Semi-Parametric Techniques for Multi-Response Optimization

Wen Wan

Dissertation submitted to the faculty of the
Virginia Polytechnic Institute & State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Statistics

Jeffrey B. Birch, Chair
John P. Morgan
Angela N. Patterson
G. Geoffrey Vining
William H. Woodall

October 29th, 2007
Blacksburg, Virginia

Keywords: Desirability Function; Genetic Algorithm (GA); Modified Genetic Algorithm (MGA); Multi-response Optimization (MRO); Response Surface Methodology (RSM); Semiparametric Regression.

Semi-Parametric Techniques for Multi-Response Optimization

Wen Wan

(ABSTRACT)

The multi-response optimization (MRO) problem in response surface methodology (RSM) is quite common in industry and in many other areas of science. During the optimization stage in MRO, the desirability function method, one of the most flexible and popular MRO approaches and which has been utilized in this research, is a highly nonlinear function. Therefore, we have proposed use of a genetic algorithm (GA), a global optimization tool, to help solve the MRO problem. Although a GA is a very powerful optimization tool, it has a computational efficiency problem. To deal with this problem, we have developed an improved GA by incorporating a local directional search into a GA process.

In real life, practitioners usually prefer to identify all of the near-optimal solutions, or all feasible regions, for the desirability function, not just a single or several optimal solutions, because some feasible regions may be more desirable than others based on practical considerations. We have presented a procedure using our improved GA to approximately construct all feasible regions for the desirability function. This method is not limited by the number of factors in the design space.

Before the optimization stage in MRO, appropriate fitted models for each response are required. The parametric approach, a traditional RSM regression technique, which is inflexible and heavily relies on the assumption of well-estimated models for the response of interests, can lead to highly biased estimates and result in miscalculating optimal solutions when the user's model is incorrectly specified. Nonparametric methods have been suggested as an alternative, yet they often result in highly variable estimates, especially for sparse data with a small sample size which are the typical properties of traditional RSM experiments. Therefore, in this research, we have proposed use of model robust regression 2 (MRR2), a semi-parametric method, which combines parametric and nonparametric methods. This combination does combine the advantages from each of the parametric and nonparametric methods and, at the same time, reduces some of the disadvantages inherent in each.

Dedication

To my husband Guimin Gao and my daughter Carolyn Gao for their love, support, encouragement, and patience.

Acknowledgments

Nothing of this magnitude can be completed without the support and help from so many people who surround me. I acknowledge those major supporters here but recognize that there are many others who will remain unnamed due to space and time constraints.

I wholeheartedly acknowledge first the help and support of my advisor, Dr. Jeffrey B. Birch, through this research. He has been a tremendous help and support in giving me many invaluable suggestions and comments, guiding me to keep in a right and efficient way, encouraging me with always a kind word, and keeping my research in a high quality. He has always kept regular meetings with me to help and support me. He has also kept his door open for my many questions even when it may have not been convenient. It seems that he has always known how to train me, guide me, and help me to complete my PhD dissertation and papers and, at the same time, helped me to become a better writer.

I would like to thank Dr. G. Geoffrey Vining for his helpful guidance and suggestions when I started my research on genetic algorithms. I would also like to thank the other members of my committee, Dr. John P. Morgan, Dr. Angela N. Patterson, Dr. William H. Woodall, and my former committee Dr. Dan Spitzner, for their valuable comments and suggestions, and for their time, support, and encouragement.

I would like to express my gratitude to the professors in my department of Statistics for their teaching to make me have a wide interest in the statistical field, for their patience to answer my many statistical questions including silly questions, and for their support and help as a teacher and as a friend. Many thanks also go to the staff and the graduate students of the Virginia Polytechnic Institute & State University Department of Statistics for their support and help in my study and in my life.

I would like to thank my friends in Blacksburg, with whom I have spent great time in the past five years. I would also like to thank my parents, my parents-in-law, my sister Jun Wan, and my relatives in China. They have all been very encouraging and have done their best to support us in this endeavor from long distance.

Many thanks to my beautiful daughter Carolyn for the great happiness she has brought to my life.

Finally, I cannot thank enough my husband, Guimin Gao, for his love and support in many different ways including my life, my study and my research. My love has only grown for him over the last five years.

— Wen Wan

Contents

List of Figures

List of Tables

Glossary of Acronyms

AD	Average Desirability function	98
ASEL	Average Squared Error Loss	98
CCD	Central Composite Design	37
DFDS	Derivative-Free Directional Search method	37
GA	Genetic Algorithm	5
KER	Kernel Regression	13
LLR	Local Linear Regression	15
LPR	Local Polynomial Regression	15
MC	Monte Carlo	37
MGA	Modified Genetic Algorithm	6
MRO	Multi-Response Optimization	1
MRR2	Model Robust Regression 2	3
NR	Newton-Raphson method	37
OLS	Ordinary Least Squares	10
RSM	Response Surface Methodology	1
SD	Method of Steepest Descent	37
SIMSE	Simulated Integrated Mean Squared Error	97

Chapter 1

Introduction

1.1 Multi-Response Problem

In industry and in many other areas of science, data collected often contain several responses (or dependent variables) of interest for a single set of explanatory variables (also called independent variables, controllable variables, factors, regressors, or input variables). It is relatively straightforward to find a setting of the explanatory variables that optimizes a single response. However, it is often hard to find a setting that optimizes multiple responses simultaneously. Thus, a common objective is to find an optimal setting or several feasible settings of the explanatory variables that provides the best compromise of the multiple responses simultaneously. This is called the multiple response problem (Khuri, 1996 and Kim and Lin, 2006). The multiple response problem consists of three stages: data collection (related to experimental design), model building (related to regression techniques), and optimization, specifically called multi-response optimization (MRO). In this research, we assume that the data have been collected and we will focus on the latter two stages—model building techniques and MRO techniques.

1.2 Modeling Techniques in RSM

In response surface methodology (RSM), parametric regression methods are traditionally used to model the data for the response(s), typically, using a low-order polynomial model.

However, in many situations, the parametric model may not adequately represent the true relationship between the explanatory variables and the response(s). This does not mean that the parametric method may not be good for applications, as it does provide the foundation for data modeling in many cases. The problem is that the parametric method may not model well some portions of the mean structure, resulting in the problems caused by model misspecification such as biased estimates of the mean response functions.

An example of model misspecification associated with the parametric method is illustrated by the tensile strength data in Mays, Birch and Starnes (2001), presented in Figure 1.1. Figure 1.1 shows that the raw data reveals a strong peak, a peak of interest to the subject-matter scientist. The data also exhibits a strong quadratic trend and researchers may be satisfied with a second-order polynomial model. However, the second-order polynomial model clearly underfits at the peak of the data so as to suggest that the quadratic model has been misspecified. Consequently, inference from a misspecified parametric regression model may be misleading and the optimization solution(s) may be highly biased.

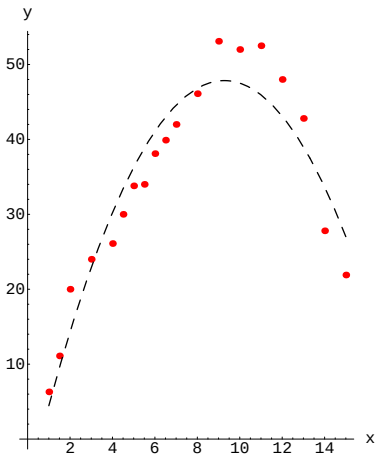

Figure 1.1: Plot of the tensile data with model misspecification by quadratic OLS fits. [••• Raw data and − − − OLS]

When modeling the data parametrically, certain assumptions about the relationship between the explanatory variables and the response(s) must be made. For simplification and ease of interpretation of coefficients, researchers tend to assume the relationship is not very complex and that lower polynomial models provide an appropriate approximation of the true underlying function (or relationship). However, in practical applications, this relationship is not

always so well behaved.

Recently, nonparametric regression techniques have been investigated to address the model misspecification problem associated with the use of parametric regression in the RSM framework. See, for example, papers by Vining and Bohn (1998), Anderson-Cook and Prewitt (2005), Pickle (2006), and Pickle et al. (2006). Nonparametric regression approaches make no assumptions about the parametric relationship between variables. Kernel-based methods use the philosophy that observations closest to the point of interest, $\mathbf{x}_0$, have the most information about the mean response at $\mathbf{x}_0$ while observations farthest from $\mathbf{x}_0$ have the least information, and assign local weights to the observations accordingly. Nonparametric methods can provide superior fits by capturing the structure in the data unable to be captured by a misspecified parametric model.

However, in general, nonparametric approaches depend completely on the data itself without the underling stability of the specified form from the parametric model. Therefore, nonparametric approaches tend to identify mean structure where no structure exists and their fits may be more variable than a parametric fit. Additionally, the successful application of the nonparametric approaches in regression has been limited to those cases with fairly large sample sizes and space-filling designs. But the typical properties of traditional RSM experiments, such as small sample size, typically sparse data, and most of the design points on the edge of design space, may restrict the applications of nonparametric regression in RSM.

Another alternative methodology is to use a semiparametric method which combines the parametric method with the nonparametric methods. One semiparametric method, model robust regression 2 (MRR2) proposed by Mays, Birch and Starnes (2001), was originally developed for situations when there is partial knowledge about the underlying model, a situation very common in applications. MRR2 essentially combines the advantages from the parametric and nonparametric methods and avoids their disadvantages. For the case of a single response, Pickle (2006) and Pickle et al. (2006) have demonstrated that the MRR2 technique can be successfully applied to model the mean response for data from designed experiments. We wish to extend the MRR2 method to the multiple response problem. More details on MRR2 will be discussed in Chapter 2.

One goal of our research is to adapt the MRR2 to the MRO problem in order to reduce both the bias in estimation of mean response due to model misspecification of the user's parametric

model and the high variability in estimation of mean response due to use of nonparametric methods. We will apply the MRR2 to the elementary MRO situation where the random error variance is constant across all responses. We will compare optimal solutions obtained by the parametric, nonparametric, and semiparametric methods to the true optimal solutions.

1.3 Multi-Response Optimization Problems

After the model building stage is completed, where each regression model built for each response is assumed to be appropriate, the optimization stage begins. Several multi-response optimization (MRO) techniques are available that may be used to find an optimal setting or several feasible settings with the best compromise of the multiple responses. The simple and intuitive approach to MRO is to overlay the response contour plots and find the appropriate set of operating conditions for the process by a visual inspection. This method, however, is limited to two or three dimensional domains of explanatory variables. Another method, called the constrained optimization method, is essentially a single response optimization, i.e., the optimization is of the most primary response among the multiple responses with the constraints on the other responses. This method does not directly optimize the multiple responses simultaneously.

One of the most popular and formal approaches is to use some specific function (an objective function) to combine the responses so that the multiple dimensional problem can be transformed into one dimensional problem. There are several popular methods, such as the desirability function method by Derringer and Suich (1980), the generalized distance measure method by Khuri and Conlon (1981), and the weighted squared error loss method by Vining (1998). The desirability function method is one of the most flexible and popular MRO approaches. The generalized distance measure method may be considered as a special case of the square error loss method (Vining, 1998). These two methods take correlation among responses into account. More details on the MRO techniques will be discussed in Chapter 3.

Another problem in the MRO, as mentioned in Montgomery (1999), for a single overall objective function (such as the desirability function) is that there are often multiple optimal solutions. Some of the MRO procedures currently used in practice and implemented in

widely-used computer software do not deal with it very effectively.

Myers et al. (2004) also stated that there may exist several disjoint feasible operating regions for the simultaneous operating process of the multiple responses, resulting in multiple local optima. In applications, practitioners usually prefer to find all of the optimal solutions because some solutions may be more desirable than others based on practical considerations. For example, some of the feasible operating regions which come from the corresponding optimal solutions may be larger than other feasible regions. Large feasible operating regions are desirable as they represent more robust operating conditions found for the process.

In this research, we will investigate the number of available multiple optimal solutions, as determined by the desirability function method. In addition, we will explore use of the genetic algorithm in finding all possible feasible operating regions in high dimensions.

1.4 Genetic Algorithm and Modified Genetic Algorithm

Once the multiple response surfaces have been modelled and once one of the MRO methods has been selected for use, such as the desirability function method, the goal becomes finding the optimal setting(s) of the regressors, based on the MRO method chosen. There are many optimization routines available to use for the MRO problem. For the constrained optimization method with parametric models, some local optimization algorithms are mentioned in Myers et al. (2004), such as the direct search method, the Nelder-Mead simplex method, and the generalized reduced gradient (GRG) method. But these local optimization methods are no longer useful for those highly nonlinear and multi-modal functions such as the desirability function, the generalized distance measure function, and the weighted squared error loss function. Myers et al. (2004) and Carlyle, Montgomery and Runger (2000) recommended use of a heuristic search procedure such as a genetic algorithm to find global optima. Therefore, we will use the genetic algorithm for optimization.

The genetic algorithm (GA), originally developed by Holland (1975), is a stochastic optimization tool whose search technique is based on the Darwinian survival of the fittest principles from biological genetics. Many papers have applied the GA to a broad variety of fields, including ecology, psychology, artificial intelligence and computational mathematics. The

reason that a GA is so popular and useful is that a GA has some attractive features and properties, such as employing multiple concurrent search points (not a single point), not requiring the derivatives of the objective function, using probabilistic transition rules (not deterministic rules), and being able to find a global or near-global optimum from a very complex surface of an objective function, even with very high-dimensional domains of the function. Details on GA will be discussed in Chapter 6.

However, a GA has several disadvantages. One is that the GA is a heuristic search technique and is not theoretically guaranteed to find an optimum or near-optimum. The second is that the efficiency of the GA greatly depends on the choice of selected settings/levels of GA operations from an extremely large set of possibilities. The third one is a computational issue, in that typically the GA, in order to find the optimum, must evaluate an objective function a large number of times. The computational cost is the biggest disadvantage among the three, in that the other two may be ameliorated by increasing the search space and the number of evaluations and by proper choice of levels for each GA operations.

To deal with the computational problem, we will propose and evaluate four versions of a more computationally efficient GA based on modifying a traditional GA. The main idea of each version of the modified GAs (MGAs) is to gather numerical information from the GA itself so that a local directional search may be incorporated into a GA process to make computational improvements. Details on MGAs will be presented in Chapter 5

1.5 Outline of Dissertation

This dissertation is organized as follows. Chapter 2 gives an overview of the current modeling techniques in RSM, including parametric, nonparametric and semiparametric methods. Chapter 3 summarizes the current MRO techniques in RSM. Chapter 4 introduces a genetic algorithm and its basic features. Chapter 5 proposes four different versions of a modified GA and presents results from Monte Carlo simulation studies on comparisons of GA and MGAs. In Chapter 6, based on the stochastic property of the GA/MGA, we use one MGA to find all possible feasible region(s) of the desirability function method, one of the most popular MRO techniques. Chapter 7 extends estimation results from the modeling techniques in the univariate case to the multivariate case. In Chapter 8, our semiparametric approach will be

applied to the MRO problem. Examples from the RSM literature and simulation studies will be used to compare the performance of the modeling techniques. Finally, Chapter 9 gives a summary of our completed work and possibilities for extended future work.

Chapter 2

Current Modeling Techniques in RSM

2.1 Introduction

Many industrial statisticians, engineers, and other researchers use the techniques of RSM. RSM, as described in Myers (1999), is usually viewed in the context of design of experiments (DOE), model fitting, and process optimization. Obviously, model fitting is one of the most important components in RSM.

For the multiple response problem, we may use multivariate regression techniques (which is an extension of multiple linear regression for a single response) to model the relationships between the explanatory variables and the multiple responses simultaneously. But actually, the fits by the regression techniques in the univariate case are equivalent to the fits by the multivariate regression techniques, as discussed in Chapter 7. Therefore, for the multiple response problem considered in this research, we will model each response separately using the modeling techniques for a single response. Details on modeling a single response will be presented in the following sections.

Once the data are collected, our goal is to fit a model to estimate the relationship between the explanatory variables and each response. Suppose the true relationship between the k explanatory variables, $x_{1i}, x_{2i}, ...x_{ki}$, and the response, y_i, is

$$y_i = f(x_{1i}, x_{2i}, ...x_{ki}) + \varepsilon_i, \quad i = 1, ..., n, \tag{2.1}$$

where the function f represents the true relationship, n is the sample size, and ε_i represents a

random error term from the process assumed to be independent, identically distributed, with mean zero and constant variance σ^2. Consequently, $E(y_i|x_{1i}, ..., x_{ki}) = \mu_i = f(x_{1i}, ..., x_{ki})$. That is, $f(x_{1i}, ..., x_{ki})$ is the mean response function.

Usually, the true relationship f is unknown and must be estimated, based on the collected data. The function must be well estimated, otherwise misspecification of the fitted model may have serious implications in process optimization. As mentioned in Chapter 1, the current modeling techniques include the parametric, nonparametric and semiparametric methods. In many situations, the parametric method does not adequately estimate this true relationship, while the nonparametric method is more variable due to completely depending on the data itself. We propose the model robust regression technique (MRR), a semiparametric method, which can improve the estimates of mean response by combining both the parametric and nonparametric results into one set of estimates, simultaneously reducing both bias and variance of estimation. In next section we give details concerning these three modeling methods in RSM.

2.2 Parametric Approach

As stated in Chapter 1, the parametric approach to estimate the relationship between the explanatory variables and the response(s) is to assume that the response surface is relative smooth in a relatively small region of those explanatory variables so that the true mean function f in equation (2.1) can be adequately approximated by a low-order polynomial. In practice, either a first-order or second-order polynomial is widely used in RSM.

A second-order model is given by

$$y_i = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ji} + \sum_{j=1}^{k} \beta_{jj} x_{ji}^2 + \sum_{j<l} \beta_{jl} x_{ji} x_{li} + \varepsilon_i, \tag{2.2}$$

where the β's are the unknown regression coefficients and j and $l = 1, ..., k$. If the β_{jj}'s are all zero, then the second-order model becomes a first-order model with interactions. If the β_{jj}'s and β_{jl}'s are all zero, then the second-order model becomes a first-order model. Given n observations in the data, the second-order model in (2.2) may be expressed in matrix notation as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2.3}$$

where $\mathbf{y}$ is a $n \times 1$ vector of responses, $\mathbf{X}$ is a $n \times \left(1 + 2k + \begin{pmatrix} k \\ 2 \end{pmatrix}\right)$ matrix of regressor data, $\boldsymbol{\beta}$ is a $\left(1 + 2k + \begin{pmatrix} k \\ 2 \end{pmatrix}\right) \times 1$ vector of unknown parameters, and $\boldsymbol{\varepsilon}$ is the $n \times 1$ vector of random errors.

2.2.1 Ordinary Least Squares

Under the assumption that the random error ε_i's have constant variance σ^2, the ordinary least squares method (OLS) is used to obtain the best linear unbiased estimator (BLUE), $\hat{\boldsymbol{\beta}}$, for $\boldsymbol{\beta}$. That is, the OLS estimator has component-wise minimum variance among all linear unbiased estimators. OLS is utilized to seek the estimator for $\boldsymbol{\beta}$ such that the sum of squared errors (SSE) , given as

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i^{OLS})^2, \tag{2.4}$$

is minimized, where $\hat{y}_i^{OLS} = \mathbf{x'}_i\hat{\boldsymbol{\beta}}$ and $\mathbf{x'}_i$ is the ith row of $\mathbf{X}$.

If it is also assumed that the random errors, ε_i's, follow a normal distribution, then the OLS estimator is equivalent to the maximum likelihood estimator (MLE). In addition, the elements of $\hat{\boldsymbol{\beta}}$ under normality have minimum variance among all unbiased estimators. That is, $\hat{\boldsymbol{\beta}}$ is the uniform minimum variance unbiased estimate (UMVUE).

The OLS estimator $\hat{\boldsymbol{\beta}}$ is obtained as:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'y}. \tag{2.5}$$

The estimated responses can be further obtained as:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'y} = \mathbf{H}^{(OLS)}\mathbf{y}, \tag{2.6}$$

where the $n \times n$ matrix $\mathbf{H}^{(OLS)}$ is known as the "HAT" matrix, since the observed y values are transformed into the $\hat{y}$ values through the HAT matrix.

From equation (2.6), the fitted value $\hat{y}_i$ at location $\mathbf{x}_i$ can be written as:

$$\hat{y}_i^{(OLS)} = \sum_{j=1}^{n} h_{ij}^{(OLS)} y_j = \mathbf{h'}_i^{(OLS)}\mathbf{y}, \tag{2.7}$$

where the $h_{ij}^{(OLS)}$ is the i, j^{th} element of the $\mathbf{H}^{(OLS)}$ and the $\mathbf{h}'^{(OLS)}_i$ is the i^{th} row of the $\mathbf{H}^{(OLS)}$. Equation (2.7) shows that the fit $\hat{y}_i^{(OLS)}$ at location $\mathbf{x}_i$ is a weighted average of the observed y_j's where the weights are the elements of the i^{th} row of the $\mathbf{H}^{(OLS)}$. For more details on the OLS, MLE and the HAT matrix, see Myers (1990) and Rencher (2000).

2.2.2 Weighted Least Squares

The weighted least squares (WLS) method may be used to obtain the BLUE for $\boldsymbol{\beta}$, when the observed y's are uncorrelated with different variances. That is, $\text{cov}(\mathbf{y}) = \text{cov}(\boldsymbol{\varepsilon}) = \mathbf{V} = diag(\sigma_1^2, ..., \sigma_n^2) \neq \sigma^2\mathbf{I}$, where the $n \times n$ matrix $\mathbf{V}$ is a positive definite diagonal matrix. The idea of WLS is to use the inverse of the variance-covariance matrix, $\mathbf{V}^{-1}$, as weights to give more weight to those observations which have small variability and give less weight to those which have large variability. In RSM, for example, Vining and Bohn (1998) use WLS to estimate a parametric model for a response, due to the nonconstant variance of the response.

The WLS estimator of the $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}}^{(WLS)} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\mathbf{y} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}, \tag{2.8}$$

where $\mathbf{W} = \mathbf{V}^{-1}$ and the estimated response can be obtained as

$$\hat{\mathbf{y}}^{(WLS)} = \mathbf{X}\hat{\boldsymbol{\beta}}^{(WLS)} = \mathbf{X}(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y} = \mathbf{H}^{(WLS)}\mathbf{y}, \tag{2.9}$$

where the $n \times n$ matrix $\mathbf{H}^{(WLS)} = \mathbf{X}(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}$, called the "WLS HAT" matrix. This formula (2.9) essentially shows that $\mathbf{W}$ represents a "global" weight matrix since the weights are unchanged cross all values of $x_1, ..., x_k$, locations where the estimated response is derived. These global weights are different from "local" weights, which are changed at different values of $x_1, ..., x_k$ locations. More details on local weights will be discussed in Section 2.3.

In practice, the variance-covariance matrix $\mathbf{V}$ is usually unknown and a possible method to obtain the estimators for $\boldsymbol{\beta}$ is to estimate the variance-covariance matrix $\mathbf{V}$ from the observed data, $\hat{\mathbf{V}}$, first and then compute the estimated weighted least squares (EWLS) estimates of $\boldsymbol{\beta}$ by replacing $\mathbf{W}$ in equation (2.8) and (2.9) by $\hat{\mathbf{W}} = \hat{\mathbf{V}}^{-1}$. For more details on WLS and EWLS, see Rencher (2000).

2.3 Nonparametric Approach

A parametric function with unknown parameters in the parametric approach has to be assumed correct first before the parameters can be estimated by methods such as the OLS and WLS. If the parametric function is not correct in practice, then the parametric approach becomes inappropriate and the nonparametric approach may be an alternative choice due to flexibility.

Myers (1999) suggests the use of nonparametric RSM (NPRSM) in the following three scenarios:

(i) The main focus of the experiment is on optimization and not on parameter interpretation.

(ii) There is less interest in an interpretive function and more interest in the shape of a response surface.

(iii) The functional form of the relationship between the explanatory variables and the response is highly nonlinear and not well behaved.

Vining and Bohn (1998), Anderson-Cook and Prewitt (2005), Pickle (2006), and Pickle et al. (2006) are some examples of nonparametric applications in RSM. Vining and Bohn (1998) use a nonparametric technique to estimate the process variance. Anderson-Cook and Prewitt (2005) explore several nonparametric techniques such as kernel regression and local linear regression applied in RSM and give recommendations for their use. Both kernel regression and local linear regression will be discussed later. Pickle (2006) and Pickle et al. (2006) compare parametric, nonparametric and semiparametric methods in the traditional RSM setting.

Recall the true underlying but unknown function f in equation (2.1), the mean response function. An estimated function $\hat{f}$ is usually considered effective if it can adequately capture the structure in the data. Typically, $\hat{f}$ is a smooth function. Since there is no assumed relationship between the factors and the response, the nonparametric methods have to rely on the data itself for estimation of the mean response. To estimate $f(\mathbf{x}_0)$ at location $\mathbf{x}_0$, (assuming that f is smooth), is to assume that those responses which are close to $\mathbf{x}_0$ should contain more information about $f(\mathbf{x}_0)$ than those responses which are far away from $\mathbf{x}_0$. To obtain a smooth function $\hat{f}$, some nonparametric methods use the local weighted averaging philosophy such that responses closest to the point of interest, $\mathbf{x}_0$, have more information

about the mean response at $\mathbf{x}_0$ and are therefore assigned higher weight while observations further away from $\mathbf{x}_0$ have less information and are therefore assigned smaller weight. Thus, as stated in Hardle (1990), the basic idea of local averaging is equivalent to the procedure of finding a local weighted least squares estimator.

In the nonparametric regression literature, there are several popular smoothing fitting techniques such as kernel regression (also called Nadaraya-Watson estimator), local polynomial regression, and spline-based regression. For details, see Hardle (1990) and Takezawa (2006). Essentially, the local polynomial regression is an extension of kernel regression but with better properties than kernel regression. Both can be regarded as members of the local polynomial regression family which employs a simple and effective weighting scheme. Details on both kernel regression and local polynomial regression will be presented in the next two subsections.

2.3.1 Kernel Regression

Kernel regression (KER) is designed to fit local constants (or a 0-order polynomial) with a distance-based weighting scheme to obtain estimates. Like a global parametric method with only an intercept in a model, the model matrix (essentially a vector in this special case) may be defined as the $n \times 1$ vector $\mathbf{1}' = (1, 1, ...1)$. By the local weighted least squares method, the KER fit at the point of interest $\mathbf{x}_0$ is given by

$$\hat{y}_0^{(KER)} = (\mathbf{1}'\mathbf{W}_0\mathbf{1})^{-1}\mathbf{1}'\mathbf{W}_0\mathbf{y} = \frac{\sum_{i=1}^{n} h_{0i}^{(KER)} y_i}{\sum_{i=1}^{n} h_{0i}^{(KER)}} = \sum_{i=1}^{n} h_{0i}^{(KER)} y_i = \mathbf{h}_0^{(KER)\prime}\mathbf{y}, \tag{2.10}$$

where the $n \times n$ diagonal matrix $\mathbf{W}_0$, known as the local weight matrix at location $\mathbf{x}_0$, is given by $\mathbf{W}_0 = \left\langle h_{0i}^{(KER)} \right\rangle$, and $\mathbf{h}_0^{(KER)\prime} = (h_{01}^{(KER)} h_{02}^{(KER)} ... \ h_{0n}^{(KER)})$, and $h_{0i}^{(KER)}$ represents a kernel weight assigned to y_i in the estimation of $\hat{y}_0^{(KER)}$. For more details on the local weighted least squares method, see Hardle (1990) and Takezawa (2006).

