JOBF,NMG,NATT,SAFET,NLATE,
  *NUMJOB
C
C*** SCHEDULE NEXT JOB ARRIVAL
C
  TBA = 56.
  PT = 16.59
  IDT = EXPON(TBA,1)
  ATRIB(4) = EXPON(PT,4)
  ATRIB(5) = EXPON(PT,4)
  ATRIB(6) = EXPON(PT,4)
  ATRIB(7) = EXPON(PT,4)
  ATRIB(8) = EXPON(PT,4)
  ATRIB(9) = EXPON(PT,4)
  DT = MAX0(IDT,1)
  CALL SCHDL(1,DT,ATRI)
C
C*** SET UP ATTRIBUTES AND CHARACTERISTICS OF THIS JOB
C
  CALL DESJOB(AAJOB)
  ATRIB(1) = JOBF
C
C*** STORE ROUTE IN JOB INFORMATION ARRAY 'XJINF'
C
  CALL PUTAA(NATT,XJINF,JOBF,AAJOB)
  ATRIB(3) = DDM(AAJOB)
  ATRIB(2) = AAJOB(NMG + 1) + 1000.

```

```

    IF(ATTRIB(3).LE.TNOW) ATTRIB(2) = AAJOB(NMG+1)
C
C*** DETERMINE DISPOSITION OF JOB
C
    MACH = AAJOB(1)
    IF(XX(MACH).GT.0.0) GO TO 10
C
C*** SET MACHINE BUSY. SCHEDULE END OF SERVICE
C
    XX(MACH) = 1.0
    ETIME = AAJOB(NMG+1)
    CALL SCHEM(MACH,ETIME)
    RETURN
C
C*** PUT JOB IN QUEUE FOR MACHINE
C
10 CALL FILEM(MACH,ATTRIB)
    RETURN
    END

C*****

    SUBROUTINE DESJOB(A)
    COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
    *MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
    *SSL(100),TNEXT,TNOW,XX(100),TBA,PT,INDEXX(6)
    COMMON/UCOM1/ XJINF(3000),AAJOB(15),JOBF,NMG,NATT,SAFET,NLATE,
    *NUMJOB
    DIMENSION A(15),MA(6)
    NOPS = RNORM(4.0,1.0,2) + 0.5
    IF (NOPS.LT.3) NOPS = 3
    IF (NOPS.GT.6) NOPS = 6
    DO 30 I=1,NATT
30 A(I)=0.0
C
C*** SAMPLE TO SET MACHINE ROUTE WITHOUT REPLACEMENT
C
    DO 35 I=1,NMG
35 MA(I) = I
    TOP = NMG + 1
    SUM = 0.0
    INDEXX(1) = UNFRM(1.,7.,3)
    INDEXX(2) = UNFRM(2.,7.,3)
    INDEXX(3) = UNFRM(3.,7.,3)
    INDEXX(4) = UNFRM(4.,7.,3)
    INDEXX(5) = UNFRM(5.,7.,3)
    INDEXX(6) = UNFRM(6.,7.,3)
    DO 40 I=1,NOPS
    BOT = I
    IF(I.EQ.1) INDEX = INDEXX(1)

```

```

IF(I.EQ.2) INDEX = INDEXX(2)
IF(I.EQ.3) INDEX = INDEXX(3)
IF(I.EQ.4) INDEX = INDEXX(4)
IF(I.EQ.5) INDEX = INDEXX(5)
IF(I.EQ.6) INDEX = INDEXX(6)
A(I) = MA(INDEX)
MA(INDEX) = MA(I)
IF(INDEX.EQ.1) IETIM = ATRIB(4)
IF(INDEX.EQ.2) IETIM = ATRIB(5)
IF(INDEX.EQ.3) IETIM = ATRIB(6)
IF(INDEX.EQ.4) IETIM = ATRIB(7)
IF(INDEX.EQ.5) IETIM = ATRIB(8)
IF(INDEX.EQ.6) IETIM = ATRIB(9)
A(I+NMG) = MAX0(IETIM,1)
40 SUM = SUM + A(I+NMG)

```