In Equation 2.10, the kernel weight $h_{0i}^{(KER)}$, originally proposed by Nadaraya (1964) and Watson (1964), is given by:

$$h_{0i}^{(KER)} = \frac{K\left(\frac{x_0 - x_i}{b}\right)}{\sum_{i=1}^{n} K\left(\frac{x_0 - x_i}{b}\right)} \tag{2.11}$$

where K is a univariate kernel function, utilized to give a weight to y_i based on the distance from x_i to the location where the fit is desired, x_0, and b is a specific bandwidth (sometimes called the smoothing parameter) utilized to determine the smoothness of the estimates. The choice of the bandwidth is critical and will be discussed in Section 2.4.1.

The kernel function is a decreasing function in the distance between x_i and x_0. The kernel function takes a larger value when x_i is close to x_0 while it takes a smaller value when x_i is far away from x_0. The kernel function is typically chosen to be symmetric about zero, nonnegative and continuous. There are several choices for the kernel function such as the Gaussian kernel, the uniform kernel, and the Epanechnikov kernel. For more details on types of kernel functions, see Hardle (1990). Since the choice of the kernel function has been shown to be not critical to the performance of the kernel regression estimator (Simonoff (1996)), we will use the simplified Gaussian kernel function given by

$$K\left(\frac{x_0 - x_i}{b}\right) = e^{-\left(\frac{x_0 - x_i}{b}\right)^2}. \tag{2.12}$$

The kernel function presented above in equation (2.11) is for the univariate case. For the multivariate case with k regressors, at the point of interest $\mathbf{x'}_0 = (x_{10}, x_{20}, ..., x_{k0})$, the Gaussian kernel function is given by

$$K(\mathbf{x}_0, \mathbf{x}_i) \propto K\left(\left\|\frac{\mathbf{x}_0 - \mathbf{x}_i}{b}\right\|\right) \quad or \quad \prod_{j=1}^{k} K\left(\tfrac{x_{0j} - x_{ij}}{b}\right), \tag{2.13}$$

where $\mathbf{x'}_i = (x_{1i}, x_{2i}, ..., x_{ki})$ and $\|\|$ stands for the standard L_2 (Euclidean) norm. The two forms of the multivariate kernel function in equation (2.13) are equivalent when the Gaussian kernel function is utilized. For more details on the multivariate kernel function, see Scott (1992).

In terms of a HAT matrix, the kernel fits in matrix notation may be expressed as

$$\hat{\mathbf{y}}^{(KER)} = \mathbf{H}^{(KER)}\mathbf{y}, \tag{2.14}$$

where $\mathbf{H}^{(KER)}$ is the kernel HAT matrix, defined as

$$\mathbf{H}^{(KER)} = \begin{bmatrix} \mathbf{h}_1^{(KER)\prime} \\ \mathbf{h}_2^{(KER)\prime} \\ \vdots \\ \mathbf{h}_n^{(KER)\prime} \end{bmatrix} \tag{2.15}$$

and $\mathbf{h}_i^{(KER)\prime} = (h_{i1}^{(KER)}\ h_{i2}^{(KER)}\ ...\ h_{in}^{(KER)})$ and $h_{ij}^{(KER)} = \frac{K(\mathbf{x}_i,\mathbf{x}_j)}{\sum_{j=1}^{n} K(\mathbf{x}_i,\mathbf{x}_j)}$. The kernel HAT matrix $\mathbf{H}^{(KER)}$ is also called "the kernel smoother matrix", due to its involving the bandwidth b, which determines the smoothness of the fitted function (or model), the estimate of the mean function of y.

2.3.2 Local Polynomial Regression

Kernel regression is the simplest nonparametric method and suitable for many cases (Hardle (1990)), however, it has a problem, called "boundary bias", when a symmetric kernel function, such as the Gaussian, is utilized. This problem can be alleviated by the use of local polynomial regression (LPR), originally introduced by Cleveland (1979). For more details on the boundary bias problem, see Takezawa (2006, pp. 146-148).

LPR can be regarded as a general form of kernel regression. Kernel regression may be considered as a method of fitting constants locally, while LPR may be considered as a method of fitting a polynomial locally. Thus, LPR can be generalized from the kernel regression simply replacing the local constants (or "0-order" polynomials) with the nonzero local polynomials. The local polynomial may be 1st- or higher-order. In our study, we focus on the 1st-order, which is commonly referred to the local linear regression (LLR).

The LLR fit at $\mathbf{x}'_0 = (x_{10}, x_{20}, ..., x_{k0})$ is given by

$$\hat{y}_0^{(LLR)} = \tilde{\mathbf{x}}'_0(\tilde{\mathbf{X}}'\mathbf{W}_0\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_0\mathbf{y}, \tag{2.16}$$

where the $n \times n$ diagonal matrix $\mathbf{W}_0 = \left\langle h_{0j}^{(KER)} \right\rangle$ and $h_{0j}^{(KER)}$ is a kernel weight associated with the distance of $\mathbf{x}'_j$ to $\mathbf{x}'_0$, $j = 1, ..., n$, and $\tilde{\mathbf{x}}'_0 = (1\ x_{10}\ ...\ x_{k0})$. Similarly, the LLR model matrix, $\tilde{\mathbf{X}}$, is defined as

$$\tilde{\mathbf{X}} = \begin{bmatrix} \tilde{\mathbf{x}}'_1 \\ \tilde{\mathbf{x}}'_2 \\ \vdots \\ \tilde{\mathbf{x}}'_n \end{bmatrix}, \tag{2.17}$$

where $\tilde{\mathbf{x}}'_i = (1\ x_{1i}\ ...\ x_{ki})$. In matrix notation, the LLR estimated fits may be expressed as

$$\hat{\mathbf{y}}^{(LLR)} = \mathbf{H}^{(LLR)}\mathbf{y}, \tag{2.18}$$

where $\mathbf{H}^{(LLR)}$, known as the LLR HAT matrix, is given by

$$\mathbf{H}^{(LLR)} = \begin{bmatrix} \mathbf{h}_1^{(LLR)\prime} \\ \mathbf{h}_2^{(LLR)\prime} \\ \vdots \\ \mathbf{h}_n^{(LLR)\prime} \end{bmatrix}, \tag{2.19}$$

where $\mathbf{h}_i^{(LLR)\prime} = \tilde{\mathbf{x}}_i'(\tilde{\mathbf{X}}'\mathbf{W}_i\tilde{\mathbf{X}})^{-\mathbf{1}}\tilde{\mathbf{X}}'\mathbf{W}_i$. It is easy to see from the formula above that estimation of mean response at any location, either $\mathbf{x}'_i$ (an observed data location) or $\mathbf{x}'_0$ (an unobserved data location) is associated with its special weight matrix, due to the local weighting scheme.

Since the LLR fits involve the kernel weight function which depends on the size of the smoothing parameter (the bandwidth), b, as mentioned earlier, the choice of bandwidth is critical and will be discussed in Section 2.4.1. For more details on LLR, see, for example, Fan and Gijbels (1996) and Fan and Gijbels (2000).

2.4 Semiparametric Approach: MRR2

As mentioned earlier, both parametric and nonparametric methods have shortcomings. Parametric methods are inflexible in that a parametric function must be specified before fitting and if this model is incorrect, the resulting fits are subject to the consequence of model misspecification error such as bias in estimating mean response. Nonparametric methods are too flexible in that the resulting estimates of mean response completely depend on the observed data itself and these fits are subject to high variance. In addition, the successful application of the nonparametric approach has usually been limited to fairly large sample sizes and space-filling designs. However, the typical characteristics of traditional RSM experiments, such as small sample size, sparse data, with most of the design points on the edge of design space, all restrict the application of the nonparametric approach.

Semiparametric approaches combine a parametric method with a nonparametric method. One semiparametric method, model robust regression 2 (MRR2) proposed by Mays, Birch and Starnes (2001), was originally developed for situations when there is partial knowledge about the underlying model, a situation very common in practical applications. Mays, Birch and Starnes (2001) compare MRR2 with OLS, LLR, and some other semiparametric

methods and their examples and simulations results show that MRR2 performs the best among these methods in terms of model comparison criteria such as df_{model}, SSE, PRESS, PRESS**, AVEMSE and INTMSE. (PRESS and PRESS** will be discussed in Section 2.4.1 on bandwidth selection. AVEMSE and INTMSE will be discussed in our section on simulation studies.) Unlike the nonparametric method, MRR2 does not require a large sample and tends to work very well when the sample size is small. For examples of MRR2 with small sample sizes, see Mays, Birch and Starnes (2001), Mays and Birch (2002) and Pickle et al. (2006).

MRR2 can improve estimates of mean response by combining both the parametric and nonparametric estimates into one estimate, simultaneously reducing both bias and variance of estimation. MRR2 essentially combines the advantages from the parametric and nonparametric methods and avoids their disadvantages. Pickle (2006) and Pickle et al. (2006) have demonstrated that the MRR2 technique can be successfully applied to model mean response for data from designed experiments for the case of a single response. In this research, we will extend the MRR2 method to the MRO problem. Details concerning the MRR2 technique are presented in the reminder of this section.

MRR2 combines the parametric fit to the raw data with a nonparametric fit to the residuals from the parametric fit via a mixing parameter, λ. The MRR2 approach allows one to specify any other type of parametric and nonparametric methods for some special situations and conditions. In this research, for simplification, as in Mays, Birch and Starnes (2001) and Pickle (2006), our MRR2 combines the parametric fit by the OLS method with the nonparametric fit by the LLR method.

Our final MRR2 fit is given by

$$\hat{\mathbf{y}}^{(MRR2)} = \hat{\mathbf{y}}^{(OLS)} + \lambda \hat{\mathbf{r}}^{(LLR)}, \tag{2.20}$$

where $\lambda \in [0, 1]$, $\hat{\mathbf{r}}^{(LLR)} = \mathbf{H}_r^{(LLR)}\mathbf{r}$, $\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}^{(OLS)}$ and $\mathbf{H}_r^{(LLR)}$ is the LLR HAT matrix for fitting the residuals $\mathbf{r}$ from the parametric fit $\hat{\mathbf{y}}^{(OLS)}$. In terms of HAT matrices, the equation above may be expressed as

$$\hat{\mathbf{y}}^{(MRR2)} = \mathbf{H}^{(OLS)}\mathbf{y} + \lambda \mathbf{H}_r^{(LLR)}\mathbf{r} = \left[\mathbf{H}^{(OLS)} + \lambda \mathbf{H}_r^{(LLR)}(\mathbf{I} - \mathbf{H}^{(OLS)})\right]\mathbf{y} = \mathbf{H}^{(MRR2)}\mathbf{y}. \tag{2.21}$$

Essentially, MRR2 is a semiparametric method in that the MRR2 fits are a combination of parametric and nonparametric fits through the mixing parameter, λ. If the parametric fit is

adequate, then λ should be chosen close to zero by some appropriate λ selector (which will be discussed later). If the parametric fit is inadequate, then λ will be chosen large enough (close to one) so that the nonparametric fit to the OLS residuals can be used to make up for the parametric fit's inadequacy. Thus, as stated in Mays, Birch and Starnes (2001), the amount of misspecification of the parametric model, and the amount of correction needed from the residual fit, is reflected in the size of λ. In practical applications, the user does not know the true underlying function and, consequently, does not know the amount of model misspecification. Thus, the MRR2 method provides an alternative method that is robust to the model misspecification that may be present in the user's proposed model and to the variability that may be present in a nonparametric method.

Obviously, from the equations (2.20 and 2.21), the MRR2 fit involves the choice of bandwidth b, and the mixing parameter, λ. As discussed in Mays, Birch and Starnes (2001), Mays and Birch (2002) and Pickle et al. (2006), λ and b will be chosen separately. The bandwidth b will be chosen first by a data-driven method (which will be discussed later) to enable the smoothing the residuals from the parametric fit. Then based on this selected bandwidth, the MRR2 fit can be calculated and λ chosen by the same data-driven method as the bandwidth, or by an asymptotically optimal data driven method, introduced by Mays, Birch and Starnes (2001). Details on the choice of an optimal λ will be discussed in Section 2.4.2.

2.4.1 Choice of the Smoothing Parameter b

The nonparametric methods require the choice of smoothing parameter b. In addition, the MRR2 also requires the selection of b to be used by the nonparametric method, which is utilized to fit the residuals from the parametric fit. In this research, since LLR is used as the nonparametric method or as part of the semiparametric method to fit the residuals, the following discussion on the choice of the bandwidth will be related to LLR. It is easy to extend the data-driven method for the choice of bandwidth to the nonparametric part of MRR2 by considering residuals as response values.

As mentioned earlier, the smoothness of the estimated function $\hat{f}$ by a LPR method is controlled by the bandwidth b. A smaller bandwidth value gives less weight to points which are further from the point of interest $\mathbf{x}_0$, resulting in the estimation fit, $\hat{f}_0$, based on fewer

data points and therefore resulting in a less-smooth function. On the other hand, a larger bandwidth value gives more weights to those points further away, resulting in a smoother function. As the value of b goes to infinity, all of the data points have equal weights and essentially, the LLR fit becomes a first-order parametric regression fit (that is, a single line regression fit in the single regressor case or a plane in the multiple regressor case), resulting in fits with low variance but possibly high bias, especially if the first-order model is misspecified. On the other hand, when the b goes to zero, the only response receiving a non-zero weight of $\mathbf{x}_i$ in the estimation of f_i is y_i. Therefore, the $\hat{f}$ becomes the "connect-the-dots" function, resulting in a rougher fit with low bias but high variance. Thus, an appropriate choice of b for smoothing achieves a suitable balance of bias and variance of the fitted function.

The choice of bandwidth is crucial in obtaining a "proper" estimate of function f (Mays and Birch, 2002). Any suitable criterion to deal with the trade-off between bias and variance such as the mean squared error(MSE) may be used here to select an appropriate bandwidth. The literature on the bandwidth selection is rich and for a thorough discussion of bandwidth selectors, see Hardle (1990) and Hardle, Muller, Sperlich, and Werwatz (2004). A bandwidth selected by minimizing the traditional MSE has been shown to tend to be too small. The reason is that the criterion relies too much on the individual data points, using them for both fitting and validation (Mays and Birch, 2002). The "leave one out" criterion of Cross-Validation (CV), which is the PRESS statistic (prediction error sum of squares), is introduced to alleviate this problem. The prediction error sum of squares, PRESS, is given by $PRESS = \sum_{i=1}^{n} (y_i - \hat{y}_{i,-i})^2$, where $\hat{y}_{i,-i}$ is the fit at $\mathbf{x}_i$ with the i^{th} observation left out. But, it has been shown that b chosen by the PRESS is still too small on the average, and the resulting fit is biased toward overfitting, resulting in a fit that is too rough (or under smoothed). Einsporn (1987) introduces a penalized PRESS bandwidth selector called "PRESS*", given by

$$PRESS^* = \frac{PRESS}{n - tr(\mathbf{H})}. \tag{2.22}$$

It is essentially the PRESS adjusted by the error degrees of freedom, DF_{error}, (Pickle, 2006; Einsporn, 1987) in the denominator, which is given by

$$DF_{error} = n - tr(\mathbf{H}). \tag{2.23}$$

It arises from its penalty for a fit that is too rough (high bias, relatively too small bandwidth).

However, Mays and Birch (2002) show that PRESS* was found to choose b too large, on the average, and results in a fit that tends to be too smooth. Based on PRESS*, Mays and Birch (1998) and (2002) introduce a new penalized PRESS bandwidth selector called "PRESS**" to counter the shortcoming of PRESS*. The PRESS** is given by

$$PRESS^{**}(b) = \frac{\sum (y - \hat{y}_{i,-i}(b))^2}{n - trace(\mathbf{H}^{(LLR)}(b)) + (n-k-1)\frac{SSE_{\max} - SSE_b}{SSE_{\max}}} \tag{2.24}$$

$$= \frac{PRESS(b)}{n - trace(\mathbf{H}^{(LLR)}(b)) + (n-k-1)\frac{SSE_{\max} - SSE_b}{SSE_{\max}}}, \tag{2.25}$$

where $SSE_{\max}$ is the largest sum of square error over all possible bandwidth values (essentially, $SSE_{\max}$ is the parametric SSE by OLS that results when b goes to infinity) and SSE_b is the sum of square error associated with a specific bandwidth value b. The term added into the denominator, $(n-k-1)\frac{SSE_{\max} - SSE_b}{SSE_{\max}}$, provides protection against a fit which is too smooth (high variance, relatively too large bandwidth).

Mays and Birch (1998) and (2002) also compare PRESS** with other popular bandwidth selectors such as the generalized cross-validation (GCV) and Akaike's Information criterion (AIC). Their examples and simulation results show that PRESS** is the best choice in terms of minimizing integrated mean squared error of fit across a broad variety of data scenarios. Consequently, we will use PRESS** as a bandwidth selector in this research.

2.4.2 Choice of the Mixing Parameter λ in MRR2

After the bandwidth, b^*, is obtained by the data-driven method (PRESS**), a value of the mixing parameter λ, which is utilized to combine the parametric fits on the raw data with the nonparametric fits on the parametric residuals from the raw data, is required. As mentioned earlier and discussed in Mays, Birch and Starnes (2001), two methods may be utilized to obtain λ. One is a data-driven method, which is the same as the one for the bandwidth selection, and the other is an asymptotically optimal data driven method.

One data-driven method is to chose $\hat{\lambda}$ so that PRESS**(λ) is minimized overall $\lambda \in [0, 1]$. Here, PRESS**(λ) is defined as

$$PRESS^{**}(\lambda) = \frac{\sum (y - \hat{y}_{i,-i}(b^*, \lambda))^2}{n - trace(\mathbf{H}^{(MMR2)}(b^*, \lambda)) + (n-k-1)\frac{SSE_{\max} - SSE_{b^*}}{SSE_{\max}}} \tag{2.26}$$

$$= \frac{PRESS(b^*, \lambda)}{n - trace(\mathbf{H}^{(MRR2)}(b^*, \lambda)) + (n-k-1)\frac{SSE_{\max} - SSE_{b^*}}{SSE_{\max}}}. \tag{2.27}$$

As a second data-driven method, pick $\hat{\lambda}$ as the estimated asymptotically optimal value of the mixing parameter for MRR2, given by

$$\hat{\lambda}_{opt} = \frac{\left\langle \hat{\mathbf{r}}, \mathbf{y} - \hat{\mathbf{y}}^{(OLS)} \right\rangle}{\|\hat{\mathbf{r}}\|^2}, \tag{2.28}$$

where $\langle\rangle$ represents the inner product and $\|\|$ represents the standard L_2 (Euclidean) norm.

The examples in Mays, Birch and Starnes (2001) show that the results by the data-driven method and asymptotic method are quite similar even though the sample sizes they considered are not large (e.g., n = 15 for the one regressor case). In this research, we will compare the data-driven method using PRESS** to the estimated asymptotic optimal data driven method to see if the results found by Mays, Birch and Starnes (2001) extend to the MRO problem.

Chapter 3

Overview of Multi-Response Optimization Techniques in RSM

After the model building stage is completed where each regression model built for each response is assumed to be appropriate, the MRO techniques can then be utilized. That is, the i^{th} predicted response value at location $\mathbf{x}$, $\hat{y}_i(\mathbf{x})$, $i = 1, 2, ..., m$, (where m is the number of the responses), is assumed to be an appropriate approximation of the true underlying relationship between the factors and the i^{th} response. Otherwise, the model for the i^{th} response would be misspecified and this misspecification would likely result in misleading optimization solutions. The choice of modeling technique to build an appropriate model is presented in Chapter 2.

As mentioned in Chapter 1, a graphical approach to MRO is to superimpose the response contour plots, originally proposed by Lind et al. (1960), and then determine an "optimal" solution or some feasible regions by visual inspection. This approach is very simple and easy to understand. But it is limited to two or three dimensions of experimental domains. That is, the number of factors are limited to only two or three.

The second approach is a constrained optimization method. The idea of this approach is to formulate the MRO problem into a single response optimization problem with some appropriate constraints on each of the other responses. This approach is desirable when one response is much more important than the other responses and the appropriate constraints are easily determined for each of the other responses. Obviously, the constrained optimization method is not suitable for those situations where the responses are of equfal importance or those

situations where it is not possible to place constraints on less important responses. For more details on the constrained optimization method see, for example, Myers and Montgomery (2002).

The third approach, which is more general, flexible and popular than the two approaches mentioned above is to transform the multiple dimensional problem into a single dimensional problem in terms of some objective function. There are many methods having such objective functions including the desirability function method, the generalized distance measure method, and the weighted squared error loss method. All of these methods can "optimize" all the responses simultaneously with different weights among the responses. Details on these three methods will be discussed in the next three sections.

3.1 Desirability Function Method

The desirability function method, proposed by Derringer and Suich (1980), transforms each response into a dimensionless individual desirability scale and then combines these individual desirabilities into one whole desirability using a geometric mean. That is, a fitted value of the i^{th} response at location $\mathbf{x}$, $\hat{y}_i(\mathbf{x})$, $i = 1, 2, ..., m$, is transformed into a desirability value $d_i(\mathbf{x})$ or d_i, where $0 \leq d_i \leq 1$. The overall desirability (denoted by "$D(\mathbf{x})$" or "D") (which is an objective function) is the geometric mean of all the transformed responses, given by

$$D = (d_1 \times d_2 \times \cdots \times d_m)^{1/m}. \tag{3.1}$$

The value of d_i increases as the "desirability" of the corresponding response increases. The single value of D gives the overall assessment of the entire desirability of the combined m responses levels. Obviously, the range of the value of D is from zero to one. If the value of D is close to zero or equal to zero, then at least one of the individual desirabilities is close to zero or equal to zero. In other words, the corresponding setting for the explanatory variables would be not acceptable. If the value of D is close to one, then all of the individual desirabilities are simultaneously close to one. In other words, the corresponding setting would be a good compromise or trade-off among the m responses. The optimization goal in this method is to find the maximum of the overall desirability D and its associated optimal location(s).

To transform $\hat{y}_i(\mathbf{x})$ to d_i, there are two cases to consider: one-sided and two-sided transformations. One-sided transformations are used when the goal is to either maximize the response or minimize the response. Two-sided transformations are used when the goal is for the response to achieve some specified target value. When the goal is to maximize the i^{th} response, the individual desirability is given by the one-sided transformation

$$d_i = \begin{cases} 0 & \hat{y}_i(\mathbf{x}) < L \\ \left[\frac{\hat{y}_i(\mathbf{x})-L}{T-L}\right]^r & L \leq \hat{y}_i(\mathbf{x}) \leq T \\ 1 & \hat{y}_i(\mathbf{x}) > T \end{cases}, \tag{3.2}$$

where T represents an acceptable maximum value, L represents the acceptable minimum value and r is known as a "weight", specified by the user. Similarly, when the goal is to minimize the i^{th} response, the corresponding individual desirability is written as the one-sided transformation

$$d_i = \begin{cases} 1 & \hat{y}_i(\mathbf{x}) < T \\ \left[\frac{U-\hat{y}_i(\mathbf{x})}{U-T}\right]^r & T \leq \hat{y}_i(\mathbf{x}) \leq U \\ 0 & \hat{y}_i(\mathbf{x}) > U \end{cases}, \tag{3.3}$$

where T is an acceptable minimum value and U is the acceptable maximum value.

When the goal is to obtain a target value, the individual desirability is given by the two-sided transformation

$$d_i = \begin{cases} 0 & \hat{y}_i(\mathbf{x}) < L \\ \left[\frac{\hat{y}_i(\mathbf{x})-L}{T-L}\right]^{r_1} & L \leq \hat{y}_i(\mathbf{x}) \leq T \\ \left[\frac{U-\hat{y}_i(\mathbf{x})}{U-T}\right]^{r_2} & T \leq \hat{y}_i(\mathbf{x}) \leq U \\ 0 & \hat{y}_i(\mathbf{x}) > U \end{cases}, \tag{3.4}$$

where T is the target value, and L and U are the acceptable minimum and maximum values respectively, and r_1 and r_2 are weights, specified by the users.

This desirability function D offers the user great flexibility in the setting of the desirabilities due to allowing users to chose appropriate values of L, U, and T, and of r, r_1, and r_2, for their different specific situations. For more details on the desirability function, see, for example, Derringer and Suich (1980) and Myers and Montgomery (2002).

Derringer (1994) propose an extended and general form of D, using a weighted geometric

mean, given by

$$D = (d_1^{w_1}, d_2^{w_2} ... d_m^{w_m})^{1/\sum w_i}, \tag{3.5}$$

where w_i is the i^{th} weight on the i^{th} response, specified by users. A larger weight is given to a response determined to be more important. There are some other versions of the desirability function D, such as the method, proposed by Kim and Lin (2000), which finds the largest value of the smallest individual desirability, instead of the maximum value of D. For details on other versions of the desirability function including the Kim and Lin method, see Park and Kim (2005). In this research, we will focus on the conventional desirability function in equation (3.1), since it is still the most commonly used method in MRO problems.

3.2 Generalized Distance Method and Weighted Squared Error Loss Method

The generalized distance method, originally proposed by Khuri and Conlon (1981), measures the distance between the overall closeness of the response functions to their respective optima at the same set of conditions (or factors). The objective function is given by

$$(\hat{\mathbf{y}}(\mathbf{x}) - \theta)' \mathbf{\Sigma}_{\hat{\mathbf{y}}(\mathbf{x})}^{-1} (\hat{\mathbf{y}}(\mathbf{x}) - \theta), \tag{3.6}$$

where $\hat{\mathbf{y}}(\mathbf{x})$ is the $m \times 1$ vector of estimated responses at location $\mathbf{x}$, $\mathbf{\Sigma}_{\hat{\mathbf{y}}(\mathbf{x})}$ is the variance-covariance matrix for the estimated responses at this location, and θ is the vector of target values or ideal optimal values. Obviously, the optimization goal is to find the minimum of the distance function and its associated optimal location(s).

The weighted squared error loss method (proposed by, for example, Pignatiello (1993), Ames et al. (1997) and Vining (1998)) can be considered as a general form of the generalized distance method. In Vining's method (1998), the weighted squared error loss function is given by

$$L = (\hat{\mathbf{y}}(\mathbf{x}) - \theta)' \mathbf{C} (\hat{\mathbf{y}}(\mathbf{x}) - \theta),$$

where $\mathbf{C}$ is an appropriate positive definite matrix of weights or costs. The expected loss function is given by $E(L) = \{E[\hat{\mathbf{y}}(\mathbf{x})] - \theta\}' \mathbf{C} \{E[\hat{\mathbf{y}}(\mathbf{x})] - \theta\} + trace(\mathbf{C}\mathbf{\Sigma}_{\hat{\mathbf{y}}(\mathbf{x})})$. Since the $E[\hat{\mathbf{y}}(\mathbf{x})]$ is unknown and $\hat{\mathbf{y}}(\mathbf{x})$ is an unbiased estimator of $E[\hat{\mathbf{y}}(\mathbf{x})]$, a reasonable estimate of $E(L)$ is

$$\hat{E}(L) = (\hat{\mathbf{y}}(\mathbf{x}) - \theta)' \mathbf{C} (\hat{\mathbf{y}}(\mathbf{x}) - \theta) + trace(\mathbf{C}\mathbf{\Sigma}_{\hat{\mathbf{y}}(\mathbf{x})}). \tag{3.7}$$

Here we shall assume that the variance-covariance structure for the responses, $\boldsymbol{\Sigma}$, is known, implying that the variance-covariance matrix at location $\mathbf{x}$, $\boldsymbol{\Sigma}_{\hat{\mathbf{y}}(\mathbf{x})}$, is known. When $\boldsymbol{\Sigma}$ is unknown, Vining (1998) estimates it using the maximum likelihood method.

The optimization goal is to find the minimum of the estimated expected loss function. Vining discusses several possible choices for $\mathbf{C}$. When $\mathbf{C} = \boldsymbol{\Sigma}^{-1}_{\hat{\mathbf{y}}(\mathbf{x})}$, then minimizing the estimated expected loss function is essentially equivalent to minimizing the generalized distance function.

Both the generalized distance method and the squared error loss method take the correlation among the responses into account. Actually, the variance-covariance matrix $\boldsymbol{\Sigma}_{\hat{\mathbf{y}}(\mathbf{x})}$ is a weight matrix (which is similar to the nonconstant variance-covariance matrix $\mathbf{V}$ in WLS in Chapter 2, but weighted on $\mathbf{X}$). When there are no correlation among the responses, the $\boldsymbol{\Sigma}_{\hat{\mathbf{y}}(\mathbf{x})}$ becomes a diagonal matrix. In this case, larger variance of some responses would imply less weight on these responses while smaller variance of some responses would imply more weight on these corresponding responses. See Kros and Mastrangelo (2001) for more discussion on this concept.

3.3 Some Other Studies

Achieving high-quality of products or processes is an important issue in MRO. High-quality is usually related to small variances of the responses. The desirability function method does not take into consideration the variances of the responses and thus it ignores an important aspect of quality. Although the generalized distance method and the weighted squared error loss method both consider the variance-covariance of the responses, their underlying assumption is that each response has their own constant variances. This assumption may not always be true. To achieve the high-quality of products, some researchers apply techniques utilized in a single response into the MRO problem, by considering the simultaneous optimization of both mean and variance of each response, the so-called dual response problem.

For example, Kim and Lin (2006) apply the dual response approach to the MRO problem with the lower-ordered polynomial regression technique for both mean and variance models. Usually, however, lower-ordered polynomial modeling is not appropriate for a variance

process (Pickle, 2006). Ch'ng, Quah and Low (2005) introduce the index C^*_{pm}, a new optimization criterion, to the MRO problem, which is also originally proposed in the dual response surface. The index C^*_{pm} which can be regarded as an extension of the MSE, allows experimenters to find an optimal setting with the mean responses close to their respective target values while the variance of the responses are kept small. But with this method one does not take the relationship among the responses into account and assumes that there are constant variances for each response.

Chapter 4

A Genetic Algorithm

As mentioned in Chapter 1, a genetic algorithm (GA) is a powerful stochastic optimization tool. It is an iterative optimization procedure that repeatedly applies GA operation components (such as selection, crossover and mutation) to a group of solutions until some convergence criterion has been satisfied. In a GA, a search point, a setting in the search space, is coded into a string which is analogous to a *chromosome* in biological systems. The string/chromosome is composed of characters which are analogous to *genes*. In a response surface application, the chromosome corresponds to a particular setting of k factors (or regressors), denoted by $\mathbf{x} = [x_1, x_2, ..., x_k]'$, in the design space and ith gene in the chromosome corresponds to a x_i, the value of the ith regressor. A set of concurrent search points or a set of chromosomes (or individuals) is called a *population*. Each iterative step where a new population is obtained is called a *generation*.

Figure 4.1 illustrates a basic GA procedure. The process begins by randomly generating an initial population of size M and evaluating each chromosome or individual in the population in terms of an objective function. An offspring population is then generated from the initial population, which becomes a parent population, using GA operations such as selection, crossover and mutation. The objective function is evaluated for each individual in the offspring population. M individuals among the offspring and/or current parent population are selected into the next generation by some strategy such as the ranking or the tournament methods (for more details on ranking and tournament, see Section 4.7). Notice that this step is called "replacement" in that the current parent population is "replaced" by a new population, whose individuals come from the offspring and/or current parent population.

After the replacement step, the process is terminated if some stopping rule is satisfied or continued to another generation where the new population will become a parent population to generate an offspring population by GA operations. The GA process is continued until the stopping criterion is satisfied.

GAs are a large family of algorithms that have the same basic structure and differ from one another with respect to several strategies and operations which control the search process. Although the overall performance of the various GA operations remains likely to be problem-dependent (Mayer et al., 2001 and Goldberg, 1989), there are general rules that govern their use. The following sections give more details concerning each GA operation.

4.1 Continuous versus Binary GA

If each chromosome consists of an encoded binary string and a GA works directly with these binary strings/chromosoms, then the GA is a binary GA. However, if each chromosome consists of a real-valued string and a GA works directly with these real-valued strings/chromosomes, then the GA is a continuous GA.

Which type of GA, a binary or continuous GA, is better? Davis (1991) has found that the GA using real number representations has out-performed one with purely binary representations. A similar opinion was given in Haupt and Haupt (2004). In addition, the real-valued coding of chromosomes is simple, convenient, and easy to manipulate. Hamada et al. (2001), Mayer el al. (2001), Heredia-Langner et al. (2003), Borkowski (2003), Heredia-Langner et al. (2004) have successfully utilized continuous GAs. Therefore, in our study, we utilize a continuous GA.

4.2 Parent Population Size

The current population usually refers to a parent population as one that is utilized to generate an offspring population. The size of a parent population, denoted by M, affects both quality of the solution and efficiency of a GA. If the size is too small, not enough information about the entire search space is obtained. Therefore, the GA may fail to find a global or near-global

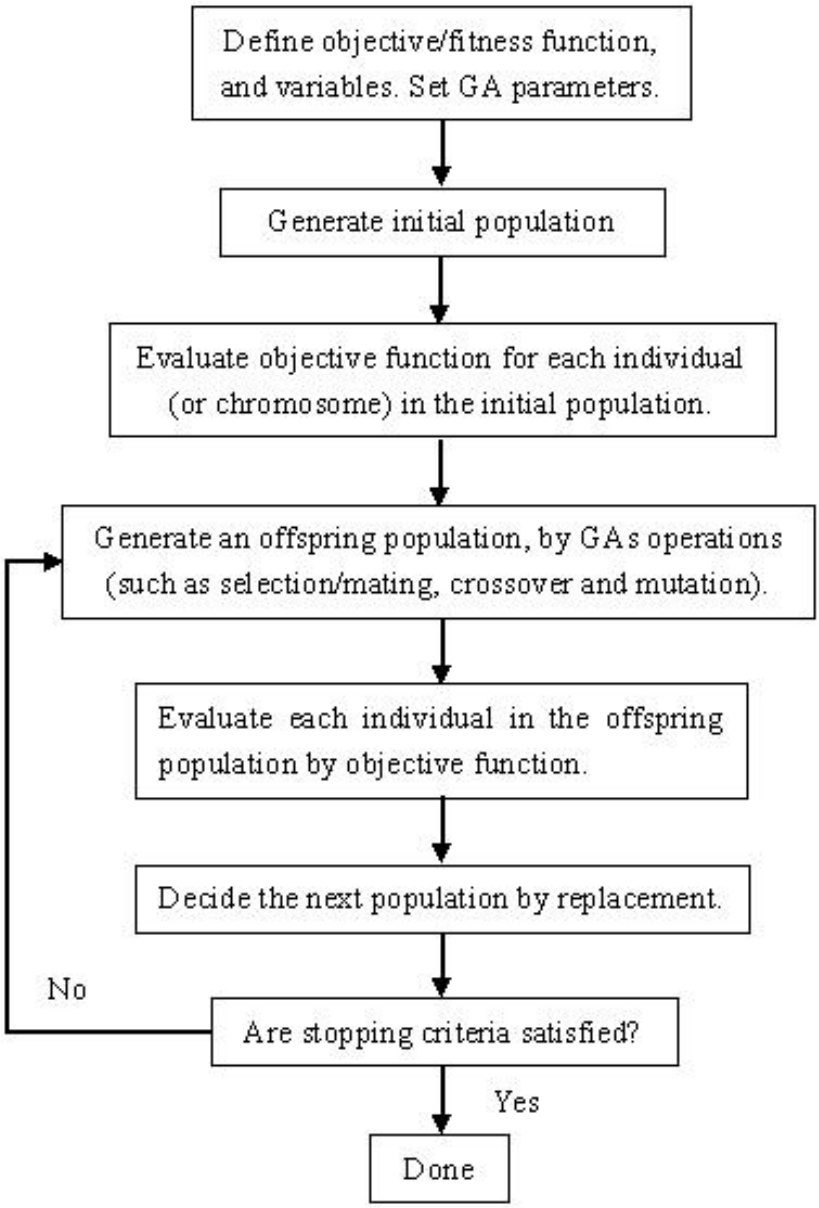

Figure 4.1: A basic GA flowchart

optimum. However, if the size is too large, a large number of evaluations in each generation is required and the GA may become inefficient.

Mayer et al. (2001) suggested that the parent population size depends on the dimensionality of the domain of an objective function. They prefer to use a population size equal to twice the number of factors. For more details, see Peck and Dhawan (1995), Mayer et al. (1996, 1999a, b). In our study, we utilize M equal to 2*k, where k is the number of factors.

4.3 Offspring Population Size

Typically, there are three main choices to determine the size of an offspring population.

First, the offspring population size may be chosen to be much smaller than the parent population size, as in the steady-state GA (SSGA) proposed by Wu and Chow (1995). In the SSGA, only the best two individuals are selected to reproduce two new individuals. Then the two offspring replace the worst two individuals in that current population. Thus, a very small percentage of the population is replaced in each generation. Wu and Chow (1995) show that a SSGA can converge faster and more efficiently than a traditional GA. However, all of the examples they provide only utilize discrete searching spaces, not continuous ones. Our work has checked the SSGA for the continuous case and found that the SSGA offered fast convergence often to a local solution far away from the global optimum. The related results are not presented in the dissertation.

Second, the size of offspring population may be chosen much larger than the size of parent population in each generation. For example, the parent-to-offspring ratio is 1:7 in Heredia-Langner (2003), Ortiz et al. (2004) and Herdia-Langner (2004), and 1:2 in Hamada (2001) and Borkowski (2003). We believe that more offspring generated in each generation can maintain diversity in populations. However, a much larger number of evaluations in each generation is required and the GA may become computationally expensive and inefficient.

Third, the size of offspring population may be chosen the same as the parent population. This suggestion is followed by Goldberg (1989), Holland (1992), and Haupt and Haupt (2004). We utilize this suggestion in our study.

4.4 Selection

Two chromosomes are selected from a mating pool (which could be the current parent population) to be a pair of parents to be utilized to produce two new offspring. Actually, there are several types of selection for mating, including (1) pairing from top to bottom; (2) random pairing; (3) weighted random pairing which includes rank weighting and cost weighting; and (4) tournament selection. For more details, see Haupt and Haupt (2004). In our study, the random pairing is utilized. In the random pairing operation, each individual has an equal chance of being selected for reproducing without replacement, i.e., random partition into sets of two. The reason for choosing this type of selection scheme is that some "bad" individuals may have some "good" components or subregions. Although together they perform badly in terms of an objective function, the "good" components may be beneficial when the "bad" individuals are mated with each other or another individual. The random pairing method also has been used in many papers, such as Hamada et al. (2001), Borkowski (2003), Heredia-Langner et al. (2003), Heredia-Langner et al. (2004) and Ortiz et al. (2004).

4.5 Crossover

Goldberg (1989) and Holland (1992) both believe that the crossover operation is the most important operation in a GA through the concept of schemata and schemata theorem. For more details on schemata, see Goldberg (1989). Crossover allows the exchange of some information from a pair of parents and its transmission to next generation. If the length of a chromosome is k, then the number of crossover points could be m, called "m-point discrete crossover", where $0 < m < k$. Crossover rate is usually high. A crossover rate of 100% (i.e., to make sure to have the crossover operation at each step) is recommended by many researchers including Heredia-Langner et al. (2003), Hamada et al. (2001), Cieniawski et al. (1995), Brokowski (2003), Heredia-Langner et al. (2004) and Wu and Chow (1995).

There are two types of crossover (Haupt and Haupt, 2004): uniform crossover and blending. The uniform crossover is usually used in a binary GA, while the blending crossover, introduced by Radcliff (1991), is primarily used in a continuous GA. The uniform crossover operation, the traditional method, randomly generates positions of crossover points, splits a

pair of chromosomes/individuals at the same positions, and creates two offspring by combining the alternate portions of the parent individuals. If the uniform crossover is utilized in the continuous GA, a problem is that no new information is introduced: each continuous value that was randomly initiated in the initial population is propagated to the next generation, only in different combinations. Although this type of crossover works fine for binary strings, it does not work well in the continuous case as the uniform crossover is merely interchanging two sets of data points and, as such, limits the number of offspring possibilities in the design space.

The blending crossover remedies the problem via a blending parameter β, (which is a value randomly generated within [0, 1]), to combine variable values (or genes) from the two parents into new variable values (or genes) in the offspring. Actually, the blending crossover used in a continuous GA is equivalent to the uniform crossover in a binary GA, by doing encoding and decoding in the binary GA. The blending method has been successfully utilized in Borkowski (2003).

Another important issue on crossover is the number of crossover points. In genetics, the number of crossover points depends on the length of a chromosome. In a continuous GA, the number of crossover points would depend on the number of dimensions of the domain of an objective function. Wu and Chow (1995) conclude that in binary GAs, there is no difference between two-, three- and four-point crossovers, while all of them perform better than one-point crossover. However, Eshelman, Caruna, and Schaffer (1989) point out that eight-point crossover is empirically optimal. In our study, blending crossover with 100% rate is utilized and several different numbers of crossover points are compared.

4.6 Mutation

The mutation operation is used to alter a very small number of the "genetic material" in a random fashion, enhancing the diversity of the population and expanding the volume of the current search space. Generally and naturally, the mutation rate may be very small in a population. Holland (1992) and Goldberg (1989) believe that the role of mutation is not as important as the crossover operation and it is primarily used as a complement to the crossover.

There are two types of mutation operations: random uniform mutation and Gaussian mutation (for more details, see Heredia-Langner et al. (2003)). Haupt and Haupt (2004) suggest use of the random uniform mutation because a good value for σ in Gaussian mutation must be chosen. We will utilize random uniform mutation in our study.

Mutation type is not as an important issue as mutation rate, the probability that a mutation will occur on each population. Back (1996) investigates that an optimal mutation rate depends on the length of an individual chromosome. If the length is l, then the optimal mutation rate is approximately close to $1/l$. Although the results obtained by Back were based on binary GAs, the examples in Ortiz et al. (2004) and Haupt and Haupt (2004) show that in continuous GAs, the optimal mutation rate seems close to $1/k$ (where k is the number of genes). In this paper, several mutation rate levels around or equal to $1/k$ are chosen and compared.

4.7 Replacement

After evaluations of an offspring population, the replacement operation seeks to replace a current parent population by a new population whose individuals are selected from the offspring and/or the current parent population. Heredia-Langner et al. (2003) point out that replacement is an entirely deterministic step and that this operator is to transform the volume-oriented search into a path-oriented exploitation of promising regions.

In an initial GA proposed by Holland (1992) and Goldberg (1989), an offspring population completely replaces a parent population and is involved into next generation. This method has been found to be inefficient. There are several attractive types of replacement: (1) ranking replacement; (2) proportional replacement; (3) tournament replacement with a tournament size of a small number; and (4) extinctive replacement. For more details, see Back (1996), Heredia-Langner et al. (2003), (2004) and Mayer et al. (2001). The most two popular ones, ranking and tournament with size two, are utilized and compared in this study.

In the ranking replacement operation, all individuals in the offspring and current parent population are sorted from best to worst, only the top M individuals replace the parent

population and are involved into next generation. The size of the current parent population, M, is constant in each generation. In the tournament with size two, two individuals are randomly selected, and the best one of the two is moved into next generation. This procedure is repeated until the number of individuals, M, is reached.

4.8 Stopping Rules

The stopping rule operation is important, especially when considering the efficiency of the GA. If the stopping rule stops a GA too early, the GA fails to find an acceptable solution. If the stopping rule stops a GA too late, the GA may be computationally inefficient. There are two rules utilized in this study.

(1) If the global optimum is unknown and a near-global optimum is impossible to know, then the stopping rule could be to simply and arbitrarily choose a pre-selected generation number based on previous experience. This number would be problem-dependent, due to both the complexity of the problem itself and the number of dimensions of the problem. There are many papers utilizing this rule such as Hamada et al. (2001), Cieniawski et al. (1995), Heredia-Langner et al. (2004), Wu and Chow (1995) and Meyer (2003).

(2) If the global optimum is known and it would be easy to select an acceptable value, which is a near-global optimum, then a possible stopping rule stops a GA when the acceptable value is achieved. The reason for an acceptable value is to reduce computational cost. It has been observed that the GA process quickly converges to a region close to the optimum and then slows down in finding the optimum.

If one wants to find the exact global optimum, rather than a near-global optimum, then a hybrid GA would be a better choice than a GA. A hybrid GA procedure is one where a GA helps a local optimization method by finding a good starting point, which is a near-global optimum. Then, the local method is used to find an exact global optimum. For more details on a hybrid GA, see Haupt and Haupt (2004).

4.9 GA Operations Settings or Rules in Our Examples

Our GA operations/parameters settings are chosen, based on the rules above. Table 4.1 is the summary on these settings used in our study.

Table 4.1: Summary on a Continuous Genetic Algorithm Operations Settings or Rules Used in Our Examples

Parameters description	Size or Rule
The number of dimensions	k
The size of parent population	$2k$ or $3k$
The size of offspring population	$2k$ or $3k$
Selection	Random pairing
Blending crossover	m-point, $m \in \{0, 1, 2, 3, 4, 8\}$, and $m < k$
Random uniform mutation	Rate is around or equal to $1/k$
Replacement over parent and offspring	Ranking and tournament
Stopping rules	2 rules used

As mentioned in Chapter 1, a modified GA (MGA) is proposed in Chapter 5. To compare the GA with the MGA, various combinations of several levels of the three main factors (replacement type, crossover points, and mutation rates) are considered through a split-plot design. Details on the comparisons are also discussed in Chapter 5.

Chapter 5

An Improved Genetic Algorithm Using a Directional Search

The genetic algorithm (GA), a very powerful tool used in optimization, has been applied in various fields including statistics. However, the general GA is usually computationally intensive, often having to perform a large number of evaluations of an objective function. This study presents four different versions of computationally efficient genetic algorithms by implementing several different local directional searches into the GA process. These local searches are based on using the method of steepest descent (SD), the Newton-Raphson method (NR), a derivative-free directional search method (denoted by "DFDS"), and a method that combines SD with DFDS. Some benchmark functions (Araujo and Assis, 2000), such as a low-dimensional function versus a high-dimensional function, and a relatively bumpy function versus a very bumpy function, are employed to illustrate the improvement of these proposed methods through a Monte Carlo (MC) simulation study using a split-plot design. A real problem (Myers and Montgomery, 2002) related to the multi-response optimization problem is also used to illustrate the improvement of these proposed methods over the traditional GA and over the method implemented in the Design-Expert statistical software package used by Myers and Montgomery (2002). Our results show that the GA can be improved both in accuracy and in computational efficiency in most cases by implementing a local directional search into the GA process.

5.1 Introduction

A genetic algorithm (GA) is a stochastic optimization tool whose search technique is based on the principals of Darwinian survival of the fittest in biological genetics. The GA, originally developed by Holland (1975), simulates an evolutionary process of a living species, using typical biological genetics operations such as "selection", "mutation" and "crossover". GAs have been applied to a broad variety of fields, including ecology, psychology, biochemistry, biology, computational mathematics, and statistics (*e.g.*, Haupt and Haupt, 2004; Heredia-Langner et al., 2003).

The reason that a GA is so popular and useful is that a GA has some attractive features and advantages (Holland, 1992; Haupt and Haupt, 2004), such as employing multiple concurrent search points (not a single point), not requiring the derivative of an objective function, and being able to find a global or near-global optimum of an objective function with a very complex surface and/or in very high-dimensional domains of the function. A disadvantage of the GA, however, is that it is computationally intensive (Haupt and Haupt, 2004). Typically a GA, in order to find the optimum, must evaluate an objective function a large number of times. For example, if taking 12 hours for only a single evaluation of a complex objective function (which is not unusual in applications), then it could be imagined that the GA would become very time-consuming.

To deal with the computational problem, we proposes and evaluates four versions of a more computationally efficient GA based on modifying a traditional GA in this chapter. The main idea of each version of the modified GAs (MGAs) is to gather numerical information from the GA itself so that a local directional search may be used to make computational improvements to the traditional GA. Four local directional searches used in our MGAs include the method of steepest descent (SD), the Newton-Raphson method (NR), a derivative-free directional search method (DFDS), and a method that combines SD with DFDS.

The remainder of this chapter is organized as follows. Section 5.2 is a brief introduction to a traditional GA and its operations. Section 5.3 gives a brief description of the four local directional search methods. Section 5.4 is focused on the four MGAs. Section 5.5 shows some results for several objective functions giving paired comparisons of the GA and the MGAs across a variety of level combinations of the GA operations and two different stopping rules.

A real case study where the GA is compared to the MGAs is also illustrated. Section 5.6 lists a summary and conclusions, and suggestions for future work.

5.2 The Genetic Algorithm

Genetic algorithms are iterative optimization procedures that repeatedly apply GA operations (such as selection, crossover and mutation) to a group of solutions until some criterion of convergence has been satisfied.

A basic GA procedure has the following steps:

1. Define an objective/fitness function, and its variables. Set GA operations (such as population size, parent/offspring ratio, selection method, number of crossovers and mutation rate).
2. Randomly generate initial population.
3. Evaluate each individual (or chromosome) in the initial population by the objective function.
4. Generate an offspring population, by GA operations (such as selection/mating, crossover, and mutation).
5. Evaluate each individual in the offspring population by the objective function.
6. Decide which individuals to include in the next population. This step is referred to as "replacement" in that individuals from the current parent population are "replaced" by a new population, whose individuals come from the offspring and/or parent population.
7. If a stopping criterion is satisfied, then the procedure is halted. Otherwise, go to Step 4.

GAs are a large family of algorithms that have the same basic structure but differ from one another with respect to several strategies such as stopping rules and operations which control the search process. Based on previous experiences, in this study, we use a continuous GA where chromosomes are coded as continuous measurement variables. We also make the following assumptions. The (parent) population size is $2k$ and the offspring population size is also $2k$. The type of selection we utilize is random pairing. The blending crossover

is utilized and the number of crossover points depends on the number of dimensions of a specific objective function. Random uniform mutation is utilized and the mutation rate is set around or equal to $1/k$. The type of replacement over both parent and offspring population is ranking or tournament. There are two stopping rules used in this study. Stopping rule 1 is that the GA is halted at the pre-selected number of generations. Stopping rule 2 is that the GA is halted when a cutoff value (which is pre-selected and considered as a near-global value) is achieved. For details on the setting of the GA operations, see, for example, Goldberg (1989), Hamada et al.(2001), Mayer, Belward and Burrage (2001), Francisco Ortiz et al.(2004) and Haupt and Haupt (2004).

5.3 Local Directional Search Methods

The GA itself does not utilize a directional search explicitly. In order to improve the computational efficiency of the GA, we modify the GA by incorporating a directional search into the GA process. As mentioned in the introduction, we use four different methods of a local directional search to develop the four MGAs: the method of steepest descent (SD), the Newton-Raphson method (NR), the method of a derivative-free directional search (DFDS), and the method that combines SD and DFDS. SD, NR, and DFDS will be discussed in the next three subsections, respectively.

5.3.1 The Method of Steepest Descent

The method of the steepest descent (SD) was originally introduced by Cauchy in 1874. It starts at an arbitrary point on the surface of an objective function, $f(\mathbf{x})$, where f is the objective function and $\mathbf{x}$ is the arbitrary point, and minimizes along the direction of the gradient. The simple formula for the $(n+1)^{th}$ iteration at location $\mathbf{x}_n$ (where $\mathbf{x}_n = [x_{n1}, ..., x_{nk}]'$) is given by

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \rho_n \nabla f(\mathbf{x}_n), \tag{5.1}$$

where ρ_n is a non-negative scalar and $\nabla f(\mathbf{x}_n) = [\partial f/\partial x_{n1}, ..., \partial f/\partial x_{nk}]'$ is the gradient-based vector. Obviously, each step by SD requires the first derivative of f to calculate a specialized gradient based on that particular location. Note that if one wants to find a maximum of

f and to maximize along the direction of the gradient, then the ρ_n should be non-positive. More details on SD can be seen in Haupt and Haupt (2004).

5.3.2 Newton-Raphson Method

Newton-Raphson method (NR), a second directional search procedure, is based on a first-order Taylor series expansion of the function about the point $\mathbf{x}_n$ given by

$$f(\mathbf{x}) \approx f(\mathbf{x}_n) + (\mathbf{x} - \mathbf{x}_n)'\nabla f(\mathbf{x}_n), \tag{5.2}$$

where $\mathbf{x}$ is some point near $\mathbf{x}_n$. To find an optimal value of f, taking the gradient of both sides of (5.2) and setting it equal to zero yields

$$\nabla f(\mathbf{x}) \approx \nabla f(\mathbf{x}_n) + \mathbf{H}_n(\mathbf{x} - \mathbf{x}_n) \equiv \mathbf{0},$$

where $\mathbf{H}_n$ is the Hessian matrix with elements given by $h_{njl} = \partial^2 f/\partial x_{nj}\partial x_{nl}$, j and $l = 1, ..., k$. Thus the next point, $\mathbf{x}_{n+1}$, can be found by

$$\mathbf{x}_{n+1} \approx \mathbf{x}_n - \mathbf{H}_n^{-1}\nabla f(\mathbf{x}_n). \tag{5.3}$$

More details on NR can be seen in Haupt and Haupt (2004).

Compared to SD in (5.1), NR requires calculating the Hessian matrix (which involves the second derivative of f) and its inverse and thus it usually takes more time than SD for each function evaluation. However, NR does not require the adjustment to the moving step (ρ_n in formula (5.1)) as SD does, since $-\mathbf{H}_n^{-1}$ takes the amount of the moving step into account. In practice, the NR method often requires fewer steps than the SD method to converge to an optimal solution.

5.3.3 A Derivative-free Directional Search Method

The SD and NR methods both require the partial derivatives of an objective function f. It is not expected that SD or NR can always find a proper direction from the current point, since an objective function usually is not simple and unimodal, but very complicated, locally rough and unsmoothed. Thus, we developed a new local directional search method which is derivative-free and denoted by "DFDS."

The goal of DFDS is to find an appropriate direction so as to build the path without requiring the gradient, $\nabla f(\mathbf{x}_n)$. Here we build three potential directions associated with the best offspring in a GA process. When the best offspring is also the best in the current parent population, there is an improvement from its parents to the best offspring in terms of the objective function. It may be possible to make continuous improvements by moving along the directions/paths from its parents to the best offspring. That is, some data points are "collected" along the paths until no further improvement can be found.

When the best offspring among both the offspring and parent populations is found, we can trace back to find its parents. These parents then can be considered as two different starting points. Both of their first steps from the two starting points go to the same point: the best offspring. So two directions are established: one is from one of the parents to the best offspring; the other is from the second of the parents to the offspring. Both directions have obtained improvement, since the best offspring of interest is an improvement over both its parents in terms of values of an objective function.