```

C
C*** SET CURRENT OPERATOR NUMBER TO 1

```

```

C
  A(NATT-2) = 1.0

```

```

C
C*** SET ARRIVAL TIME OF JOB TO TNOW

```

```

C
  A(NATT-1) = TNOW

```

```

C
C*** SET DUE DATE TO TWICE THE ESTIMATED PROCESSING TIME

```

```

C
  A(NATT) = 2.*SUM + TNOW
RETURN
END

```

```

C*****

```

```

FUNCTION DDM(A)
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
*MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
*SSL(100),TNEXT,TNOW,XX(100),TBA,PT,INDEXX(6)
COMMON/UCOM1/ XJINF(3000),AAJOB(15),JOBF,NMG,NATT,SAFET,NLATE,
*NUMJOB
DIMENSION A(15)

```

```

C
C*** SET DUE DATE FOR THE JOB

```

```

C
  DDM = A(NATT) - 0.5*(A(NATT)-TNOW)-SAFET
RETURN
END

```

```

C*****

```

```

SUBROUTINE SCHED(MG,ET)
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,

```

```

*MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
*SSL(100),TNEXT,TNOW,XX(100),TBA,PT,INDEXX(6)
SIGMA = 0.3*ET
SERVT = ET + RNORM(0.0,SIGMA,5)
IF(SERVT.LT.0.0) SERVT = 0.0
ATRIB(2) = MG
C
C** UPDATE THE MODIFIED DUE DATE TO NOT INCLUDE ESTIMATED SERVICE
C** TIME OF THE CURRENT OPERATION
C
ATRIB(3) = ATRIB(3) + ET
C
C** SCHEDULE END OF SERVICE
C
CALL SCHDL(2,SERVT,ATRIB)
RETURN
END

C*****

SUBROUTINE ENDSV
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
*MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
*SSL(100),TNEXT,TNOW,XX(100),TBA,PT,INDEXX(6)
COMMON/UCOM1/ XJINF(3000),AAJOB(15),JOBF,NMG,NATT,SAFET,NLATE,
*NUMJOB
JBPTR = ATRIB(1)
C
C*** SAVE MACHINE NUMBER ON WHICH SERVICE ENDED
C
MACHE = ATRIB(2)
C
C*** DETERMINE DISPOSITION OF JOB ON WHICH SERVICE ENDED
C
ICOPN = JBPTR + NATT - 2
IOPN = IFIX(XJINF(ICOPN)) + 1
C
C*** IF NEXT OPERATION IS OPERATION # 7, JOB HAS COMPLETED
C
IF(IOPN.EQ.7) GO TO 50
XJINF(ICOPN) = IOPN
NEWM = XJINF(JBPTR + IOPN)
C
C*** IF NEXT MACHINE IS 0, JOB HAS COMPLETED
C
IF ( NEWM .EQ. 0 ) GO TO 50
C
C*** DETERMINE THE ESTIMATED PROCESSING TIME
C
ET = XJINF(JBPTR + IOPN + NMG)

```

```

IF(XX(NEWM).GT.0.0) GO TO 20
XX(NEWM) = 1.0
CALL SCHEM(NEWM,ET)
GO TO 100
C
C** IF NEW MACHINE IS BUSY, PLACE THE JOB IN QUEUE
C
20 ATRIB(2) = ET + 1000.
IF(ATRIB(3).LE.TNOW) ATRIB(2) = ET
CALL FILEM(NEWM,ATRIB)
GO TO 100
C
C** UPDATE COUNTER AND COLLECT STATISTICS FOR COMPLETE JOBS
C
50 NUMJOB = NUMJOB + 1
CALL GETAA(JBPTR,NATT,XJINF,JOBF,AAJOB)
TISYS = TNOW - AAJOB(NATT-1)
XX(7) = TISYS
CALL COLCT(TISYS,1)
TLATE = AAJOB(NATT) - TNOW
XX(8) = XX(7) + 0.5*TBA - 5.0*PT - 0.02*TBA*PT
DUM1 = 20.0*(PT-31.0)*(PT-31.0)
DUM2 = 10.0*(TBA-81.0)*(TBA-81.0)
DUM3 = 2.0*(PT-31.0)*(PT-31.0)
DUM4 = 2.0*(TBA-81.0)*(TBA-81.0)
IF ( (PT.GE.31.0).AND.(PT.LE.35.0) ) THEN
IF ( (TBA.GE.81.0).AND.(TBA.LE.100.0) ) THEN
XX(8) = XX(7) + 0.5*TBA - 5.0*PT - 0.02*TBA*PT - DUM1 - DUM2
END IF
END IF
IF ( (PT.GT.31.0).AND.(TBA.GT.100) ) THEN
XX(8) = XX(7) + 0.5*TBA - 5.0*PT - 0.02*TBA*PT - 3930.0 + DUM3 + DUM4
END IF
IF ( (TBA.GT.81.0).AND.(PT.GT.35) ) THEN
XX(8) = XX(7) + 0.5*TBA - 5.0*PT - 0.02*TBA*PT - 3930.0 + DUM3 + DUM4
END IF
XX(9) = XX(9) + 1.0
IF (XX(9).LE.500.0) THEN
WRITE (88,31) XX(8)
31 FORMAT(F12.4)
ENDIF
C CALL COLCT(TLATE,3)
IF(TNOW.LE.AAJOB(NATT)) GO TO 100
NLATE = NLATE + 1
C CALL COLCT(NLATE,2)
C
C*** DETERMINE DISPOSITION OF MACHINE
C
100 IF(NNQ(MACHE).GT.0) GO TO 110
XX(MACHE) = 0.0

```