For example, consider a 2-dimensional ($k = 2$) problem along with the contours of a response (or values of an objective function) as illustrated in Figure 5.1. In general, the best offspring among the offspring and the current parent population is denoted by O (expressed as $\mathbf{x}_O = [x_{O1}, ..., x_{Ok}]'$) and its parents are denoted by P1 ($\mathbf{x}_{P1} = [x_{P11}, ..., x_{P1k}]'$) and P2 ($\mathbf{x}_{P2} = [x_{P21}, ..., x_{P2k}]'$). Obviously, there are two directions: one is from P1 to O, expressed as $\delta_{P1O} = \mathbf{x}_O - \mathbf{x}_{P1} = [\delta_{11}, \delta_{12}, ..., \delta_{1k}]'$ and the other is from P2 to O, expressed as $\delta_{P2O} = \mathbf{x}_O - \mathbf{x}_{P2} = [\delta_{21}, \delta_{22}, ..., \delta_{2k}]'$. We refer to these two directions as the Parent 1 and Parent 2 directions.

The third direction is the "common" direction, expressed as $\delta = [\delta_{31}, \delta_{32}, ..., \delta_{3k}]'$, and based on the two parent directions. If δ_{1i} and δ_{2i}, for $i = 1, ..., k$, are both positive (negative), then δ_{3i} is positive (negative). That is, if both the parent directions are in common, say, both positive (negative) along the X_i axis, then the third direction is positive (negative) along the X_i axis. If δ_{1i} and δ_{2i} are opposite in direction, then δ_{3i} is set to 0. That is, if the parent directions are not in common on the X_i axis, then the third direction has no movement along the X_i axis. For more details on the three directions and determining their moving distances for each moving step, see Appendix A.1.

Figure 5.1 illustrates the three defined directions. The optimal point is denoted by "Θ". It

is easy to see the two parents directions, expressed as $\delta_{P1O} = [\delta_{11}, \delta_{12}]'$ and $\delta_{P2O} = [\delta_{21}, \delta_{22}]'$ respectively. The third direction $\delta = [\delta_{31}, \delta_{32}]'$. Obviously, $\delta_{31} > 0$ since both $\delta_{11} > 0$ and $\delta_{21} > 0$. That is, the common direction in this case is positive along the X_1 axis. And $\delta_{32} = 0$ since $\delta_{12} > 0$ and $\delta_{22} < 0$. That is, the common direction has no relative movement along the the X_2 axis.

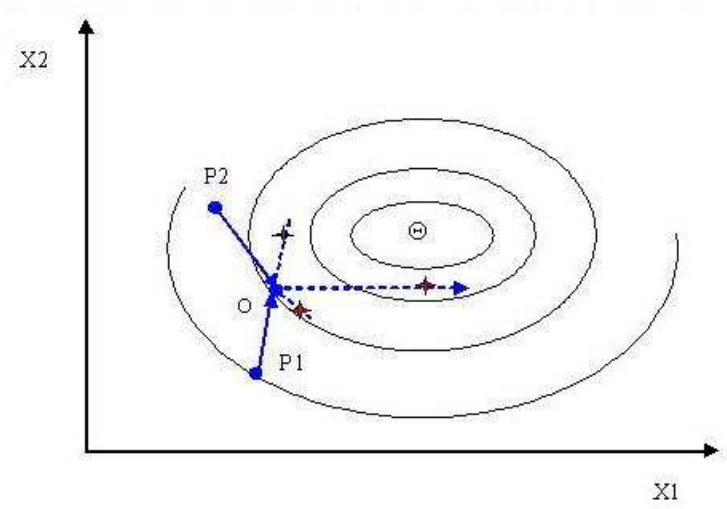

Figure 5.1: A contour plot of a 2-dimensional problem with the three directions indicated: Parent 1 direction is from P1 to O; Parent 2 direction is from P2 to O; the common direction is a horizontal dotted line, starting at O towards the positive values on the X_1 axis. The three "stars" represent the three points stopped on the three paths with no further improvement.

Once the three directions are defined, starting at O, the DFDS method moves along the three directions/paths, with some appropriate moving distance for each moving step until no improvement is found in terms of an objective function. In Figure 5.1, the three "stars" on the paths denote that the three best points found on each path and the processes of moving along the paths will be stopped at their next points due to no further improvement.

5.3.4 A Method Based on Combining SD and DFDS

Unlike the SD, NR, DFDS methods, the fourth method we used in this study is a "combined" method that combines SD, a derivative-base search method with one direction generated, with, DFDS, a derivative-free search method with three directions generated. This method provides a total of four directions to search for the best point.

5.3.5 A Summary of the Methods of a Local Directional Search

In summary, the four local directional search methods used in our four MGAs in this study are SD, NR, DFDS, and the method that combines SD with DFDS. There are many other MGAs that may be considered by using other derivative-based directional searches combined with other derivative-free directions.

We choose these four local directional search methods for our four MGAs because we have the following concerns: (1) SD is quite simple, efficient, but requires the first derivative of f. (2) NR is a very popular optimization tool but requires calculating the Hessian matrix and its inverse matrix. Thus, it may take much more time than SD for each function evaluation. (3) DFDS with the three directions generated is intuitive, reasonable, and derivative-free. (4) For the method that combines SD with DFDS, we want to determine if such a combination performs better than either the SD or the DFDS, separately.

5.4 Modified Genetic Algorithms

We developed four versions of a modified genetic algorithm (MGA). These MGAs are listed as follows: (1) if a directional search by SD is utilized by the GA process, then the MGA is denoted by "MGA$_{SD}$;" (2) if a directional search by NR is utilized, then the MGA is denoted by "MGA$_{NR}$;" (3) if a directional search with the three directions, described in 5.3.3, the DFDS method, is utilized, then the MGA is denoted by "MGA$_3$;" (4) if a directional search with a total of four directions combined by SD and DFDS, then the MGA is denoted by "MGA$_4$."

These MGAs have the same main idea: utilizing numerical information from a GA process itself to find some appropriate local directions by only requiring a few extra function evaluations so that the GA process may be guided to further possible improvement. The numerical information we utilized in our study is focused on the best offspring among both the current parent and offspring populations.

The general procedure for each MGA is the same as that of GA, except that in the i^{th} generation we add Step D between Step 5 and 6 in the original GA procedure in Section 5.2 as follows:

D. Is the best offspring in the offspring population also the best over the current parent population?

D-1. If no, directly go to Step 6.

D-2. If yes, then define and implement a local direction. Collect data points along the paths with some appropriate moving distance until no improvement is observed in the objective function. Find the best point and replace the best offspring by the best point. Then go to Step 6.

The choice of the size of an appropriate moving distance, d, depends on how bumpy the surface of an objective function is. If the surface is very bumpy relative to the region of the domain, then the appropriate d should be relatively small. Otherwise, the appropriate d should be relatively large to make the MGAs more efficient.

Actually, the general MGA process is a special GA process with an extra "branch" (illustrated in Step D) (i.e. requiring only a few extra function evaluations), where the best offspring which is also the best over the current parent population is found. Within the branch, a local direction can be defined by SD, NR, DFDS, or the method that combines SD and DFDS. Along the direction(s)/path(s), data points are collected (i.e. evaluated in terms of an objective function) with an appropriate moving distance for each moving step in a manner similar to the method of steepest ascent/descent until no further improvement is found. The best offspring from the parent population is replaced by the best point found on the paths. Then the branch is ended with possible improvement for the MGA by replacing the best offspring with the new best point found and the MGA process is continued like a GA process until a new extra "branch" is found and generated. That is, a new best offspring, which is also the best in a new current parent population, is found. The whole process is iterated until some appropriate stopping rule is satisfied.

Each MGA is essentially a modification to a GA. Thus, if the GA can jump out of a local optimum, so can the MGAs. In addition, each MGA will more likely produce an improved solution than that obtained by the GA with the same setting of the GA operations. An improved solution results when, under the same situation and the same stopping rule, the best solution found by each MGA is closer to the true global solution (in accuracy) and/or converges faster to a global optimum than by the GA (in computational efficiency).

Computational details for implementation of a directional search into a GA process by the

SD, NR, and DFDS methods are found in Appendixes A.2, A.3 and A.1, respectively. Details for implementation by the method that combines SD and DFDS are straightforward from the details of implementation by SD and DFDS.

5.5 A Simulation Study

In our examples, the main goal is to compare the four MGAs with the GA in computational efficiency and in accuracy for different objective functions under a variety of combinations using different levels of GA operations. At the same time, our sub-goal is to find optimal levels for each operation among a variety of levels of interest for each MGA and the GA.

To make the comparisons more fairly comparable, whenever possible, the same random numbers generated within the GA are also used within each version of the MGAs. Therefore, an experiment is conducted through a split-plot design (Hinkelmann and Kempthorne, 1994) so that paired comparisons can be made under the same settings of the operations and using the same random numbers.

The three whole-plot factors are the three main GA operations: replacement type (denoted by "type" in subsequent references), crossover points (denoted by "crossover"), and mutation rates (denoted by "mutation"). The factor type has two levels: ranking (0) and tournament (1). The factor crossover and the factor mutation have two or three levels, depending on the number of variables used in the objective function. Essentially, these combinations of the three factors correspond to the settings of the three main GA operations. The sub-plot factor is "method" which has five levels: one is the GA (denoted by method = 0) and the other four are MGA_{SD}, MGA_{NR}, MGA_3, and MGA_4 (denoted by method = SD, NR, 3, and 4, respectively).

Under the same setting of the GA operations, the GA and the four MGAs may obtain a different optimum value for different random seeds. Therefore, a Monte Carlo experiment is performed for each specific combination of levels of these three GA operations and repeated 500 times using different random seeds.

5.5.1 Two Stopping Rules

Two different stopping rules are utilized in the experiment. Under rule 1, the algorithm will be halted at a pre-selected number of generations. Thus, this stopping rule can be used to compare the five algorithms for accuracy in finding the optimal value of the objective function. Our MGAs all require extra evaluations of the objective function, f, and MGA_4 usually requires the most evaluations of f among the four MGAs found in our studies. Thus, under the same random seed and the same settings of the GA operations, we let the traditional GA run the number of evaluations equal to the number of the extra evaluations of MGA_4 added to the pre-selected number of evaluations (which is equal to the number of generations times the population size).

Under the rule 2, the algorithm will be halted when the cut-off value, which is close to the optimal value and beyond all-possible local optima, is achieved. In our examples except for the case study, the optimal values are known. Thus, the second rule can be used to compare the five algorithms for efficiency in finding the optimal value of the objective function.

5.5.2 Comparison Criteria

There are three responses of interest used for comparing the four MGAs to the GA. The first one is the best optimal value of an objective function obtained by the GA or MGAs. We denote this optimal value by "best." The second response of interest is the distance from the location of the best value obtained to the location of the true optimal value, denoted by "distance." The third response is the total number of evaluations of the objective function, denoted by "evaluation." Under stopping rule 1, the interesting responses are best and distance. Under stopping rule 2, the most interesting response is evaluation, with best and distance also of interest.

Boxplots (from Minitab) will be our graphical tool to compare the four MGAs to the GA across all the combinations. The numerical criteria utilized for comparison are (1) the mean squared error (MSE) of the responses best and distance, denoted by "MSE(best)" and "MSE(distance)," respectively; (2) the mean and variance of the number of evaluations (the response evaluation), denoted by "Mean(evaluation)" and "Var(evaluation)," respectively; and (3) the number of winners among the 500 replications between any two of the

four algorithms for each setting of the operations (or each combination) in terms of the responses best, distance or evaluation, denoted by "Count(best)," "Count(distance)," and "Count(evaluation)," respectively.

For Criteria (1), the MSE is given by

$$MSE = \frac{\sum_{i=1}^{500} (y_i - T)^2}{500}, \tag{5.4}$$

where y_i is a response (either best or distance) and T is an true optimum for the response y_i. For Criteria (2), the MSE cannot be used for the response evaluation, since no optimum exists for this response. Thus the estimated mean and variance are used as criteria for evaluation in our study. For Criteria (3), among the five algorithms there are a total of ten paired comparisons in terms of the number of winners among the 500 replications for each combination, where a "winner" refers to the most favorable response among each pair of responses being compared. For each paired comparison, there may be some ties when the values obtained by one algorithm are equal to the values by another algorithm. For example, to compare GA versus MGA_{SD} in terms of Count(evaluation), it follows that "Count(evaluation) by GA" + "Count(evaluation) by MGA_{SD}" + "Ties(evaluation)" = 500. In the following examples, the numbers of counting ties will not be presented.

Computational time is used to compare the computational efficiency. Besides MGA_{NR}, the computational time of a single function evaluation with a local directional search implemented for each of the other three MGAs is not very different from that by GA in our C++ code, especially for the cases with a single function evaluation, a time-consuming task. Thus, the number of evaluations under stopping rule 2 will be an appropriate indirect measurement of total computational time for each of the GA, MGA_{SD}, MGA_3, and MGA_4 procedures.

5.5.3 Comparisons for the Benchmark Functions

For the comparisons of GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR}, we have selected five objective functions used in previous GA literature. The five objective functions are (1) the sphere model with smooth surface in 2-dimension (Back, 1996), (2) and (3) the Schwefel's function with relatively a bumpy surface (which has been utilized as a benchmark function by Araujo and Assis (2001)) in 5- and 20-dimensions respectively, (4) and (5) the Rastrigin's

function with a very bumpy surface (another benchmark function by Araujo and Assis (2001)) in 5- and 20-dimension respectively. The results from all these five objective functions show that MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} all perform better in both accuracy and computational efficiency than GA over nearly all combinations and all criteria. The exception is for function (4) (the Rastrigin's function in 5-dimension) in terms of the MSE of the response distance where all MGAs outperform GA in seven combinations out of 18. As an example, we only present some comparison results for function (5) (Rastrigin's function in 20-dimension). This function presents a serious challenge to the GA and the MGAs, due to a very bumpy surface of the function in high dimensions. For details about the sphere model and the Schwefel's function, see Appendix A.4.

Comparisons for the Rastrigin's function with 20 dimensions

A generalized Rastrigin's function is given by

$$f(\mathbf{x}) = \sum_{i=1}^{k} (x_i^2 - 10\cos(2\pi x_i) + 10), \text{ where} - 5.12 \leq x_i \leq 5.12, \tag{5.5}$$

where k is the number of dimensions of the function. Figure 5.2 shows its 1- and 2-dimensional surfaces. The surfaces are very bumpy in a narrow range [-5.12, 5.12]. The goal is to find a minimal value and its corresponding location by GA and MGAs. The minimum of this function is known as $\min(f(\mathbf{x})) = f(0, ..., 0) = 0$. In this study, we compare the five algorithms using the function in 20 dimensions (that is, $k = 20$).

To conduct a split-plot design, the levels of the three whole-plot factors are as follows. The factor type has 2 levels: ranking and tournament; the factor crossover has levels: 2, 4, and 8; and the factor mutation has levels: 0.04, 0.05, and 0.06. There are a total of 18 combinations of type, crossover and mutation. Note the middle level for mutation of 0.05 is $1/k$ where k is the number of genes (or dimensions). For stopping rule 2, the cut-off value, which is a near-global optimum, is set to 0.5. The pre-selected number of generations used by stopping rule 1 is 5,000. The appropriate moving distance for each moving step, d, is set to 0.05.

Under stopping rule 1, Figure 5.3 presents boxplots for the responses best and distance across the 500 repetitions for the MGAs and GA models for each of the 18 combinations of type, crossover and mutation. This figure illustrates that MGA_{SD}, MGA_3, MGA_4, and

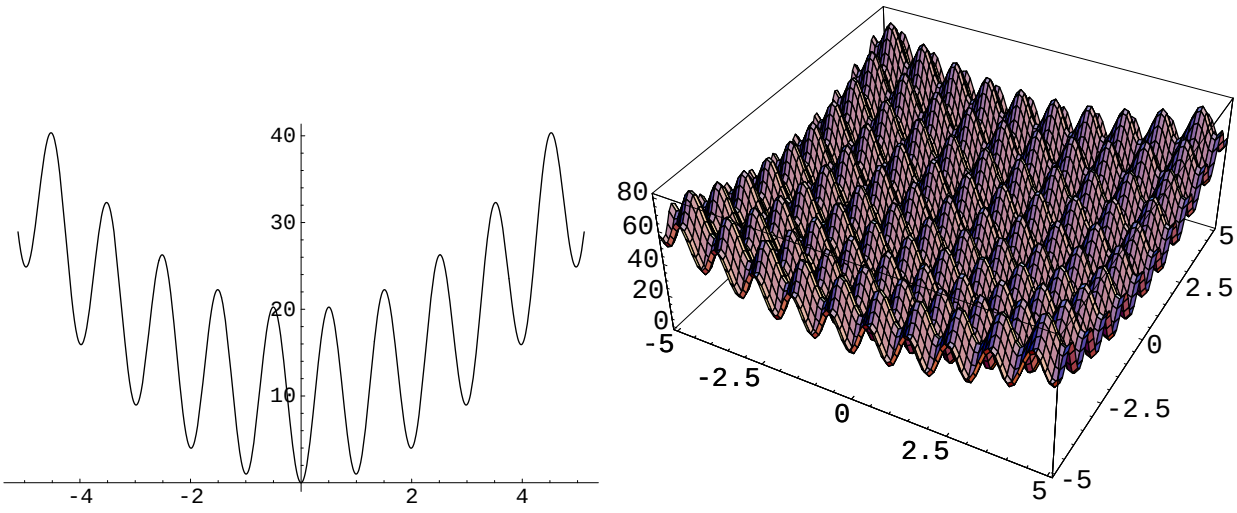

Figure 5.2: Surface of Rastrigin's function. Left: 1-dimension; right: 2-dimension.

MGA_{NR} all perform better than GA over all 18 combinations in terms of the best value and the distance. Not only are all these four new methods more accurate (plots closer to the true minimal value 0), but also more precise (plots exhibit less spread) over all situations. Among the four MGAs, MGA_{NR} performs the best in both accuracy and precision, since the 500 best values obtained by MGA_{NR} all achieve zero (the true minimum) across all of 18 combinations. In addition, MGA_{SD} and MGA_4 both perform much better than MGA_3: when type is 0 (ranking), the best values found by both MGA_{SD} and MGA_4 are all zero across all of the nine combinations shown in the top left boxplot; when type is 1 (tournament), most of the best values by both the MGA_{SD} and MGA_4 are zero except for a few outliers shown in the top right boxplot. The boxplots by the response best and by the response distance express similar patterns. That is, lower best values have smaller distances. The numerical results including the MSE of best and distance (which are not presented here) also match well with Figure 5.3.

Under stopping rule 1, the amounts of time recorded to complete the 500 repetitions for GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} are 22959, 23614, 23120, 23628, and 38616 seconds, respectively. Except for MGA_{NR}, the times of the other four algorithms are relatively similar to each other. The slight differences in the amounts of time between the other four are due to the slightly different computations required for each MGA/GA and to the slightly different numbers of extra function evaluations for MGA_{SD}, MGA_3, and MGA_4. The reason that

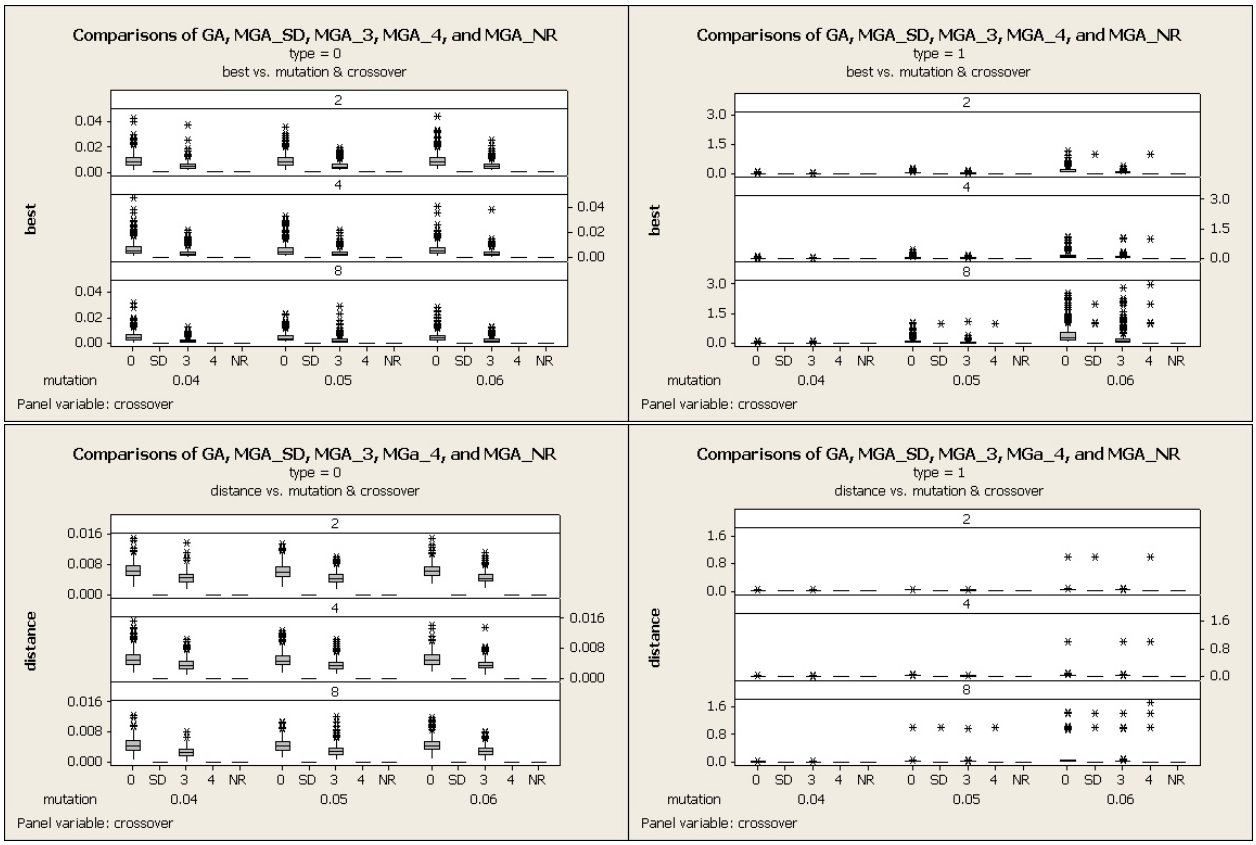

Figure 5.3: Multiple boxplots for comparisons of GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} (denoted by "0, SD, 3, 4, and NR," respectively) in 18 combinations of the factors type, crossover, and mutation for the Rastrigin's function with 20 dimensions by stopping rule 1: the top left is for the response best when type = 0, the top right is for best when type = 1, the bottom left is for the response distance when type = 0 and the bottom right is for distance when type = 1.

MGA_{NR} took much longer than the other four is in calculating the Hessian and its inverse matrix in formula (5.3).

Under stopping rule 2, Table 5.1 presents the mean of the number of function evaluations and its estimated Monte Carlo (MC) error as a summary of the 500 repetitions for comparisons of the five algorithms. It shows that the numbers of evaluations required to obtain a value of the objective function within 0.5 of the true minimum by MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} are all consistently less than required by GA over all combinations. Among the four MGAs, MGA_{NR} performs the best with much smaller mean values for the number of function evaluations than the other three MGAs over all combinations. Among the other three MGAs, MGA_{SD} has the smallest mean values of the number of function evaluations in 12 combinations out of 18, MGA_4 has the smallest mean values in five combinations, while MGA_3 has the smallest value in only one combination (which is the 17^{th}).

Also under stopping rule 2, Table 5.2 presents the paired comparisons of GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} (denoted by "0, SD, 3, 4, and NR," respectively) in terms of the number of winners among the 500 replications for each combination with respect to the response evaluation (denoted by "Count(evaluation)"). Note that Table 5.2 presents only six paired comparisons, not ten (the total number of paired comparisons), because these six paired comparisons are sufficient to rank these five algorithms. These paired comparisons show that all MGAs have more winners than GA over all combinations in terms of the count of the number of evaluations. Among the four MGAs, MGA_{NR} has consistently the most winners over all combinations. MGA_{SD} has the most winners than the other two MGAs across all the combinations, and MGA_4 has more winners than MGA_3 over all combinations.

The amounts of time recorded for GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR}, under stopping rule 2, are 5622, 3717, 4137, 3702, and 15 seconds, respectively. Obviously, MGA_{NR} finds the optimal solution very quickly, with MGA_{SD} and MGA_4 as the next fastest, while the GA is the slowest. These results match well with those in Tables 5.1 and 5.2.

Table 5.1: Comparisons of GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} (denoted by "0, SD, 3, 4, NR," respectively) in terms of mean of the number of evaluations and the estimated Monte Carlo (MC) error of the mean under the 18 combinations of the factors type, crossover, and mutation for the Rastrigin's function in 20-dimensions by stopping rule 2

Combinations			Mean(evaluation)					MC error(mean(evaluation))				
type	cross	muta	0	SD	3	4	NR	0	SD	3	4	NR
0	2	.04	30706	15792	22266	15925	**115**	331	207	272	208	2
		.05	31310	15859	22585	15996	**114**	339	208	274	204	2
		.06	32366	16415	23274	16600	**113**	327	215	281	218	2
	4	.04	27280	15598	19511	15749	**113**	317	222	270	222	2
		.05	26870	15407	19463	15495	**108**	299	213	254	212	2
		.06	28537	16096	21003	16264	**111**	305	208	262	213	2
	8	.04	25270	15933	17354	16122	**108**	331	227	246	229	2
		.05	25604	16018	17554	16224	**106**	293	218	238	214	1
		.06	26705	16564	18917	16744	**107**	315	227	267	225	2
1	2	.04	51259	31001	36751	30899	**118**	413	371	364	342	2
		.05	77109	45175	53467	45011	**116**	722	548	590	555	2
		.06	118768	69806	81530	70206	**122**	1335	1125	1001	1092	2
	4	.04	45254	31973	33970	31918	**113**	392	338	299	342	2
		.05	70515	49371	53342	49323	**117**	738	627	635	613	2
		.06	113514	78707	87176	78988	**116**	1558	1260	1332	1228	2
	8	.04	46250	37753	37965	38155	**109**	444	429	380	416	2
		.05	89548	74159	71903	73197	**111**	1379	1179	1082	1250	2
		.06	177024	149430	152007	148541	**112**	3482	3237	3327	3356	2

Table 5.2: Numerical six paired comparisons of GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} (denoted by "0, SD, 3, 4, and NR," respectively) in terms of the number of winners among the 500 replications for each combination with respect to the response evaluation (denoted by "Count(evaluation)") for the Rastrigin's function in 20-dimensions by stopping rule 2. The maximal MC error is 11.

	Count(evaluation)											
	0	SD	0	3	0	4	SD	4	3	4	SD	NR
1	13	**487**	79	**421**	12	**488**	**475**	25	78	**422**	0	**500**
2	8	**492**	66	**434**	11	**489**	**467**	32	72	**428**	0	**500**
3	14	**486**	69	**431**	13	**487**	**477**	23	75	**425**	0	**500**
4	27	**473**	85	**415**	28	**472**	**482**	18	106	**394**	0	**500**
5	25	**475**	80	**420**	25	**475**	**472**	28	107	**393**	0	**500**
6	23	**477**	84	**416**	26	**474**	**480**	20	95	**405**	0	**500**
7	61	**439**	86	**414**	62	**438**	**482**	18	168	**331**	0	**500**
8	49	**451**	65	**435**	53	**447**	**483**	17	187	**313**	0	**500**
9	50	**450**	89	**411**	55	**445**	**483**	17	155	**344**	0	**500**
10	14	**486**	38	**462**	24	**476**	**348**	152	143	**357**	0	**500**
11	25	**475**	53	**447**	25	**475**	**358**	142	149	**351**	0	**500**
12	44	**456**	66	**434**	47	**453**	**367**	133	175	**325**	0	**500**
13	61	**439**	67	**433**	51	**449**	**355**	145	199	**301**	0	**500**
14	65	**435**	93	**407**	68	**432**	**402**	98	204	**296**	0	**500**
15	95	**405**	128	**372**	93	**407**	**409**	91	203	**297**	0	**500**
16	118	**382**	120	**380**	123	**377**	**438**	62	247	**253**	0	**500**
17	158	**342**	150	**350**	149	**351**	**447**	53	243	**257**	0	**500**
18	183	**317**	190	**310**	179	**321**	**435**	65	238	**262**	0	**500**

Some Other Details on Comparisons Among the Four MGAs using the Benchmark Functions

Among our four MGAs, in the examples of the benchmark functions, MGA_{NR} performs the best in terms of our criteria (as mentioned in Section 5.5.2) except for the amount of time recorded under stopping rule 1 for the Rastrigin's function with very bumpy surface in 5- and 20- dimensions. Among the other three MGAs, under stopping rule 1, MGA_{SD} and MGA_4 are quite competitive with each other and both perform much better than MGA_3 in most cases. In addition, under stopping rule 2, MGA_{SD} performs better than MGA_4 and MGA_3, and MGA_4 performs better than MGA_3 in most cases.

In the examples using Rastrigin's function in 5- or 20-dimensions (as presented in Section 5.5.3), the results show that MGA_{NR} exhibits superior performance over the other three MGAs, especially when using stopping rule 2. When under stopping rule 1, MGA_{NR} took much more time to finish the MGA process than the other three MGAs, although MGA_{NR} still has the best performance by far in terms of other criteria such as MSE(best) and MSE(distance). Except for the time concern, it seems that the local directional search using NR greatly helps the GA process jump out of local peaks or valleys towards the global optimum. But this superior performance by MGA_{NR} appears to hold only for a function with a very bumpy surface. When the Schwefel's function is used with its relatively bumpy surface, the results (not presented in this study) show MGA_{NR} still performs better than MGA_{SD}, but both algorithms are very competitive with each other in terms of all criteria including the amount of time taken. It seems that when the peaks or valleys are further away from each other, the search by NR does not easily jump over them as when the peaks or valleys are quite close to each other.

5.5.4 Comparisons for the Case Study: A Chemical Process

The real example used to illustrate our methods is taken from Myers and Montgomery (2002), where a central composite design (CCD) was conducted on a chemical process. Two independent variables (or factors) are time (x_1) and temperature (x_2). Three responses of interest are yield (y_1), viscosity (y_2) and number-average molecule weight (y_3). The collected data are given in Myers and Montgomery (2002). As in Myers and Montgomery (2002), we

transform the natural independent variables into the coded variables within the range of [0, 1].

In this case study, their multi-response optimization goal is to maximize y_1 (the minimum L = 70 and optimum T = 80), achieve a target value for y_2 (the minimum L = 62, the target T = 65, and the maximum U = 68), and, at the same time, control y_3 within the acceptable range of [3200, 3400]. The desirability function method by Derringer and Suich (1980) is utilized to find simultaneous optimum solutions of the responses y_1, y_2, and y_3.

The desirability function (which is the objective function utilized in GA and the MGAs) is given by

$$D = (d_1 \times d_2 \times \cdots \times d_m)^{1/m} \times 100\%, \tag{5.6}$$

where m is the number of responses and d_i is the i^{th} individual desirability, which is given in Derringer and Suich (1980). The researcher's goal is to find the common location, $\mathbf{x}$, where the maximum value of D is achieved, indicating, in some way, the best location, $\mathbf{x}$, where all the responses achieved their most desirable values simultaneously. In addition, the solution vector, $\mathbf{x}_s$, should be controlled within the experimental region R, which is defined as $(x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5^2$ in this case study.

Under the same conditions such as experimental priority and fitted models given in Myers and Montgomery (2002), the two solutions we found by GA are listed as follows.

1) $x_1 = 0.5758$ $x_2 = 0.1624$ $\hat{y}_1 = 78.6344$ $\hat{y}_2 = 65.0000$ $\hat{y}_3 = 3260.7992$ $D = 0.9292$

2) $x_1 = 0.2661$ $x_2 = 0.7964$ $\hat{y}_1 = 78.2694$ $\hat{y}_2 = 65.0000$ $\hat{y}_3 = 3399.1632$ $D = 0.9094$

These two solutions are different from the two solutions obtained by Design-Expert as shown in Myers and Montgomery (2002) (whose two values of D are 0.822 and 0.792) in terms of fitted optimal values for all of the three responses. The solutions obtained by GA result in larger values of D, indicating that GA performs better at finding the optimal value of D than the algorithm used by Design-Expert in this example.

Figure 5.4 represents the surface (the left graph) of the desirability function D within the experimental region R and its corresponding contour plot (the right graph). The figure shows that there are two distinct surfaces which represent two disjoint operating regions. Obviously, the surface of D matches well to the contour plot. In addition, the two optimal

solutions we found also match well to the figure. Notice that if the case study had more than two or three factors/dimensions, then it would be hard to graphicly show the surface of the desirability function D and its contour plot. Thus, in such a situation, we could not depict graphically the location of the optimal solution. But we still could use either the MGAs or GA to find its optimal or near-optimal solution.

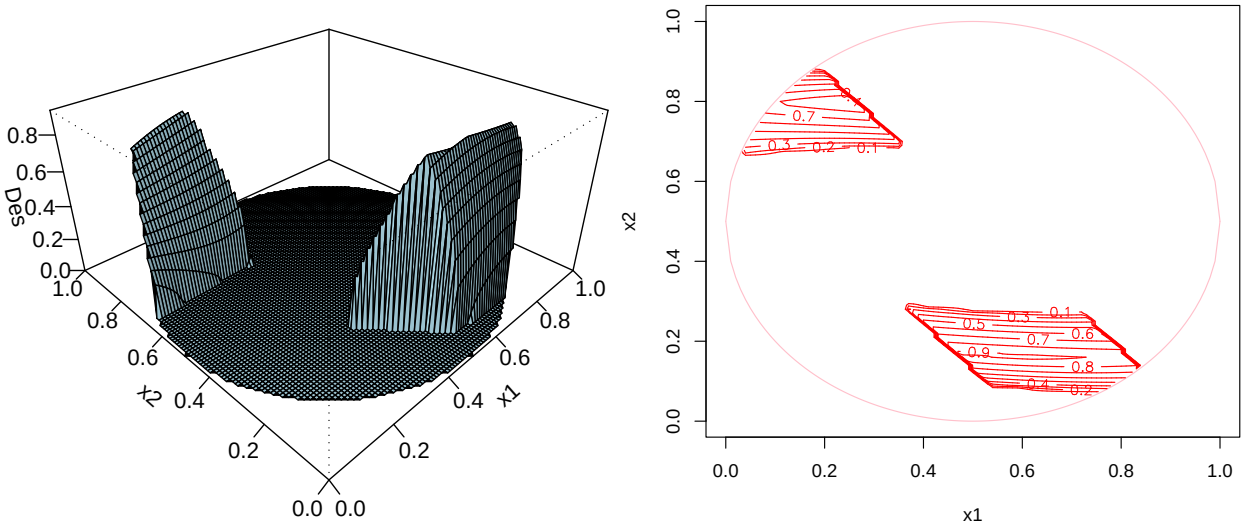

Figure 5.4: The 3-D surface and the contour of the desirability function (denoted by "Des") within the experimental region R in the case study of a chemical process: left: 3-D surface and right: contour

To compare the performance of GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} on this example, a split-plot design is conducted and repeated 500 times, similar to the design used in the five examples mentioned above. The pre-selected number of generations was set at 50. The appropriate moving distance was set at 0.001. Since the true optimal solution is unknown, the response "distance" cannot be measured in this application. Stopping rule 2 is also not suitable in this example, since the pre-specified cutoff which is a near-global optimal value is unknown. We consider only two levels of the factor crossover instead of three considered in the previous example. Thus, there are only 12 combinations of our three factors, type, crossover, and mutation. To calculate the MSE of the response "best" of the desirability function D, MSE(best), for each combination with 500 repetitions, based on formula (2), we need the value of T, the true optimum (the maximum of D), which is, however, unknown in this case. Since the maximum of D is generally close or equal to one, T is set to be one for this example.

Table 5.3 presents the results with respect to the MSE of the response "best" of the desirability function D and the estimated MC error for each combination under stopping rule 1 for this case study. It shows that MGA_4 has the smallest MSEs among the five algorithms over all 12 combinations. MGA_3 has the next smallest MSEs over all combinations. MGA_{SD} has smaller MSEs than GA in six combinations, while MGA_{NR} has only one smaller MSE value than GA.

Table 5.3: Numerical comparisons of GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} (denoted by "0, SD, 3, 4, NR," respectively) in terms of the MSE of the response best and the MC error of the MSE under the 12 combinations of the factors type, crossover, and muation for the case study by stopping rule 1

Combinations			MSE(best) $\times 10^{-3}$					MC error(MSE(best)) $\times 10^{-3}$				
type	cross	muta	0	SD	3	4	NR	0	SD	3	4	NR
0	0	.4	12.75	11.69	10.02	**8.49**	13.09	2.07	2.07	1.75	1.29	2.14
		.5	7.82	7.21	7.07	**6.64**	8.04	0.86	0.79	0.68	0.62	0.95
		.6	7.14	6.73	6.59	**6.30**	7.22	0.67	0.63	0.54	0.48	0.75
	1	.4	8.99	9.05	8.23	**7.94**	10.42	1.27	1.47	1.39	1.37	1.87
		.5	7.19	7.09	6.58	**6.37**	7.75	0.81	0.85	0.60	0.57	1.10
		.6	6.75	6.50	6.50	**6.34**	6.91	0.63	0.59	0.57	0.56	0.67
1	0	.4	10.18	9.54	8.72	**8.15**	11.37	1.45	1.54	1.34	1.29	1.81
		.5	8.25	7.95	7.22	**6.79**	8.94	0.96	1.26	0.71	0.65	1.38
		.6	7.26	6.62	6.65	**6.31**	7.18	0.71	0.57	0.57	0.48	0.72
	1	.4	8.93	9.56	6.91	**6.76**	10.20	1.43	1.70	0.73	0.70	1.75
		.5	6.53	6.53	6.29	**6.16**	6.77	0.62	0.66	0.51	0.48	0.70
		.6	6.14	6.13	5.95	**5.86**	6.38	0.51	0.52	0.41	0.40	0.57

Also under stopping rule 1, Table 5.4 presents the results on the six paired comparisons of GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} in terms of the number of winners among the 500 replications for each combination with respect to the response best (denoted by "Count(best)"). For the same reason as Table 5.2, this table presents only six paired comparisons, not ten. These paired combinations show that MGA_4 has superior performance since it consistently has more winners than the other four over all combinations, although there are some counting ties from each paired comparison (not presented in the table as mentioned in Section 5.5.2). MGA_3 performs the second best since it has more winners than

GA, MGA_{SD}, and MGA_{NR} over all combinations. MGA_{SD} is the third best since it has more winners than GA and MGA_{NR} in most combinations. However, MGA_{NR} is even worse than GA over all combinations.

Table 5.4: Numerical six paired comparisons of GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} (denoted by "0, SD, 3, 4, and NR," respectively) in terms of the number of winners among the 500 replications for each combination with respect to the response best (denoted by "Count(best)") for the case study by stopping rule 1. The maximal MC error is 11.

	Count(best)											
	0	SD	0	3	0	NR	SD	3	3	4	SD	NR
1	136	**221**	88	**347**	**129**	80	199	**284**	62	**201**	**236**	83
2	145	**247**	134	**307**	**146**	89	229	**253**	65	**215**	**257**	102
3	168	**214**	162	**276**	**157**	84	229	**256**	94	**205**	**227**	119
4	189	**211**	173	**321**	**150**	81	213	**285**	73	**185**	**215**	137
5	177	**224**	199	**293**	**156**	88	242	**255**	74	**176**	**217**	149
6	198	**212**	201	**291**	**160**	88	223	**273**	77	**189**	**246**	139
7	118	**245**	117	**340**	**144**	82	218	**258**	52	**211**	**251**	73
8	127	**264**	145	**317**	**150**	104	244	**247**	70	**228**	**261**	99
9	152	**233**	175	**292**	**136**	94	242	**245**	66	**180**	**247**	109
10	**227**	178	183	**308**	**214**	63	195	**301**	85	**164**	**212**	134
11	**244**	194	235	**261**	**242**	78	217	**280**	100	**169**	**226**	175
12	**263**	181	235	**257**	**223**	80	225	**271**	121	**165**	**223**	186

The results in Table 5.4 match well with those in Table 5.3. The amounts of times recorded for GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} are 54, 47, 52, 53, and 49 seconds, respectively.

Unlike the results for the benchmark functions, MGA_{SD} and MGA_{NR} both perform worse than MGA_3. We speculate that one reason for this result is that the surface of the desirability function in the case study has two disjoint "mountains", both of which are locally irregular, unlike the surfaces of the five objective functions which are locally smooth and regular. One reason that MGA_{NR} is worse than GA under stopping rule 1 is that the number of function evaluations required by MGA_{NR} is less than the total of number of evaluations run by GA, which is equal to the number of extra evaluations of MGA_4 added onto the pre-selected number of evaluations, as mentioned in Section 5.5.1. When we let the GA run the number of evaluations equal to the number of the extra evaluations of MGA_{NR} (which is smaller than the number of the extra evaluations of MGA_4) added onto the pre-selected

number of evaluations, the results show that MGA_{NR} has the smaller MSEs than GA in eight combinations out of 12.

5.5.5 Summary on the GA/MGAs Optimal Settings from the Examples

Recall that our goal for this study is to find optimal levels for each operation among a variety of levels of interest to the user of either the MGAs and/or theGA. A Monte Carlo experiment has been performed for each combination of levels of the three GA operations (type, crossover and mutation). In this study, the optimal settings for each algorithm are decided based on the MSEs of the response "best" when using stopping rule 1, and based on the mean of the response "evaluation" when using stopping rule 2.

Table 5.5 presents the summary of the optimal settings for GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} from the examples including the case study. The first row of this table says that in the case study with two factors, under stopping rule 1, the optimal setting for all of the five algorithms is tournament replacement, one crossover point, and 0.6 mutation rate. The presentations of the other rows follow the format of the first row.

From Table 5.5, it seems that under the different stopping rules, each specific example has its own optimal GA setting for each of the five algorithms. These results seem to agree with the "No Free Lunch Theorems for Optimization" conclusions by Wolpert and Macready (1997), which states that the optimal GA setting is problem-dependent and there are no general optimal GA settings.

However, from Table 5.5, there are some rules we may follow before either the MGAs or GA are run. First, the factors crossover and mutation both depend on the length of a chromosome/string (which is the number of genes in a chromosome). Second, ranking, a replacement type, is preferred in most cases, especially when the surface of an objective function is bumpy or very bumpy. Third, the factor crossover is important and the number of crossover points should be increased as the length of a chromosome increases. Fourth, the optimal mutation rate is approximately equal to $1/k$, as suggested in Back (1996).

Table 5.5: Summary on the GA/MGAs optimal settings (combinations) of the GA operations (type, crossover (denoted by "cross"), and mutation (by "muta")) in all of our examples

Examples (Functions)	Number factors	Stopping rules	Optimal settings type	cross	muta	Algorithms
Case study	2	1	tour	1	0.6	All
Sphere model "smooth"	2	1	tour	1	0.4	GA
			—	—	—	MGA_{SD}, MGA_4, MGA_{NR}
			tour	1	0.5	MGA_3
		2	tour	1	0.4	GA,MGA_3
			rank	1	0.4	MGA_{SD}, MGA_4, MGA_{NR}
Schwefel's "bumpy"	5	1	tour	3	0.1	All
		2	rank	3	0.2	GA, MGA_3
			rank	2	0.2	MGA_{SD}, MGA_4, MGA_{NR}
	20	1	rank	8	0.05	GA, MGA_3
			tour	8	0.04	MGA_{SD}, MGA_4, MGA_{NR}
		2	rank	8	0.04	GA, MGA_3
			rank	4	0.05	MGA_{SD}, MGA_4, MGA_{NR}
Rastrigin's "very bumpy"	5	1	rank	3	0.2	GA, MGA_{SD}, MGA_3, MGA_4
			—	—	—	MGA_{NR}
		2	rank	3	0.2	GA, MGA_{SD}, MGA_3, MGA_4
			rank	3	0.1	MGA_{NR}
	20	1	rank	8	0.05	GA
			—	—	—	MGA_{SD}, MGA_4, MGA_{NR}
			rank	8	0.04	MGA_3
		2	rank	8	0.04	GA, MGA_3
			rank	4	0.05	MGA_{SD}, MGA_4
			rank	8	0.05	MGA_{NR}

—: Algorithm achieves the optimal solution, zero, in many combinations.
tour: tournament rank: ranking

5.6 Conclusion and Discussion

This study presents the four versions of modified GAs: MGA_{SD}, MGA_3, MGA_4, and MGA_{NR}, all of which make an improvement over the traditional GA both in accuracy (by stopping rule 1) and in computational efficiency (by stopping rule 2) in most cases. The main idea in our modification is to implement a local directional search into the GA process. The local directional searches utilized in this study to develop our four MGAs include using SD, NR, DFDS, and the method that combines SD with DFDS. MGA_{SD} and MGA_4 both require the first derivative of f, MGA_{NR} requires calculating the Hessian matrix with the second derivative of f and its inverse matrix, while MGA_3 requires no derivative calculations.

Several examples, including a case study of a chemical process, are used to facilitate comparisons of GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR}. Such examples include comparisons between low-dimensional and high-dimensional problems, and smooth, relatively bumpy and very bumpy surfaces. Numerical and graphic comparison results in all of the examples show that the new MGAs procedures perform better than the traditional GA procedure in most cases.

Among the four MGAs, the results show that MGA_{NR} performs the best in the examples using the benchmark functions (Araujo and Assis, 2000) in terms of all comparison criteria, except for the amount of time taken under stopping rule 1 for the benchmark function with a very bumpy surface. Under stopping rule 2, MGA_{NR} demonstrates a considerable improvement over the other MGAs regarding all criteria including the amount of time when using the benchmark function with a very bumpy surface. However, when using the benchmark functions with relatively a less bumpy surface (like the Schwefel's function) or a smooth surface (like the spherical model), MGA_{NR} and MGA_{SD} are quite competitive in terms of all criteria including the amount of time.

Among the other three MGAs, under stopping rule 1, the comparison results in the examples of the benchmark functions show that MGA_{SD} and MGA_4 are competitive with each other and both perform much better than the MGA_3 in most situations. Under stopping rule 2, the comparison results show that MGA_{SD} performs the best with MGA_4 performing better than MGA_3 in most situations. In summary, for our benchmark functions, MGA_{NR} is the top method, followed in order by MGA_{SD}, MGA_4, and MGA_3.

However, the results in the case study are quite different from those in the examples based on the benchmark functions. In the case study, the results show that in terms of all criteria, MGA_4 exhibits superior performance, followed in order by MGA_3, MGA_{SD} and MGA_{NR}. We speculate that one reason that MGA_{SD} and MGA_{NR} both perform worse than MGA_3 is that the surface of the function in the case study has two disjoint "mountains", both of which are locally irregular, unlike the surfaces of the benchmark functions which are locally smooth and regular.

Based on all the results in the examples including the case study, we prefer to use MGA_4 if the first derivative can be taken for an objective function f. If the second derivative can be taken for f and if the surface of f is very bumpy but locally smooth and regular, then we would choose MGA_{NR}. But if derivative cannot be taken for f, then MGA_3 is the only suitable choice.

Several issues remain for further study. For example, the three derivative-free directions defined in MGA_3 may not be optimal. Additionally, the derivative-based directions defined in MGA_{SD} and MGA_{NR} may also not be optimal. Perhaps, there are other directions better than the four we have chosen in this study. Another issue concerns the appropriate moving distance, once the directions are chosen. The size of an appropriate moving distance, chosen carefully by us, may greatly affect the efficiency of the MGAs. The last issue is on the optimal setting of the GA operations. In this study, type of replacement, the number of crossover points, the mutation rate, the three main GA operations, have been studied. However, there may be some other operations affecting the GA performance, such as population size and parent/offspring ratio. We plan to study these issues in future work.

C++ code is available upon request from the authors.

Chapter 6

Using a Modified Genetic Algorithm to Find Feasible Regions of a Desirability Function

The multi-response optimization (MRO) problem in response surface methodology (RSM) is quite common in real applications. Most of the MRO techniques such as the desirability function method by Derringer and Suich (1980) (as mentioned in Section 3.1) are utilized to find one or several optimal solutions. However, in fact, as stated in Myers et al. (2004), practitioners are usually interested in finding not only the optimal solution(s) but the near-optimal solutions as well. That is, the goal in MRO is to find all feasible regions defined as those locations of the factors that result in near-optimal responses. Identifying all feasible regions is often more useful to the practitioner than finding one or several optimal solutions, as certain feasible regions may be more desirable than others based on practical considerations. For example, some of the feasible regions may be larger than other feasible regions, and thus represent a broader range of operating conditions under which the process gives optimal or near-optimal results.

The approach of overlaying the response contour plots, as recommended in Myers and Montgomery (2002), could be a choice to find the feasible operating regions by visual inspection. But this graphical approach is limited to two or three dimensional domains of explanatory variables or factors. In this chapter, a procedure using a modified GA (MGA) is presented to generate approximately all feasible regions for the desirability function without being limited by the number of factors. The GA and MGAs have been discussed in Chapters 4 and 5,

respectively.

The remainder of this chapter is organized as follows. Section 6.1 introduces how to define feasible regions for the desirability function in this study. Section 6.2 justifies the GA/MGA method as useful at approximating all feasible regions, and presents the procedure for using the MGA to approximate all feasible regions of the desirability function. In Section 6.3, a case study is employed to illustrate that this procedure successfully identifies the disjoint feasible regions. Section 6.4 gives a short summary and conclusion.

6.1 Feasible Regions of the Desirability Function

Our goal is to identify all feasible regions, which consist of all near-optimal solutions/locations. All feasible regions may be defined and constructed by all solutions/locations which achieve $D \geq D_{cutoff}$, where D_{cutoff} is some appropriate value. The choice of D_{cutoff} may depend on the global maximum value of D (which we find using a MGA in this study), denoted by "D_{max}", and on the goals of the experiment. Moreover, each solution must be within the experimental region R. For example, suppose that the maximum value of D, found by a MGA, is $D_{max} = 0.85$. The feasible regions may be defined and constructed by all those feasible solutions which achieve $D \geq D_{cutoff} = 0.80$ and which are within the experimental region R.

6.2 Using a MGA to Find Feasible Regions of the Desirability Function

In order to find all feasible regions, we utilize the stochastic nature of a GA/MGA. If one uses the same random seed and the same settings of the GA operations, the MGA has the same stochastic process as the GA because the local directional search implemented in the MGA process, which is the only difference between the GA and MGA, involves no randomness. Each individual/search point obtained in a whole GA/MGA process does not involve randomness. The initial population is completely randomly generated and the successive populations are also generated with partial randomness due to the use of the three

main GA operations which involve randomness as stated in Chapter 4.

When we run a GA/MGA twice with two different random seeds at the same settings of the GA operations, the two initial populations generated by two GAs (or MGAs) processes should be different from each other. It can be seen that these two GAs (or MGAs) should have two different paths towards the same target, a true optimal point. Recall that the feasible region should consist of all of those near-optimal points close to or equal to the true optimal point. If we repeat the GA/MGA process many times, using different random seeds and even different settings of the GA operations (such as the different number of crossover points and different mutation rates), then there are many different paths towards the true optimal point. All of those points, evaluated along each of these paths, that satisfy $D > D_{cutoff}$, can be stored as the collection of feasible points and used to approximate the true feasible region.

In the description above on using the GA/MGA process to find feasible regions, it is assumed that there is one global optimum. This process, however, is also suitable for those cases where there are multiple disjoint local optima with similar values of D.

To find all feasible regions, we prefer to use the MGA, although both the GA and MGA have the same stochastic process given the same settings of the GA operations as mentioned above. The reason is that the Monte Carlo study in Chapter 5 shows that the MGA converges faster than the GA. That is, the best value obtained by the MGA is likely to be closer to the true optimal value than the best value obtained by the GA, and the best location obtained by the MGA is more likely to be closer to the true location of the true optimum than the best location obtained by the GA. Therefore, the MGA should have a greater chance of finding those solutions that are feasible than by the GA.

As mentioned in Chapter 5, we developed four different MGAs using the four different local directional search methods. MGA_4 is the best among the four MGAs, as shown in the examples. Therefore, we use MGA_4 to find all feasible regions in this study.

The procedure of using MGA_4 to determine the approximate feasible regions has four stages, listed as follows.

1. MGA_4 is repeated several times (say, five times) with different random seeds and an optimum and its corresponding location are recorded for each time. The best optimum among all these recorded optima is considered as a global optimum.

2. Based on the results from Stage 1 and based on the priority from the experimenters, the feasible solutions of the desirability function D may be determined. That is, an appropriate cutoff value, D_{cutoff}, may be chosen so that all locations found by MGA_4 which achieve corresponding D values greater than this cutoff could be regarded as feasible solutions.

3. When MGA_4 is repeated many times with different random seeds and with different settings of the GA operations such as different crossover points and mutation rates, feasible solutions are collected when the cutoff value which has been decided in Stage 2 is achieved. As the number of repetitions of MGA_4 is increased, the approximate feasible regions approach the true feasible regions.

4. Plot these feasible points using pairwise 2-dimensional axes. Then, based on these plots, calculate the feasible regions for each factor.

6.3 Case Study: A Chemical Process

To illustrate finding the feasible regions for a specific problem, we consider the following example from Myers and Montgomery (2002), where a central composite design (CCD) was conducted in the chemical process. Two independent variables (or factors) are time (x_1) and temperature (x_2). Three responses of interest are yield (y_1), viscosity (y_2) and number-average molecule weight (y_3). The collected data are given in Myers and Montgomery (2002). As in Myers and Montgomery (2002), we transform the natural independent variables into the coded variables within the range of [0,1].

In this case study, their MRO goal is to maximize y_1 (the minimum L = 70 and optimum T = 80), and achieve a target value for y_2 (the minimum L = 62, the target T = 65, and the maximum U = 68), and, at the same time, control y_3 within the acceptable range of [3200, 3400]. The desirability function method is utilized to find simultaneous optimum solutions of the responses y_1, y_2, and y_3. In addition, the solution vector, $\mathbf{x}_s$, should be within the experimental region R, which is defined as $(x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5^2$ in this case study.

In Stage 1, under the same conditions and fitted models given in Myers and Montgomery (2002), the two solutions we found by MGA_4 are listed as follows.

1) $x_1 = 0.5767$ $x_2 = 0.1624$ $\quad \hat{y}_1 = 78.6344$ $\hat{y}_2 = 65.0000$ $\hat{y}_3 = 3261.3111$ $D = 0.9292$

2) $x_1 = 0.2676$ $x_2 = 0.7964$ $\quad \hat{y}_1 = 78.2821$ $\hat{y}_2 = 65.0000$ $\hat{y}_3 = 3400.0000$ $D = 0.9101$

These two solutions are different from the two solutions obtained by Design-Expert as shown in Myers and Montgomery (2002) (whose two values of D are 0.822 and 0.792) in terms of fitted optimal values for all of the three responses. The solutions obtained by MGA_4 result in larger values of D, indicating that MGA_4 performs better, in this example, at finding the optimal value of D than the Nelder-Mead simplex algorithm used by Design-Expert.

Figure 6.1 represents the surface (the left graph) of the desirability function D within the experimental region R and its corresponding contour plot (the right graph). The figure shows that there are two distinct surfaces which represent two disjoint operating regions. Obviously, the surface of D matches well to the contour plot. In addition, the two optimal solutions we found also match well to the figure.