```

RETURN
C
C*** UPDATE PRIORITY OF JOBS WAITING IN QUEUE OF MACHINE
C
110 CALL UPDAT(MACHE)
    CALL RMOVE(1,MACHE,ATRIB)
    ETIME = ATRIB(2) - 1000.
    IF(ETIME.LE.0.0) ETIME = ATRIB(2)
    CALL SCHES(MACHE,ETIME)
    RETURN
    END

C*****
SUBROUTINE UPDAT(MACH)
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
*MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
*SSL(100),TNEXT,TNOW,XX(100),TBA,PT,INDEXX(6)
DIMENSION NSET(1)
COMMON QSET(1)
EQUIVALENCE (NSET(1),QSET(1))
C
C*** UPDATE THE PRIORITY OF THOSE JOBS IN QUEUE
C
NTRY = MMLE(MACH)
11 IF(NTRY.EQ.0) RETURN
IF(QSET(NTRY+2).LT.1000.) RETURN
IF(QSET(NTRY+3).GT.TNOW) GO TO 15
NEXT = NPRED(NTRY)
QSET(NTRY+2) = QSET(NTRY+2) - 1000.
CALL ULINK(-NTRY,MACH)
CALL LINK(MACH)
NTRY= NEXT
GO TO 11
15 NTRY = NPRED(NTRY)
GO TO 11
END
C *****

GEN,SSJ,JOBSHOP,11/18/1991,1;
LIMITS,6,9,200;
PRIORITY/1,LVF(2)/2,LVF(2)/3,LVF(2)/4,LVF(2)/5,LVF(2)/6,LVF(2);
PRIORITY/NCLNR,HVF(JEVNT);
; ASSIGN,ATRIB(2)
TIMST,XX(1),MACHINE 1 UTIL;
TIMST,XX(2),MACHINE 2 UTIL;
TIMST,XX(3),MACHINE 3 UTIL;
TIMST,XX(4),MACHINE 4 UTIL;
TIMST,XX(5),MACHINE 5 UTIL;
TIMST,XX(6),MACHINE 6 UTIL;
STAT,1,JOB PROC TIME; ,20/100./20.0;
STAT,2,JOB TRDINESS ; ,20/-100./5.0;

```

```
STAT,3,JOB LATENESS;  
INIT,0,60000;  
SEEDS,37191(1),45679(2),68245(3),72637(4),98763(5);  
SIMULATE;  
FIN;
```

## VITA

Shirish Joshi was born on July 27, 1965, in Bombay, India. He received a B.S. in Electrical Engineering from Bombay University in June 1986. He worked as a computer hardware engineer with IDM Ltd., Bombay for one year. He received a M.S. in Mathematical Sciences from Virginia Commonwealth University, Richmond, Virginia in August, 1989. He also received an M.S. in Industrial and Systems Engineering from Virginia Polytechnic Institute and State University, Blacksburg, Virginia in May 1991.