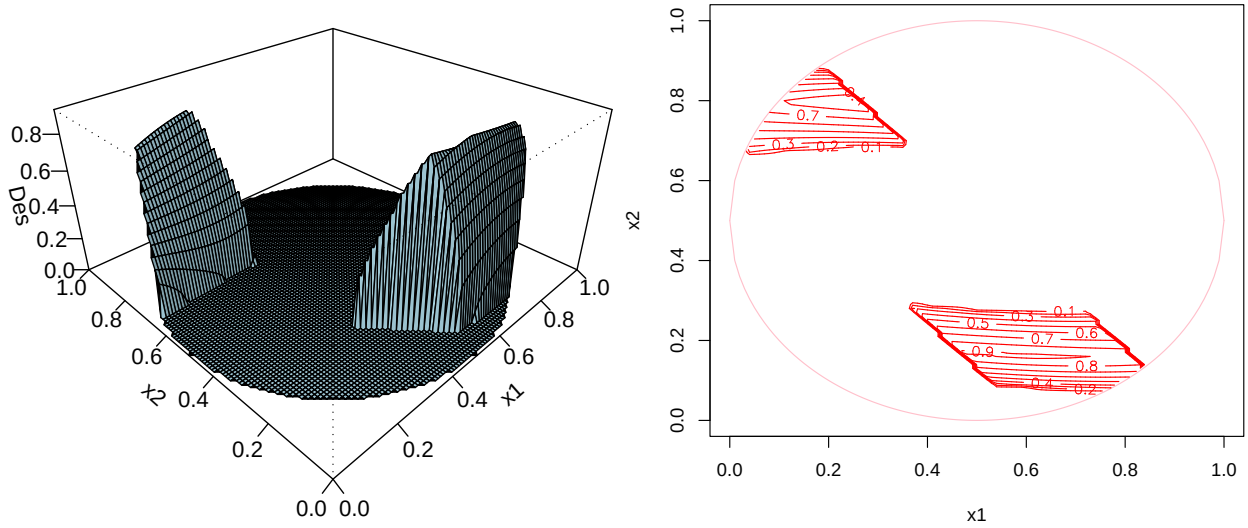

Figure 6.1: The 3-D surface and the contour of the desirability function (denoted by "Des") within the experimental region R in the case study of a chemical process: left: 3-D surface and right: contour

Based on the results from Stage 1, the feasible solutions can be defined by choosing appropriate cutoff values in terms of the desirability function D. In this study, several cutoff values, 0.2, 0.5, 0.8 and 0.9, are used to check if MGA_4 can determine the two feasible regions by collecting feasible points. That is, if the cutoff value of D (0.2, 0.5, 0.8 or 0.9) is achieved, then the corresponding location, which is regarded as a feasible point, is recorded during the MGA_4 process. MGA_4 with 100 iterations is repeated 20 times with 20 different starting random seeds and with 12 different settings of the GA operations to obtain a large enough number of the feasible points. We note that during the MGA_4 process, some of the

same feasible points/locations may be found multiple times. The CPU time is only about 8 seconds using a moderately equipped PC.

Figure 6.2 represents the plots of the feasible points collected by MGA_4 with different cutoff values (0.2, 0.5, 0.8, and 0.9) respectively. It shows that the observed feasible points define two disjoint regions, which correspond to the peaks of the two surfaces shown in Figure 6.1. With the cutoff values increasing from the left to the right in Figure 6.2, the regions become smaller and narrower. Compared to the contour plot of the desirability function in Figure 6.1, it is easy to see that the shapes and sizes of the two disjoint regions are very close to the ones of the contour plot at the four different levels of cutoff values of D. That is, the two disjoint regions are defined very well by the feasible points collected using MGA_4.

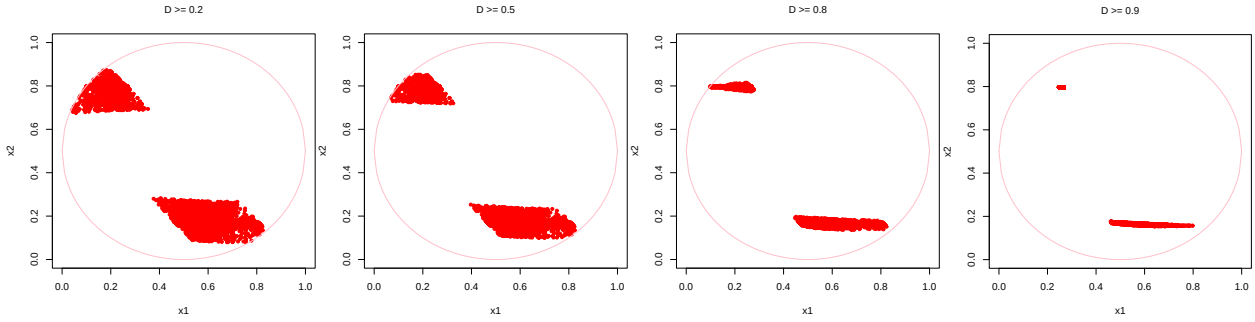

Figure 6.2: Plots of the feasible points collected by MGA_4 with four different cutoff values in the case study of a chemical process: the first graph is by 0.2; the second is by 0.5; the third is by 0.8; and the last is by 0.9.

Based on the knowledge from the plots in Figure 6.2, we can calculate the feasible regions for each factor. For example, suppose that only values of D greater than or equal to 0.9 are acceptable. To calculate the approximate feasible regions, one feasible region would be x_1 in [0.247, 0.268] with x_2 in [0.795, 0.798] and the second feasible region would be x_1 in [0.460, 0.798] with x_2 in [0.153, 0.176]. Obviously, the second feasible region is larger and wider than the first one in terms of the ranges of both factors x_1 and x_2. Therefore, the second feasible region would be considered to be more desirable than the first one, due to a broader set of the operating conditions. In addition, in the feasible region, the factor x_1 (time) is more robust than the factor x_2 (temperature), because x_1 has a wider range than x_2 to achieve the same feasible operating region. Note that the feasible regions calculated only give us the information on the upper and lower bounds for each factor. For more information about the feasible regions such as the shape, we have to rely on the plots of the feasible regions,

approximately displayed by the collected feasible points.

In this case study, the surface of the desirability function D and its contour plot are utilized to check the performance of our method to identify all feasible regions. If the case study had more than two or three factors/dimensions, then it would be difficult to graphicly show the surface of D and its contour plot. Thus, in such situations, we could not tell where the optimal solution is and where the feasible regions are. However, we still could use MGA_4 to find its optimal solution and all its feasible regions and plot them using pairwise 2-dimensional axes.

6.4 Conclusion

Benefitting from the stochastic property of a GA/MGA, this study presents a procedure using a MGA (MGA_4) to determine approximately all possible feasible regions of the desirability function without the limitation of the number of factors (or domains). A case study has been utilized to illustrate the procedure. In this case study with two independent variables, all possible feasible regions can be clearly illustrated and easily computed. In a situation with more than two independent variables, the feasible regions can be displayed by plotting pairwise-coordinates. This procedure can also be easily extended to other MRO techniques which have nonlinear objective functions such as generalized distance measure function and weighted squared error loss function mentioned in Section 1.3. C++ code is available upon request from the authors.

There may be some other alternative methods for finding all feasible regions for the desirability function method. For example, we may use a grid method, which is to put a very fine grid (say, 200×200, if the experimental space is 2-dimensional) on the entire experimental space and to evaluate each point on the fine grid in terms of the desirability function. Similar to our method, those points which achieve $D \geq D_{cutoff}$ can be collected and may approximately construct all feasible regions. Another choice is the Monte Carlo method where the desirability function is evaluated at a large number, say 10,000, randomly select points in the experimental space. Those points satisfying $D \geq D_{cutoff}$ approximately construct all feasible regions. If the experimental space is in very high dimensions (say, 15-dimensions), then the grid method would possibly become very burdensome and the Monte Carlo method

may fail to find the optimal setting. It would be interesting to compare the grid method and the Monte Carlos method with our MGA method to find feasible regions for a desirability function. We leave this project for future research.

Chapter 7

Multivariate Multiple Regression

7.1 Introduction

Recall that the regression techniques, as discussed in Chapter 2, were focused on the univariate case. That is, the regression techniques are utilized to estimate the functional relationship between a single response variable and a set of fixed explanatory variables. In this chapter, we will extend estimation results from the regression techniques in the univariate case to the multivariate case, so as to estimate the functional relationship between several responses and a set of explanatory variables, for the MRO problem.

Suppose on the i^{th} trial, $i = 1, ..., n$, the relationship between m responses, $y_{1i}, y_{2i}, ..., y_{mi}$, and k explanatory variables, $x_{1i}, x_{2i}, ...x_{ki}$, is

$$\begin{aligned} y_{1i} &= f_1(x_{1i}, x_{2i}, ..., x_{ki}) + \varepsilon_{1i} \\ y_{2i} &= f_2(x_{1i}, x_{2i}, ..., x_{ki}) + \varepsilon_{2i} \\ &\vdots \\ y_{mi} &= f_m(x_{1i}, x_{2i}, ..., x_{ki}) + \varepsilon_{mi}, \end{aligned} \tag{7.1}$$

where the function, f_j, represents the true relationship between the j^{th} response, y_{ji}, $j = 1, ..., m$, and the i^{th} set of explanatory variables, $x_{1i}, x_{2i}, ...x_{ki}$, ε_{ji} represents a random error term from the j^{th} response, y_{ji}, and the error term $\varepsilon_i = [\varepsilon_{1i}, \varepsilon_{2i}, ..., \varepsilon_{mi}]'$ has $E(\varepsilon_i) = \mathbf{0}$ and $Var(\varepsilon_i) = \mathbf{\Sigma}$. The error terms associated with different responses may be correlated. However, the n observations within each of the m responses are uncorrelated with constant

variance. $E(y_{ji}|x_{1i}, x_{2i}, ..., x_{ki}) = \mu_{ji} = f_j(x_{1i}, x_{2i}, ..., x_{ki})$. That is, $f_j(x_{1i}, x_{2i}, ..., x_{ki})$ is the j^{th} mean response function.

Similarly to the univariate case in Chapter 2, the true relationship f_j is unknown and must be estimated, based on the collected data. The three regression techniques, utilized to estimate the true relationship in the univariate case, are parametric, nonparametric, and semiparametric methods. In the next sections we will extend the estimation results from these three regression techniques to the multivariate case.

7.2 Parametric Approach

By analogy with the univariate case in Section 2.2, the parametric approach to estimate the relationship is to assume that each response surface is relatively smooth in a relatively small region of the explanatory variables so that the true mean functions f's can be adequately approximated by a low-order polynomial. In this section, based on the work from Rencher (2002), we summarize the estimation results from the second-order model in univariate case (shown in Section 2.2) to the multivariate case.

Given n observations, in matrix notation, the matrix of regressor data is

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{q1} \\ 1 & x_{12} & x_{22} & \cdots & x_{q2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{qn} \end{bmatrix},$$

where $q = 2k + \begin{pmatrix} k \\ 2 \end{pmatrix}$. The $\mathbf{X}$ matrix is essentially the same as that for the single response regression model (given in Equation 2.3). As in Rencher (2002), the other matrix quantities have multivariate counterparts:

$$\mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{m1} \\ y_{1n} & y_{22} & \cdots & y_{m2} \\ \vdots & \vdots & & \vdots \\ y_{1n} & y_{2n} & \cdots & y_{mn} \end{bmatrix} = [\mathbf{y}_1 \ \mathbf{y}_2 \ \cdots \ \mathbf{y}_m],$$

$$\mathbf{B} = \begin{bmatrix} \beta_{01} & \beta_{02} & \cdots & \beta_{0m} \\ \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \vdots & \vdots & & \vdots \\ \beta_{q1} & \beta_{q2} & \cdots & \beta_{qm} \end{bmatrix} = [\boldsymbol{\beta}_1 \ \boldsymbol{\beta}_2 \ \cdots \ \boldsymbol{\beta}_m], \tag{7.2}$$

and

$$\boldsymbol{\Xi} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{21} & \cdots & \varepsilon_{m1} \\ \varepsilon_{12} & \varepsilon_{22} & \cdots & \varepsilon_{m2} \\ \vdots & \vdots & & \vdots \\ \varepsilon_{1n} & \varepsilon_{2n} & \cdots & \varepsilon_{mn} \end{bmatrix} = [\boldsymbol{\varepsilon}_1 \ \boldsymbol{\varepsilon}_2 \ \cdots \ \boldsymbol{\varepsilon}_m].$$

Since each of the m y's will depend on the x's in its own way, each column of $\mathbf{Y}$ will need different $\boldsymbol{\beta}$'s. Thus we have a column of $\boldsymbol{\beta}$'s for each column of $\mathbf{Y}$, and these columns form the $\mathbf{B}$ matrix, $\mathbf{B} = [\boldsymbol{\beta}_1 \ \boldsymbol{\beta}_2 \ \cdots \ \boldsymbol{\beta}_m]$. Our multivariate model is therefore

$$\mathbf{Y} = \mathbf{XB} + \boldsymbol{\Xi}, \tag{7.3}$$

where $\mathbf{Y}$ is $n \times m$, $\mathbf{X}$ is $n \times (q+1)$, $\mathbf{B}$ is $(q+1) \times m$, and $\boldsymbol{\Xi}$ is $n \times m$.

By analogy with the univariate case in Section 2.2.1, we estimate $\mathbf{B}$ with

$$\hat{\mathbf{B}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}. \tag{7.4}$$

$\hat{\mathbf{B}}$ is called the *least squares estimator* because it "minimizes"

$$\mathbf{E} = \hat{\boldsymbol{\Xi}}'\hat{\boldsymbol{\Xi}} = (\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})'(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}}),$$

which is analogous to SSE by using the ordinary least squares method (OLS). Essentially, $\hat{\mathbf{B}}$ minimizes the scalar quantities $tr(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})'(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})$ and $|(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})'(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})|$ (Rencher, 2002).

By analogy with the univariate case in Section 2.2.1, some assumptions which lead to the estimator $\mathbf{B}$ possessing desirable properties are as follows : 1) $E(\mathbf{Y}) = \mathbf{XB}$ or $E(\boldsymbol{\Xi}) = \mathbf{O}$; 2) $\text{cov}(\mathbf{y}_i) = \boldsymbol{\Sigma}$ for all $i = 1, 2, ..., n$, where $\mathbf{y}'_i$ is the i^{th} row of $\mathbf{Y}$; and 3) $\text{cov}(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{O}$ for all $i \neq j$. Under these assumptions, $\mathbf{B}$ has minimum variance among all possible linear unbiased estimators (Rencher, 2002).

As previously mentioned, a column of $\mathbf{B}$ corresponds to each column of $\mathbf{Y}$; that is, each y_j, $j = 1, 2, ..., m$, is predicted differently depending on location $\mathbf{x}_0 = [x_{10}\ x_{20}\ ...\ x_{q0}]'$. In the estimate $\hat{\mathbf{B}}$, we have a similar pattern:

$$\begin{aligned}\hat{\mathbf{B}} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{y}_1\ \mathbf{y}_2\ ...\ \mathbf{y}_m) \\ &= [(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}_1\ (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}_2\ ...\ (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}_m] \\ &= [\hat{\boldsymbol{\beta}}_1\ \hat{\boldsymbol{\beta}}_2\ ...\ \hat{\boldsymbol{\beta}}_m]. \end{aligned} \tag{7.5}$$

The estimated responses can be obtained as:

$$\begin{aligned}\hat{\mathbf{Y}}^{(OLS)} &= \mathbf{X}\hat{\mathbf{B}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} = \mathbf{H}^{(OLS)}\mathbf{Y} \\ &= \mathbf{H}^{(OLS)}(\mathbf{y}_1\ \ \mathbf{y}_2\ \ ...\ \ \mathbf{y}_m) \\ &= [\mathbf{H}^{(OLS)}\mathbf{y}_1\ \ \mathbf{H}^{(OLS)}\mathbf{y}_2\ \ ...\ \ \mathbf{H}^{(OLS)}\mathbf{y}_m] \\ &= [\hat{\mathbf{y}}_1^{(OLS)}\ \ \hat{\mathbf{y}}_2^{(OLS)}\ \ ...\ \ \hat{\mathbf{y}}_m^{(OLS)}]. \end{aligned} \tag{7.6}$$

Equations 7.5 and 7.6 show that the multivariate estimation results are equal to the univariate results in Section 2.2.1 in terms of estimated coefficients and fitted values for each response.

The only difference in OLS results between the univariate case and multivariate case is in estimation of the variance-covariance structure $\text{cov}(\mathbf{y}_i) = \boldsymbol{\Sigma}$. The variance-covariance structure cannot be estimated in the univariate case, while it can be estimated in the multivariate case, due to taking the correlation among the residuals into account. The unbiased estimator of $\text{cov}(\mathbf{y}_i) = \boldsymbol{\Sigma}$ is given by

$$\mathbf{S}_e = \frac{\mathbf{E}}{n-q-1} = \frac{(\mathbf{Y}-\mathbf{X}\hat{\mathbf{B}})'(\mathbf{Y}-\mathbf{X}\hat{\mathbf{B}})}{n-q-1} = \frac{\mathbf{Y}'\mathbf{Y}-\hat{\mathbf{B}}'\mathbf{X}'\mathbf{Y}}{n-q-1}. \tag{7.7}$$

The estimated variance of each response is on the diagonal of $\mathbf{S}_e$, and equivalent to the estimated variance in the univariate case. More details on the multivariate multiple regression model can be seen in Rencher (2002).

7.3 Nonparametric Approach

Recall that the nonparametric methods do not specify a specific functional form, and rely completely on the data itself for estimation of a mean response. In this section, we extend

the estimation results of LLR (which is utilized in this research) in the univariate case to the multivariate case.

According to Equation 7.1, the true relationships for the m responses at location $\mathbf{x}_0$ can be expressed as

$$(y_{10}\ y_{20}\ \ldots\ y_{m0}) = (f_1(\mathbf{x}_0)\ f_2(\mathbf{x}_0)\ \ldots\ f_m(\mathbf{x}_0)) + (\varepsilon_{10}\ \varepsilon_{20}\ \ldots\ \varepsilon_{m0}). \tag{7.8}$$

Similar to the multivariate parametric approach in Section 7.2, we obtain a separate estimate of the mean function for each of the m responses. Similarly, we obtain the bandwidth separately for each of the m responses using the PRESS** selection method. The size of bandwidth chosen for each response is usually different for each response and thus the local weight matrices at location $\mathbf{x}_0$ for each response are also different for each response. Therefore, our multivariate model by LLR is

$$(y_{10}\ y_{20}\ \ldots\ y_{m0}) = (\tilde{\mathbf{x}}_0'\boldsymbol{\beta}_{10}\ \tilde{\mathbf{x}}_0'\boldsymbol{\beta}_{20}\ \ldots\ \tilde{\mathbf{x}}_0'\boldsymbol{\beta}_{m0}) + (\varepsilon_{10}\ \varepsilon_{20}\ \ldots\ \varepsilon_{m0}), \tag{7.9}$$

where $\tilde{\mathbf{x}}_0' = (1\ x_{10}\ ...\ x_{k0})$ is the same as mentioned in Equation (2.16), $\boldsymbol{\beta}_{j0}$ is the "local" coefficient vector for the j^{th} response at location $\mathbf{x}_0$. The difference between $\boldsymbol{\beta}_{j0}$ and the coefficient $\boldsymbol{\beta}_j$ in Equations 7.2 and 7.3 is that $\boldsymbol{\beta}_j$ is constant across all locations $\mathbf{x}_0$. However, due to the local weighting scheme as shown in Equations 2.11 and 2.13, $\boldsymbol{\beta}_{j0}$ is "localized" and changes with each location $\mathbf{x}_0$.

By analogy with the univariate case in Section 2.3.2, using the local weighted least squares method, the estimate of the coefficient $\mathbf{B}_0 = (\boldsymbol{\beta}_{10}\ \ldots\ \boldsymbol{\beta}_{m0})$ at $\mathbf{x}_0$ is obtained as

$$\begin{aligned}\hat{\mathbf{B}}_0 &= (\hat{\boldsymbol{\beta}}_{10}\ \ \ldots\ \ \hat{\boldsymbol{\beta}}_{m0}) \\ &= ((\tilde{\mathbf{X}}'\mathbf{W}_{10}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{10}\mathbf{y}_1\ \ \ldots\ \ (\tilde{\mathbf{X}}'\mathbf{W}_{m0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{m0}\mathbf{y}_m),\end{aligned} \tag{7.10}$$

where the LLR model matrix $\tilde{\mathbf{X}}$ includes the column of ones and the k first-order terms, as mentioned in Equation (2.16). $\mathbf{W}_{j0}$ is a diagonal weight matrix for the j^{th} response at location $\mathbf{x}_0$ and is the same as the weight matrix in the univariate case shown in Equations 2.10 and 2.16. The LLR fits for each response at $\mathbf{x}_0$, therefore, are

$$\begin{aligned}\hat{\mathbf{y}}_0^{(LLR)} &= (\hat{y}_{10}^{(LLR)}\ \ ...\ \ \hat{y}_{m0}^{(LLR)}) \\ &= [\tilde{\mathbf{x}}_0'(\tilde{\mathbf{X}}'\mathbf{W}_{10}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{10}\mathbf{y}_1\ \ ...\ \ \tilde{\mathbf{x}}_0'(\tilde{\mathbf{X}}'\mathbf{W}_{m0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{m0}\mathbf{y}_m] \\ &= [\mathbf{h}_{10}^{(LLR)\prime}\mathbf{y}_1\ \ ...\ \ \mathbf{h}_{m0}^{(LLR)\prime}\mathbf{y}_m].\end{aligned}$$

In matrix notation, the multivariate LLR estimated fits may be expressed as

$$\hat{\mathbf{Y}}^{LLR} = [\mathbf{H}_1^{(LLR)\prime}\mathbf{y}_1 \ \ ... \ \ \mathbf{H}_m^{(LLR)\prime}\mathbf{y}_m],$$

where $\mathbf{H}_j^{(LLR)}$, $j = 1, ..., m$, is also the same as the j^{th} LLR HAT matrix in the univariate case shown in Equation 2.19.

7.4 Semiparametric Approach

Recall that MRR2, a semiparametric method, utilized in this research, combines a parametric fit to the raw data with a nonparametric fit to the residuals from the parametric fit via a mixing parameter. In this section, we extend the estimation results of MRR2 in the univariate case to the multivariate case.

Similar to the multivariate LLR approach, we obtain separate MRR2 fits for each of the m responses. The sizes of the bandwidth b and the mixing parameter λ chosen for each response are likely different from each other. According to Equation 7.8 on the true relationships for the m responses at location $\mathbf{x}_0$, our multivariate model by MRR2 is

$$(y_{10} \ \ldots \ y_{m0}) = (\breve{\mathbf{x}}'_0\boldsymbol{\beta}_1 + \lambda_1\tilde{\mathbf{x}}'_0\boldsymbol{\beta}_{1r0} \ \ldots \ \breve{\mathbf{x}}'_0\boldsymbol{\beta}_m + \lambda_m\tilde{\mathbf{x}}'_0\boldsymbol{\beta}_{mr0}) + (\varepsilon_{10} \ \ldots \ \varepsilon_{m0}), \tag{7.11}$$

where $\breve{\mathbf{x}}'_0 = (1 \ x_{10} \ x_{20} \ \ldots x_{q0})$ includes all terms in a full second-order model, $\boldsymbol{\beta}_j$ the coefficient for the j^{th} response corresponding to the parametric part of MRR2 is the same as the j^{th} one in Equations 7.2 and 7.3 for the model by OLS, $\tilde{\mathbf{x}}'_0 = (1 \ x_{10} \ x_{12} \ldots \ x_{1k})$ is the same as in Equation 7.9, and $\boldsymbol{\beta}_{jr0}$, similar to the "local" coefficient in Equation 7.9 for the model by LLR, is the coefficient for the j^{th} residuals (considered as a response) from the parametric part $\breve{\mathbf{x}}'_0\boldsymbol{\beta}_j$ and $\tilde{\mathbf{x}}'_0\boldsymbol{\beta}_{jr0}$ corresponds to the nonparametric component of MRR2.

By analogy with the univariate case in Section 2.4, the estimate of the coefficient $\mathbf{B} = (\boldsymbol{\beta}_1 \ \ldots \ \boldsymbol{\beta}_m)$ is obtained by OLS, as shown in Equation 7.5. The estimate of the coefficient $\mathbf{B}_{0r} = (\boldsymbol{\beta}_{1r0} \ \ldots \ \boldsymbol{\beta}_{mr0})$ is obtained by the local weighted least squares method (considered the m residual vectors from the parametric fits as the m response vectors) as

$$\begin{aligned}\hat{\mathbf{B}}_{0r} &= (\hat{\boldsymbol{\beta}}_{1r0} \ldots \hat{\boldsymbol{\beta}}_{mr0}) \\ &= ((\tilde{\mathbf{X}}'\mathbf{W}_{1r0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{1r0}\mathbf{r}_1 \ldots (\tilde{\mathbf{X}}'\mathbf{W}_{mr0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{mr0}\mathbf{r}_m),\end{aligned} \tag{7.12}$$

where $\mathbf{r}_j$ is the j^{th} residuals from the j^{th} parametric fits, and similar to the weight matrix in Equation 7.10, $\mathbf{W}_{jr0}$ is a diagonal weight matrix for the j^{th} residuals $\mathbf{r}_j$ (considered as a response) at location $\mathbf{x}_0$. Therefore, the MRR2 fits for each response at $\mathbf{x}_0$, are

$$\begin{aligned}\hat{\mathbf{y}}_0^{(MRR2)} &= (\hat{y}_{10}^{(MRR2)} \ \ldots \ \hat{y}_{m0}^{(MRR2)}) \\ &= [\breve{\mathbf{x}}'_0(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}_1 + \hat{\lambda}_1\tilde{\mathbf{x}}'_0(\tilde{\mathbf{X}}'\mathbf{W}_{1r0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{1r0}\mathbf{r}_1 \ \ \ldots \\ &\quad \breve{\mathbf{x}}'_0(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}_m + \hat{\lambda}_m\tilde{\mathbf{x}}'_0(\tilde{\mathbf{X}}'\mathbf{W}_{mr0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{mr0}\mathbf{r}_m] \\ &= [\mathbf{h}_{10}^{(MRR2)\prime}\mathbf{y}_1 \ \ \ldots \ \ \mathbf{h}_{m0}^{(MRR2)\prime}\mathbf{y}_m], \end{aligned} \tag{7.13}$$

where $\mathbf{h}_{j0}^{(MRR2)\prime} = \breve{\mathbf{x}}'_0(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' + \hat{\lambda}_j\tilde{\mathbf{x}}'_0(\tilde{\mathbf{X}}'\mathbf{W}_{jr0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{jr0}(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')$, and $\hat{\lambda}_j$ is chosen by PRESS$^{**}(\lambda)$ or estimated by the asymptotic optimal data driven method, both of which are mentioned in Section 2.4.2.

In matrix notation, the multivariate MRR2 estimated fits may be expressed as

$$\begin{aligned}\hat{\mathbf{Y}}^{(MRR2)} &= [\mathbf{H}^{(OLS)}\mathbf{y}_1 + \lambda_1\mathbf{H}_{1r}^{(LLR)}\mathbf{r}_1, ..., \mathbf{H}^{(OLS)}\mathbf{y}_m + \lambda_m\mathbf{H}_{mr}^{(LLR)}\mathbf{r}_m] \\ &= [\mathbf{H}_1^{(MRR2)}\mathbf{y}_1 \ \ \ldots \ \ \mathbf{H}_m^{(MRR2)}\mathbf{y}_m],\end{aligned}$$

where $\mathbf{H}_{jr}^{(LLR)}$ is the LLR HAT matrix for fitting the j^{th} residuals $\mathbf{r}_j$, and $\mathbf{H}_j^{(MRR2)}$ is the j^{th} MRR2 HAT matrix, which is also the same as the MRR2 HAT matrix in the univariate case mentioned in Equation 2.21.

Unlike the parametric approach, neither the nonparametric nor semiparametric approaches have a nice closed-form expression for the variance-covariance estimator $\hat{\mathbf{\Sigma}}$, due to different local weight structures and different values of bandwidth and/or different values of mixing parameter among the m responses. But an estimator of the variance-covariance matrix can be obtained in a general way. As in Shah, Montgomery and Carlyle (2004), an estimator of $\mathbf{\Sigma}$ is given by

$$\hat{\mathbf{\Sigma}} = \frac{\mathbf{E}}{n} = \frac{(\mathbf{Y} - \hat{\mathbf{Y}})'(\mathbf{Y} - \hat{\mathbf{Y}})}{n}, \tag{7.14}$$

where the denominator n is the sample size, which is the same as the denominator of the variance estimator in the univariate case when using the maximum likelihood method. In Equation 7.14, if $\hat{\mathbf{Y}}$ is $\hat{\mathbf{Y}}^{(LLR)}$, then $\hat{\mathbf{\Sigma}}$ is one possible LLR variance-covariance estimator. If $\hat{\mathbf{Y}}$ is $\hat{\mathbf{Y}}^{(MRR2)}$, then $\hat{\mathbf{\Sigma}}$ is one possible MRR2 variance-covariance estimator. If $\hat{\mathbf{Y}}$ is $\hat{\mathbf{Y}}^{(OLS)}$, then $\hat{\mathbf{\Sigma}}$ is the OLS variance-covariance estimator and is equivalent to the variance-covariance estimator using the maximum likelihood method.

Chapter 8

A Semiparametric Approach to Multi-Response Optimization

As mentioned in Chapters 1 and 2, MRR2 can be a good choice for estimating the mean response regression function. In this chapter, a case study and simulation studies will be utilized to compare MRR2 with OLS and LLR during the modeling stage and then the different estimated mean functions obtained by these three methods will be utilized by the MRO techniques during the optimization stage.

As mentioned in Chapter 2, after bandwidth selection, MRR2 has two different mixing parameter selectors to choose the size of λ: one uses PRESS** to obtain an appropriate λ, denoted by λ_1; and the other uses an estimate of the asymptotically optimal λ (Mays, Birch, and Starnes, 2001), denoted by λ_2. In this chapter, we will compare both methods to check if the estimate of the asymptotic method is still suitable to RSM data, as determined by Mays, Birch, and Starnes (2001), especially when the sample size is small. The MRR2 estimator with λ_1 is denoted by "$MRR2_{\lambda 1}$" while the MRR2 estimator with λ_2 is denoted by "$MRR2_{\lambda 2}$".

After the model building stage is completed, where each regression model built for each response is assumed to be appropriate, the optimization stage starts. In this study, the desirability function method is utilized to obtain an optimal solution with the best compromise of the multiple responses. The MGA_4 method (as mentioned in Chapter 5) is used to find an optimal solution for the desirability function under the three different modeling methods respectively. Details on the GA and MGA methods are discussed in Chapters 4

and 5, respectively.

Before presenting the case study, more details on how to select an appropriate bandwidth parameter, b, and on model comparison criteria are discussed as follows.

8.0.1 Choice of the Smoothing Parameter b

As mentioned in Chapter 2, the choice of bandwidth is crucial in obtaining a "proper" estimate of the true underlying function f. Also, as mentioned in Chapter 2, in this study, we will use $PRESS^{**}(b)$ as a bandwidth selector for the LLR fit, given in Equations 2.24-2.25. Based on the work in Mays and Birch (2002) and in Pickle et al. (2006), and based on some additional examples, the following is our procedure for finding an appropriate bandwidth parameter, b, using PRESS**. Recall that the regressors are rescaled to be between zero and one.

1. Set the range for the bandwidth selection as [0.1, 1]. Note the bandwidth could be greater than 1. Usually, $b \geq 1$ means that a large bandwidth is best and, hence, the LLR fit is simply a first-order parametric regression fit.
2. Calculate PRESS** given the values of bandwidth within the range [0.1, 1].
   - 2.1 If the b which achieves the global minimum value of PRESS** is less than 1, then this value of b is used to obtain the LLR fit.
   - 2.2 If the b which achieves the global minimum value of PRESS** is equal to 1, then go to Step 2.2.1.
     - 2.2.1 If there are one or more than one local minimums of PRESS**, then the b with the smallest local minimum value of PRESS** would be chosen as an optimal bandwidth.
     - 2.2.2 If there are no local minimums of PRESS**, then the LLR fit is a first-order parametric regression fit.

The procedure above is suitable for the case study. For simulation studies, Step 2.2.1 is changed for simplification and computational efficiency as follows: the b with the first local minimum of PRESS** would be chosen to obtain the LLR fit which is the same as Mays and Birch (2002). To obtain the b with the first local minimum, in this study, we use the absolute relative error (ARE), which is given by

$$ARE = \frac{|PRESS^{**}(b_i) - PRESS^{**}(b_{i-1})|}{PRESS^{**}(b_{i-1})}, \tag{8.1}$$

where $PRESS^{**}(b_i)$ is the $PRESS^{**}$ value when the value of bandwidth is b_i, $b_i = b_{i-1}+0.01$ in this study. If ARE was smaller than some pre-selected tolerance limit(say, 0.001), then the iteration process stops and the bandwidth would be chosen as b_i. R code is written to accomplish this procedure.

8.0.2 Model Comparison Criteria

To compare the three modeling methods, some numerical criteria are utilized: (1) DF_{error}, the degree of freedom of error; (2) s^2, estimate of error variance; (3) R^2, coefficient of determination; (4) R^2_{adj}, adjusted R^2; (5) PRESS; (6) PRESS*; (7) PRESS**. Criteria 1-4 focus on describing how well a model is fit by the observed data, while Criteria 5-7 focus on describing some functions of prediction variance associated with the fitted model.

Criteria 2-5 are standard criteria for comparing models (Myers, 1992). Criteria 1, 6, and 7 have been mentioned in Chapter 2. The DF_{error}, given in Equation 2.23 can be used to compute the degrees of freedom for the model as $DF_{model} = DF_{total} - DF_{error}$. DF_{model} represents the complexity of the model. DF_{total} is the total number of degrees of freedom, equal to n, the sample size. PRESS*, a penalized PRESS bandwidth selector, given in Equation 2.22, essentially is a PRESS adjusted by DF_{error} in its denominator. PRESS**, given in Equations 2.24-2.27, the second penalized PRESS bandwidth selector, is utilized for the bandwidth selection and for the mixing parameter (λ_1) selection.

8.1 The Minced Fish Quality Example

A real example, originally introduced by Tseo et al. (1983) and utilized by Shah, Montgomery and Carlyle (2004), is from food science and is used here to illustrate the various procedures we have proposed. The goal of the study is to determine the optimum combination of the levels of three processing factors (washing temperatures (x_1), washing time (x_2), and washing ratio of water volume to sample weight (x_3)) on minced fish quality, expressed by four responses (springiness (y_1), thiobarbituric acid number (TBA) (y_2), cooking loss (y_3), and whiteness index (y_4)). The observed data were collected through a CCD and presented in Table 8.1.

Table 8.1: A CCD with three factors and four responses on minced fish quality

Coded Variables			Response values			
x_1	x_2	x_3	y_1	y_2	y_3	y_4
0.203	0.203	0.203	1.83	29.31	29.50	50.36
0.797	0.203	0.203	1.73	39.32	19.40	48.16
0.203	0.797	0.203	1.85	25.16	25.70	50.72
0.797	0.797	0.203	1.67	40.18	27.10	49.69
0.203	0.203	0.797	1.86	29.82	21.40	50.09
0.797	0.203	0.797	1.77	32.20	24.00	50.61
0.203	0.797	0.797	1.88	22.01	19.60	50.36
0.797	0.797	0.797	1.66	40.02	25.10	50.42
0	0.5	0.5	1.81	33.00	24.20	29.31
1	0.5	0.5	1.37	51.59	30.60	50.67
0.5	0	0.5	1.85	20.35	20.90	48.75
0.5	1	0.5	1.92	20.53	18.90	52.70
0.5	0.5	0	1.88	23.85	23.00	50.19
0.5	0.5	1	1.90	20.16	21.20	50.86
0.5	0.5	0.5	1.89	21.72	18.50	50.84
0.5	0.5	0.5	1.88	21.21	18.60	50.93
0.5	0.5	0.5	1.87	21.55	16.80	50.98

Shah, Montgomery and Carlyle (2004) use the second-order polynomial parametric method to model each response to obtain the optimal fitted value $\hat{y}_i(\mathbf{x})$ at location $\mathbf{x}$, for $i = 1, 2, 3, 4$. The final fitted second-order models for the four responses by OLS are given in Shah, Montgomery and Carlyle (2004). For the responses y_1 and y_4, the final fitted models include three terms: intercept, x_1 and x_1^2. The final model for the response y_2 includes five terms: intercept, x_1, x_2, x_1^2 and x_{12}. The model for y_3 has a total of eight terms, including intercept, x_1, x_2, x_3, x_1^2, x_{12}, x_{13}, and x_3^2.

For each response, the design spaces we use for the LLR and the nonparametric part of the MRR2 are the same as the ones used for the OLS by Shah, Montgomery and Carlyle (2004). In addition, according to Equation 2.21, the model spaces for the parametric part of the MRR2 we use are the same as the ones for the OLS. For example, for the response y_2, the design space consists of the two factors x_1 and x_2, and thus these two factors, excluding the factor x_3, are utilized in the LLR and the nonparametric part in the MRR2. Meanwhile, the model space consists of intercept, x_1, x_2, x_1^2 and x_{12} altogether, and thus these five terms will construct the modeling space for the parametric part in the MRR2. For the same example as above, the modeling space for the LLR consists of intercept, x_1, and x_2, because the LLR is the local linear regression.

8.1.1 Results on Model Comparisons

As mentioned before, the three modeling techniques, OLS, LLR, and MRR2, are to be compared in the case study. Table 8.2 shows the numerical results for model comparisons of OLS, LLR and MRR2 with two different methods for λ selection for all the responses respectively with respect to the seven criteria mentioned at the beginning of this chapter. If the two λ's chosen by these two methods are equal, then both corresponding results should be the same and therefore only one result is presented. Otherwise, both corresponding results are presented. Table 8.2 shows that MRR2 has smaller s^2 than OLS and LLR across all of the four responses. MRR2 has larger R^2 and R^2_{adj} than OLS and LLR in three of the four responses. MRR2 has smaller PRESS, PRESS* and PRESS** than OLS and LLR in most cases. Furthermore, in the case with MRR2 which does not perform the best among the three modeling techniques, MRR2 still is very competitive to the best modeling technique in terms of all the seven criteria.

Table 8.2 also shows that OLS has consistently larger DF_{error} than LLR and MRR2 in that LLR and MRR2 both need more degrees of freedom to estimate the local relationship between the response(s) and the factors. In addition, the DF_{error} by MRR2 is typically competitive to the DF_{error} by LLR.

Table 8.2 also shows that for responses y_1, y_3, and y_4, MRR2 using $PRESS^{**}(\lambda)$ for λ selection is the same as MRR2 using the estimated asymptotic method. For the response y_2, MRR2 has smaller λ_1 by the data-driven method PRESS** than λ_2 by the asymptotic data-driven method. MRR2 with the smaller λ_1 has much smaller PRESS, PRESS* ,and PRESS** than with the larger λ, but is a little worse in terms of s^2, R^2, and R^2_{adj}. Therefore, in the following sections, for the response y_2, the model by MRR2 with λ_1 will be utilized.

Table 8.2: Results on model comparisons of OLS, LLR, and MRR2 with two different methods for λ selection for all the responses in the minced fish quality example

		b	λ	DF_{error}	s^2	R^2	R^2_{adj}	PRESS	PRESS*	PRESS**
	OLS	—	—	14.000	1.653E-03	0.9211	0.9099	0.0582	0.0042	0.0042
y_1	LLR	0.146	—	12.138	1.039E-03	**0.9570**	0.9433	0.0682	0.0056	0.0026
	MRR2	0.170	1	12.268	**1.033E-03**	0.9568	**0.9436**	**0.0473**	**0.0039**	**0.0025**
	OLS	—	—	12.000	7.5417	0.9341	0.9122	**234.1166**	**19.5097**	19.5097
	LLR	0.436	—	11.212	21.8508	0.8217	0.7456	785.7855	70.0873	36.4222
y_2	$MRR2_{\lambda 1}$	0.277	0.6	10.164	5.5280	0.9591	0.9356	236.9712	23.3154	**15.3159**
	$MRR2_{\lambda 2}$	0.277	1	8.940	**4.8253**	**0.9686**	**0.9438**	319.3332	35.7214	19.6311
	OLS	—	—	9.000	4.5641	0.8408	0.7170	182.4468	**20.2719**	20.2719
y_3	LLR	0.537	—	8.373	9.7990	0.6821	0.3925	287.0564	34.2849	17.0554
	MRR2	0.542	1	6.596	**2.9031**	**0.9258**	**0.8200**	**177.6750**	26.9357	**13.1264**
	OLS	—	—	14.000	14.2182	0.5407	0.4751	684.7407	48.9101	48.9101
y_4	LLR	0.120	—	12.031	1.0197	0.9717	0.9624	**454.5871**	**37.7832**	**17.1484**
	MRR2	0.119	1	12.029	**1.0158**	**0.9718**	**0.9625**	486.8458	40.4725	18.6472

Figures 8.1 - 8.4 show the comparisons of the plots of the factor x_1 versus the response y_1, y_2 (at $x_2 = 0$, 0.5, and 1), y_3 (at all combinations of $x_2 = 0$, 0.5, and 1, and $x_3 = 0$, 0.5, and 1), and y_4, respectively, by OLS, LLR, and MRR2. Note that the OLS fits are smooth, but do not fit well at the several levels of the factor x_1. The LLR fits in Figures 8.1 and

8.4 are not very smooth and seem to "connect" the means of the responses y_1 and y_4 at the five levels of the factor x_1. However, the LLR fits in Figures 8.2 and 8.3 are too smooth to capture the dips and fit at the several levels of the factor x_1. A possible reason for some of the poor LLR fits, as shown in Figures 8.2 and 8.3, is that the data is sparse and the sample size is relatively small for the 2- or 3-dimensional design spaces, when compared to the 1-dimensional design space for the responses y_1 and y_4 in Figures 8.1 and 8.4. The MRR2 fits maintain the smoothness of OLS, while using LLR information to pull the fits closer to data where needed. Even in the 2- and 3-dimensional design spaces in Figures 8.2 and 8.3, the MRR2 fits are reasonably close to the sample means at each value of x_1 and much closer than many of the LLR fits. This illustrates the advantage of MRR2 over LLR for data from a sparse design.

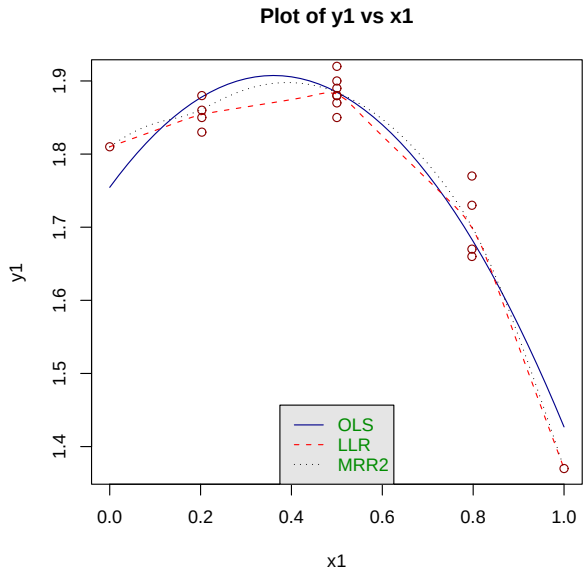

Figure 8.1: Comparison of plots of y_1 vs x_1 by OLS, LLR, and MRR2. [∘ ∘ ∘ Raw data]

8.1.2 Optimization Results Using the Desirability Function Method Under the OLS, LLR and MRR2 Methods

Shah, Montgomery and Carlyle (2004) use the Design-Expert software and the desirability function method to find an optimal location $\mathbf{x}$ that achieves the best value in terms of the

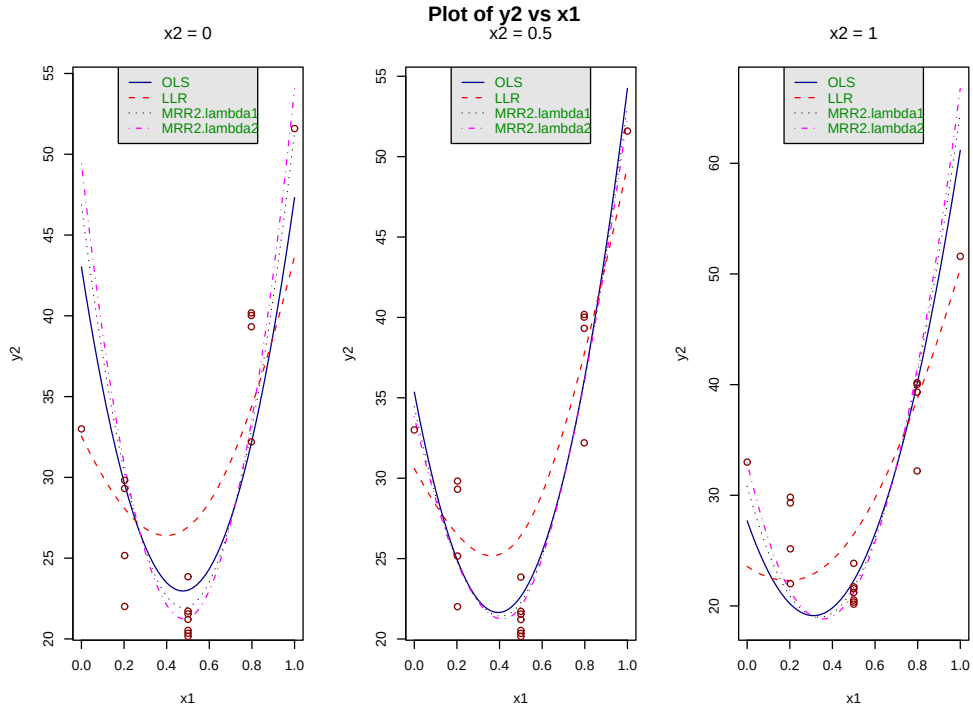

Figure 8.2: Comparison of plots of y_2 vs x_1 by OLS, LLR, MRR2$_{\lambda 1}$, and MRR2$_{\lambda 2}$, when $x_2 = 0$ (left), $x_2 = 0.5$ (center), and $x_2 = 1$ (right), respectively. [○ ○ ○ Raw data]

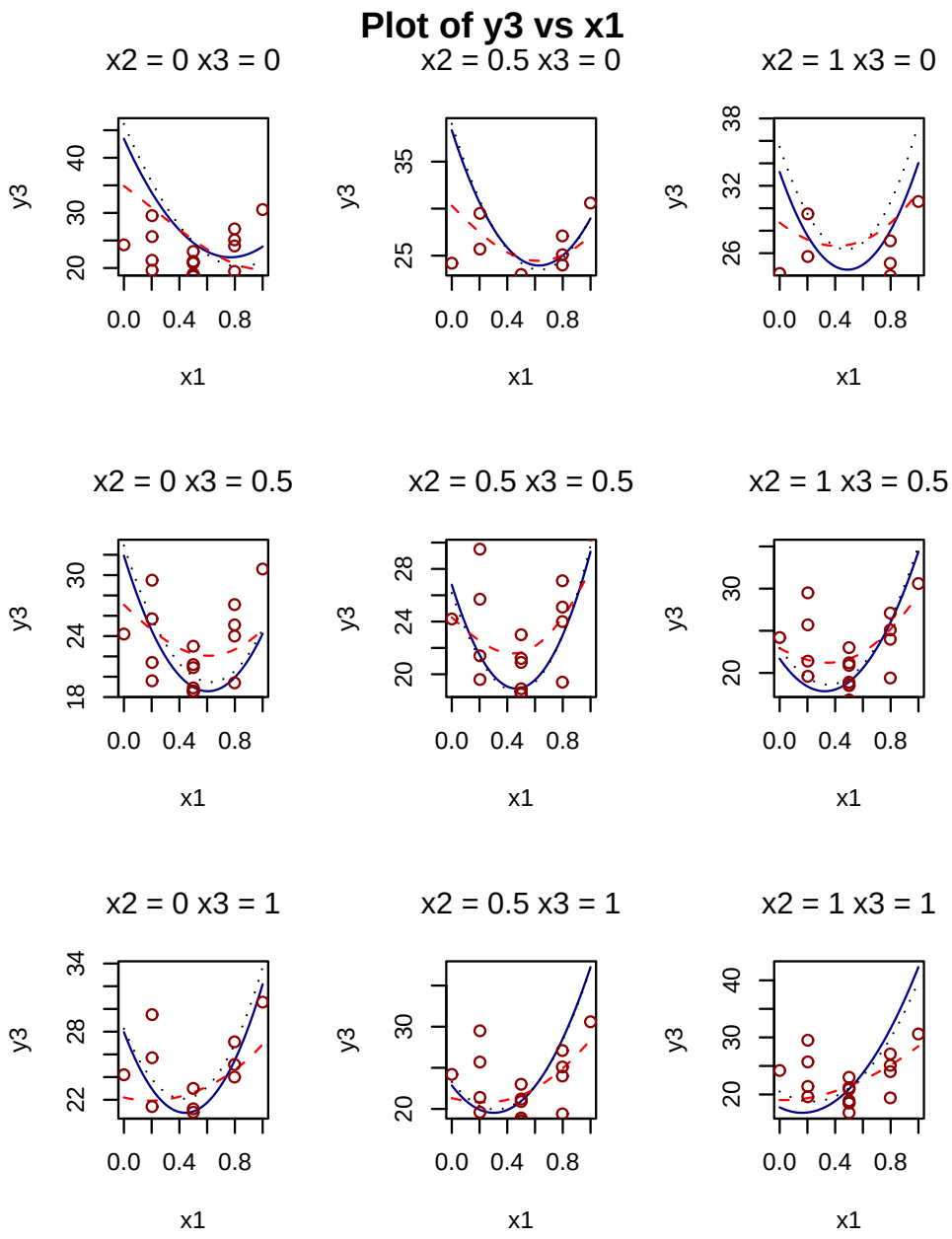

Figure 8.3: Comparison of plots of y_3 vs x_1 by OLS, LLR, and MRR2: top left: $x_2 = 0$ and $x_3 = 0$; top center: $x_2 = 0.5$ and $x_3 = 0$; top right: $x_2 = 1$ and $x_3 = 0$; middle left: $x_2 = 0$ and $x_3 = 0.5$; middle center: $x_2 = 0.5$ and $x_3 = 0.5$; middle right:$x_2 = 1$ and $x_3 = 0.5$; bottom left: $x_2 = 0$ and $x_3 = 1$; bottom center: $x_2 = 0.5$ and $x_3 = 1$; bottom right: $x_2 = 1$ and $x_3 = 1$. [∘ ∘ ∘ Raw data, solid line: OLS, dashed line: LLR, dotted line: MRR2]

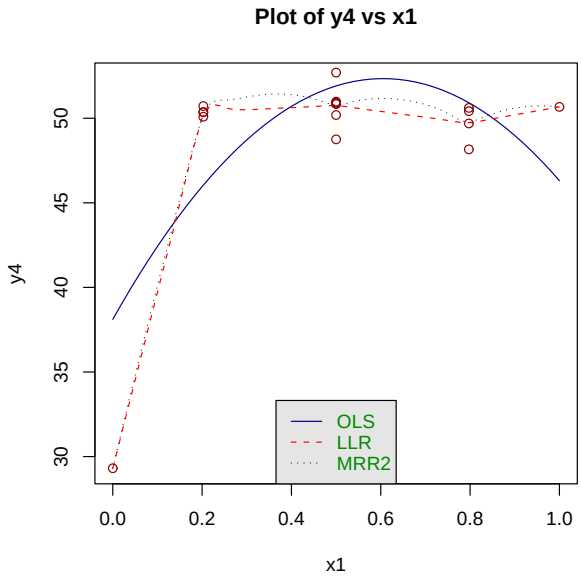

Figure 8.4: Comparison of plots of y_4 vs x_1 by OLS, LLR, and MRR2. [∘ ∘ ∘ Raw data]

desirability function D. The following conditions on the four responses they used are as follows.

springiness (y_1)	maximize ≥ 1.7
thiobarbituric acid number (y_2)	minimize ≤ 21
cooking loss (y_3)	minimize ≤ 20
whiteness index (y_4)	maximize ≥ 45

We use the maximum (or minimum) of the observed data as the T values (see pages 30-31 for definition of T) for each response, since the T values in individual desirabilities for each response are not given in Shah, Montgomery and Carlyle (2004). Thus, the T values in this case study are 1.92, 20.16, 16.80 and 50.98 for y_1, y_2, y_3 and y_4 respectively. The weights r, r_1, or r_2 in the individual desirability functions are all 1.0 in this case study, due to no priority. In addition, similar to Chapter 6, the solution vector $\mathbf{x}_s$ should be controlled within the experimental region R. That is, in the CCD design, which is a spherical design, the region constraint is $\mathbf{x}'_c\mathbf{x}_c \leq r^2$, where $\mathbf{x}_c = (x_{1c}, x_{2c}, ..., x_{kc})$ is transformed from $\mathbf{x}_s$ so that $\mathbf{x}_c$ is centered at zero, and r^2 is the squared design radius, and k, the number of independent variables, is three in this case. If x_{ic} is transformed into the range [-1.682, 1.682], then the r^2

in this example is three. The formulas of the desirability function and individual desirability have been given in Chapter 3.

As mentioned earlier, Shah, Montgomery and Carlyle (2004) use the second-order polynomial to parametrically model each response to obtain the optimal fitted value $\hat{y}_i(\mathbf{x})$ at location $\mathbf{x}$, for $i = 1, 2, 3, 4$. The final fitted models are given in Shah, Montgomery and Carlyle (2004) as well as the location where the simultaneous optimal solution is found. Based on the location they find using Design-Expert, the corresponding fitted values for the four responses are re-calculated by us as well as the desirability value D and given as follows.

The OLS solution:

$x_1 = 0.3514$ $x_2 = 0.7973$ $x_3 = 0.7319$ x_{c1} = -0.500 $x_{c2} = 1.000$ $x_{c3} = 0.780$

$y_1 = 1.9074$ $y_2 = 20.2910$ $y_3 = 17.6381$ $y_4 = 49.8346$ $D = 0.8301$

We use the MGA to find the optimization solutions by the three different modeling techniques. The parametric models we used for each response are exactly the same as the models in Shah, Montgomery and Carlyle (2004). The solutions we find by the OLS, LLR and MRR2 methods are as following.

The OLS solution:

$x_1 = 0.3857$ $x_2 = 0.9693$ $x_3 = 0.6784$ x_{c1} = -0.3844 $x_{c2} = 1.5786$ $x_{c3} = 0.6002$

$y_1 = 1.9067$ $y_2 = 19.7378$ $y_3 = 17.3903$ $y_4 = 50.4668$ $D = 0.9149$

The MRR2 solution:

$x_1 = 0.3379$ $x_2 = 0.9422$ $x_3 = 0.7080$ x_{c1} = -0.5452 $x_{c2} = 1.4877$ $x_{c3} = 0.6997$

$y_1 = 1.8947$ $y_2 = 19.5246$ $y_3 = 17.7507$ $y_4 = 51.3969$ $D = 0.8880$

We do not show the LLR solution, because the desirability function value D = 0. The reason that D = 0 is that the fits for the responses y_2 and y_3 by the LLR method can be highly inadequate as shown in Figures 8.2 and 8.3 (as mentioned in Section 8.1.1). The two dips are not captured well and the estimated smallest values for these two responses are obviously larger than the pre-selected acceptable maximum values (the L values in the

individual desirability functions), which are 21 and 20 for y_2 and y_3 respectively in this case study. Thus, the corresponding individual desirabilities (d_2 and d_3) cannot be greater than zero.

It is easy to see that our OLS solution found by the MGA is better than the OLS solution found by Design-Expert in Shah, Montgomery and Carlyle (2004). The OLS solution should not be compared to the MRR2 solution, because the OLS solution has been shown in Table 5.2 to be misspecified. However, the MRR2 solution should be viewed as more reliable than the OLS solution, because the MRR2 method provides a superior fit to the data than the OLS method in terms of our model comparison criteria.

Similar to what we have done in Figures 8.1 - 8.4 in Section 8.1.1, the surfaces of the D versus x_1 versus x_2 given at $x_3 = 0$, 0.5, and 1 are utilized to describe the 3-D surfaces of the desirability function. We also include the two optimal levels of x_3 ($x_3 = 0.68$ and 0.71) for the OLS and MRR2 methods respectively.

Figure 8.5 shows the surfaces and corresponding contours of the desirability function D by the OLS method with x_1 versus x_2 at $x_3 = 0.5$ and 0.68 (which is the optimal level for x_3 found by the MGA). The surfaces and contours of the function D at levels of $x_3 = 0$ and 1 are not shown due to flat surfaces and no contours. It is easy to see that there is only one peak for this desirability function D in this case study.

Figure 8.6 shows the surfaces and corresponding contours of the desirability function D by the MRR2 method with x_1 versus x_2 at $x_3 = 0.5$ and 0.71 (which is the optimal level for x_3 found by the MGA). Similar to the results for the OLS method, the surfaces and contours of the function D at levels of $x_3 = 0$ and 1 are not shown due to flat surfaces and no contours. It is also easy to see that there is only one peak for this desirability function D in this case study.

It is reasonable to think that the optimization results based on the different model assumptions should be different from each other. However, these differences due to model assumptions are not completely inconsistent. The surfaces and contours of the desirability function D in Figures 8.5 and 8.6 show that the results by the OLS and MRR2 have the similar shape and cover the similar area. Recall that the MRR2 fit combines the OLS fit with the LLR fit to the residuals of the OLS via the mixing parameter, λ. These surfaces and contours confirm that the results by MRR2 and OLS are consistent. But the results

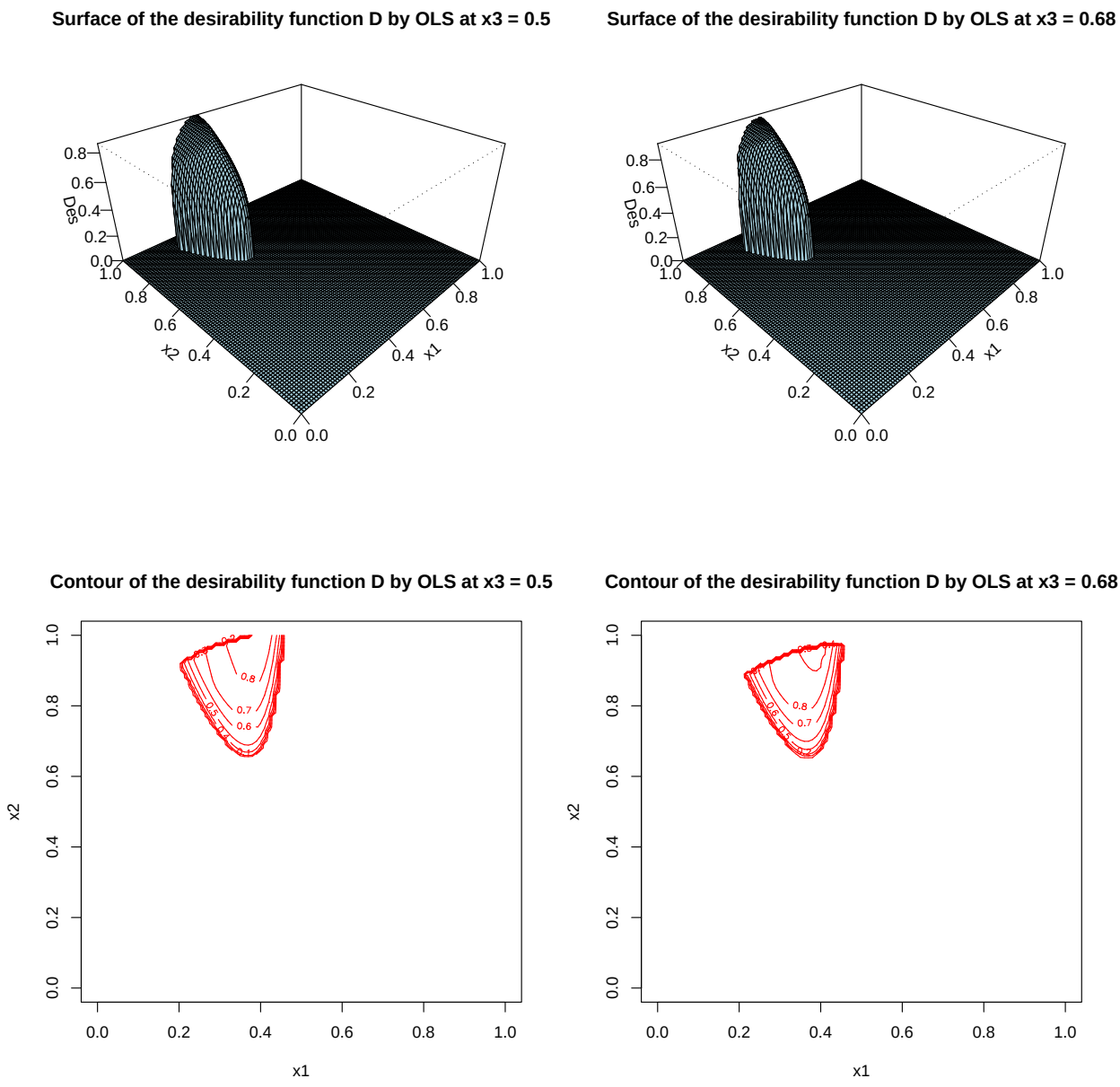

Figure 8.5: Surfaces and the corresponding contours of the desirability function D by the OLS method with x_1 versus x_2 at $x_3 = 0.5$ and 0.68

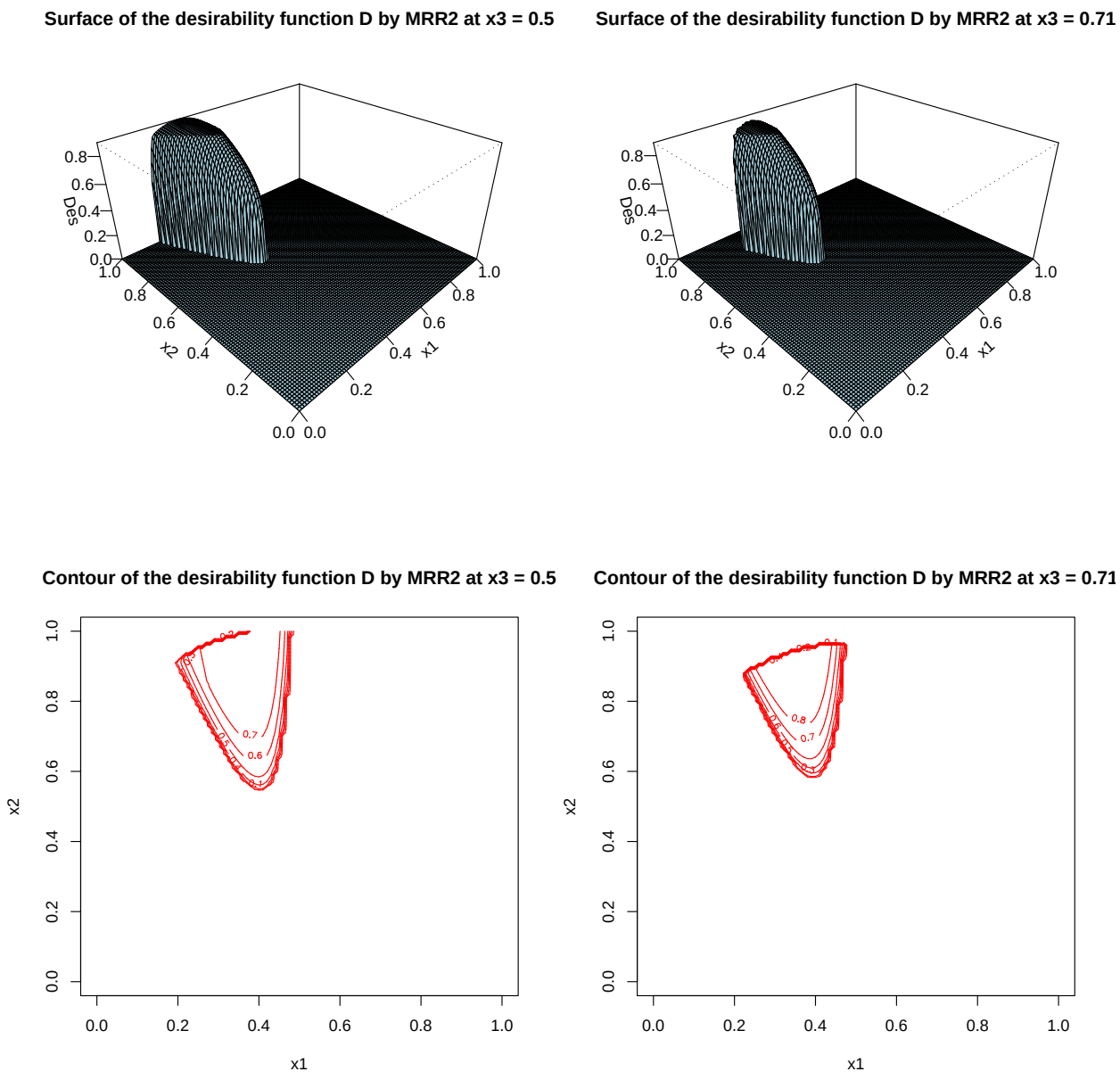

Figure 8.6: Surfaces and corresponding contours of the desirability function D by the MRR2 method with x_1 versus x_2 at $x_3 = 0.5$ and 0.71

by MRR2 should be more believable, due to having less model misspecification, as shown in Table 8.2, than by OLS.

8.2 Simulation Studies

In the minced fish quality example of the MRO problem with a CCD design, the semiparametric fit was observed to be highly competitive or superior to its parametric and nonparametric counterparts through all of the four responses in terms of the seven criteria. Under the different fits given by the OLS, LLR, $MRR2_{\lambda 1}$, and $MRR2_{\lambda 2}$ methods, respectively, the desirability function has been utilized to find the location which optimizes the multiple responses simultaneously. In this section, the MRO problem will be simulated via Monte Carlo simulations.

8.2.1 The MRO Goals and Simulation Process

To simplify the MRO problem, a CCD design with two factors and two responses will be simulated using two true underlying response functions. Like the univariate case for a CCD design with two factors in Pickle et al. (2006) and also similar to the multivariate case in the chemical process example with two factors in Myers and Montgomery (2002) mentioned in Section 6.1, the CCD will contain a total of 13 design points, including four axial runs and five center runs, which are shown in Table 8.3 for each simulated data set.

Each Monte Carlo simulation will be based on the following two underlying models:

$$\begin{aligned} \mathrm{y_{1i}} &= \mu_1(\mathbf{x}_i) + \varepsilon_{1i} \\ &= 66 + 22\mathrm{x_{1i}} + 10\mathrm{x_{2i}} + 13\mathrm{x_{1i}x_{2i}} - 23\mathrm{x_{1i}^2} - 25\mathrm{x_{2i}^2} \\ &\quad + \gamma[-2\sin(3\pi \mathrm{x_{i1}}) - 2\cos(3\pi \mathrm{x_{2i}}) + 2\sin(2\pi \mathrm{x_{1i}x_{2i}})] + \varepsilon_{1i}, \end{aligned} \tag{8.2}$$

$$\begin{aligned} \mathrm{y_{2i}} &= \mu_2(\mathbf{x}_i) + \varepsilon_{2i} \\ &= 70 - 15x_{1i} - 10\mathrm{x_{2i}} - 14\mathrm{x_{1i}x_{2i}} + 15\mathrm{x_{1i}^2} + 25\mathrm{x_{2i}^2} \\ &\quad + \gamma[2\sin(3\pi \mathrm{x_{1i}}) - 2\cos(3\pi \mathrm{x_{2i}}) + 2\sin(3\pi \mathrm{x_{1i}x_{2i}})] + \varepsilon_{2i}. \end{aligned} \tag{8.3}$$

where $\mu_1(.)$ and $\mu_2(.)$ are the true mean functions of y_1 and y_2, respectively, $\mathbf{x}_i = (x_{1i}, x_{2i})'$ is the i^{th} design point, the two error terms, $(\varepsilon_{1i}, \varepsilon_{2i})$, are independent normally distributed random variables with means of zero and variances of one, $i = 1, ..., n$, with $n = 13$. In both models (8.2) and (8.3), γ represents the model misspecification parameter. That is, the user's models will be represented by (8.1) and (8.2) with $\gamma = 0$. As the value of γ increases, the amount of misspecification increases in the models. Five degrees of model misspecification will be studied ($\gamma = 0.00$, 0.25, 0.5, 0.75, and 1.00), where $\gamma = 0$ represents a correctly specific model. According to these five levels of γ, there will be five Monte Carlo simulations respectively, in each of which 500 simulated data sets will be generated.

Table 8.3: Design points of a CCD for each simulated data set

i	x_1	x_2
1	0.8536	0.8536
2	0.1464	0.8536
3	0.8536	0.1464
4	0.1464	0.1464
5	1.0000	0.5000
6	0.0000	0.5000
7	0.5000	1.0000
8	0.5000	0.0000
9	0.5000	0.5000
10	0.5000	0.5000
11	0.5000	0.5000
12	0.5000	0.5000
13	0.5000	0.5000

Figure 8.7 shows the surfaces of the true mean function of the response y_1 when $\gamma = 0.00$, 0.25, 0.50, 0.75, and 1.00, respectively. Similarly, Figure 8.8 shows the surfaces of the true mean function of the response y_2. Both figures show that as γ increases, the curvatures of the surfaces becomes more pronounced.

Two MRO goals are used for each Monte Carlo simulation to do a comparison of OLS, LLR, and MMR2. Goal 1 is to maximize y_1 and minimize y_2 simultaneously. Goal 2 is to achieve some target value for y_1 and some target value for y_2 simultaneously. Each goal will be solved

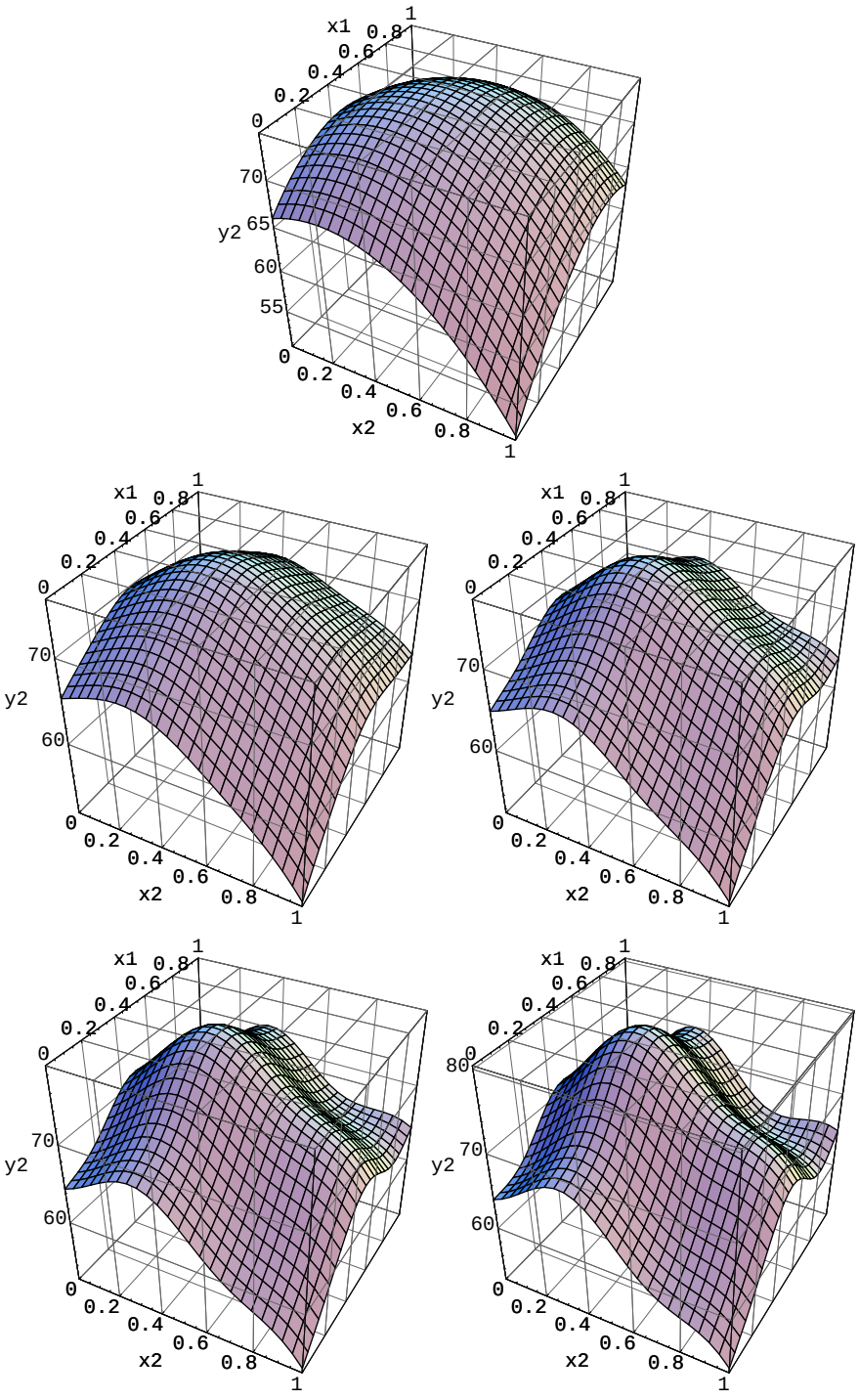

Figure 8.7: Surfaces for the true mean function of the response y_1 when $\gamma = 0.00$ (top one), 0.25 (middle left), 0.50 (middle right), 0.75 (bottom left), and 1.00 (bottom right), respectively.

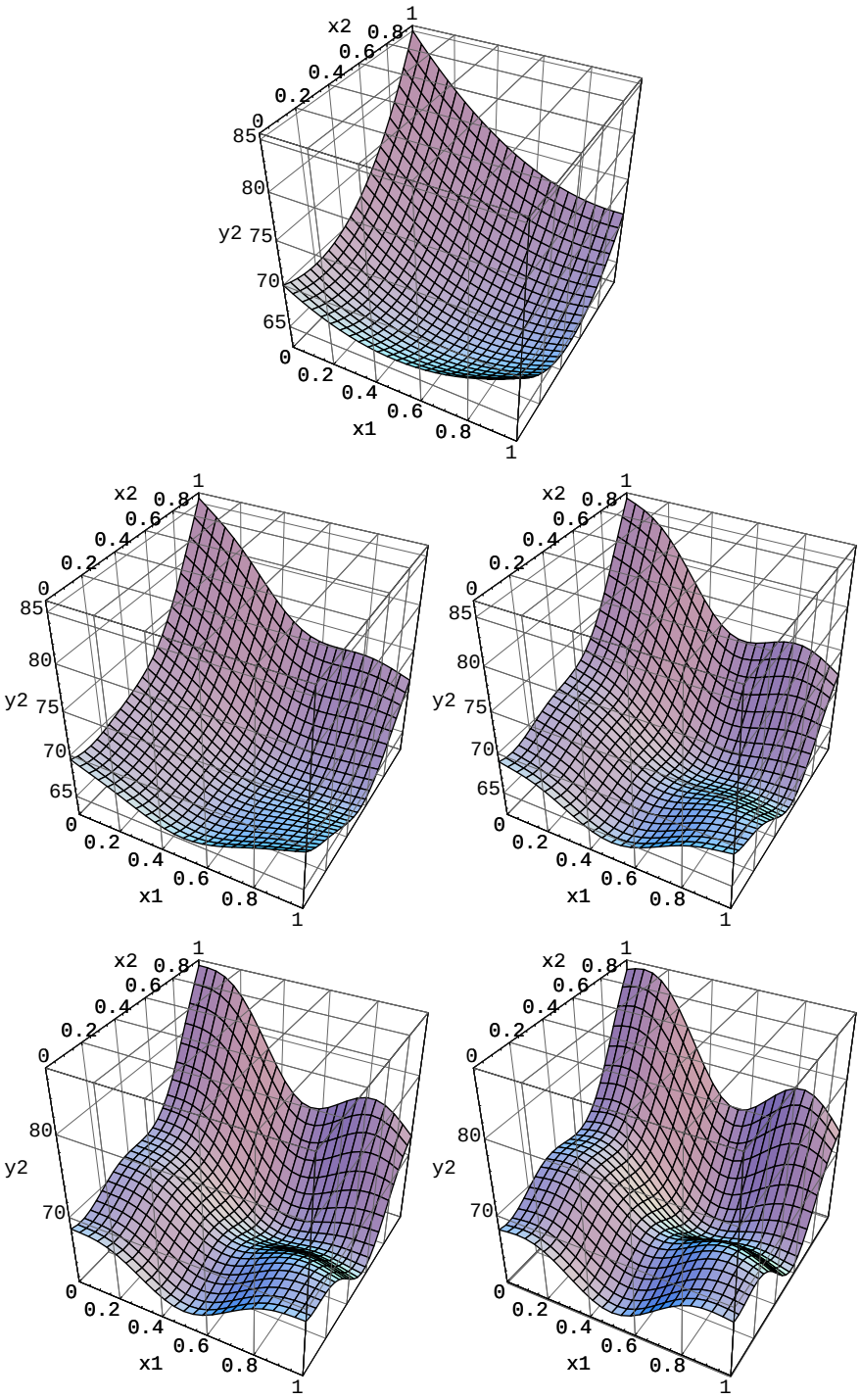

Figure 8.8: Surfaces for the true mean function of the response y_2 when $\gamma = 0.00$ (top one), 0.25 (middle left), 0.50 (middle right), 0.75 (bottom left), and 1.00 (bottom right), respectively.

by the desirability function method with some required and pre-specified parameters, such as T, L, U, indicated in Equations 3.1 - 3.4. For Goal 1, we choose $T_1 = 83$ and $L_1 = 65$ for y_1, and $T_2 = 58$ and $U_2 = 75$ for y_2, while in the second goal, we choose $T_1 = 60$, $L_1 = 55$, and $U_1 = 65$ for y_1 and $T_2 = 75$, $L_2 = 70$, and $U_2 = 80$ for y_2. All these parameters are fixed across all levels of γ.

8.2.2 One Simulation Criterion During The Modeling Stage

After completion of the data generation stage, the modeling stage begins using the OLS, LLR and MRR2 methods respectively to model the m responses, where m is 2, the number of the responses in this study. Similar to the univariate case in Pickle et al. (2006), for each simulation, we will compare the MRR2 method with the OLS and LLR methods in terms of simulated integrated mean squared error (SIMSE). The SIMSE in the multivariate case is given by:

$$SIMSE = \frac{\sum ASE}{500}, \tag{8.4}$$

where

$$ASE = \frac{\sum_{l=1}^{s} (\mu(\mathbf{x}_l) - \hat{\mathbf{y}}(\mathbf{x}_l))'(\mu(\mathbf{x}_l) - \hat{\mathbf{y}}(\mathbf{x}_l))}{s}, \tag{8.5}$$

where ASE denotes the average squared error for the estimates from the true mean functions for each of the 500 simulated data sets, $\mu(\mathbf{x}_l) = (\mu_1(\mathbf{x}_l), \mu_2(\mathbf{x}_l))'$ is the true mean functions of y_1 and y_2 at location $\mathbf{x}_l = (x_{1l}, x_{2l})'$, $\hat{\mathbf{y}}(\mathbf{x}_l) = (\hat{y}_1(\mathbf{x}_l), \hat{y}_2(\mathbf{x}_l))'$ is the fits at $\mathbf{x}_l$ by OLS, LLR, or MRR2, $l = 1, ..., s$, and s is the number of locations within the experimental space used for prediction. When the ASE is calculated based on the 41×41 uniform grid of points, those points outside of the experimental space are excluded. In this study, $s = 1257$. SIMSE provides an indication of the fit performance of each of the three methods over the entire design space.

8.2.3 Two Simulation Criteria During The Optimization Stage

After completion of the modeling stage, the optimization stage begins using the desirability function method to find an optimal solution. In practice, the optimization stage is concerned with finding an optimal location, not an optimal fit for each response. Therefore, there are

two criteria used to compare the three methods, both of which are related to an optimal location found by MGA_4 through the desirability function for each simulated data set.

For each Monte Carlo simulation, the three methods will be compared in terms of the average squared error loss (ASEL) from the true target response values. The ASEL is given by

$$ASEL = \frac{\sum_{q=1}^{q=500} (\mu(\mathbf{x}_q^*) - \mathbf{T})'(\mu(\mathbf{x}_q^*) - \mathbf{T})}{500}, \tag{8.6}$$

where $\mu(\mathbf{x}_q^*)$ is a 2×1 vector of the values of the true mean function at an optimal location $\mathbf{x}_q^*$, the location $\mathbf{x}_q^* = (x_{1q}^*, x_{2q}^*)'$ is obtained by MGA_4 through the desirability function for the q^{th} simulated data set, $q = 1, ..., 500$, and $\mathbf{T}$ is a 2×1 vector of the target values for the responses. For Goal 1, as mentioned in Section 8.2.1, $\mathbf{T} = (83, 58)'$, and for Goal 2, $\mathbf{T} = (60, 75)'$.

For each Monte Carlo simulation, the three methods will also be compared in terms of average desirability function (AD), which is given by

$$AD = \frac{\sum_{q=1}^{500} D_q}{500}, \tag{8.7}$$

where

$$D_q = (d_1(\mathbf{x}_q^*) d_2(\mathbf{x}_q^*))^{1/2}, \tag{8.8}$$

and similar to Equation 8.6, $\mathbf{x}_q^*$ is an optimal location obtained by MGA_4 through the desirability function for the q^{th} simulated data set, d_1 and d_2 are individual desirabilities for y_1 and y_2, respectively. Essentially, both ASEL and AD measure the performance of the locations chosen by each modeling method. We measure with ASEL the average Euclidean distance of the mean response vector from the target vector at the chosen locations for each method. And, with AD we measure the average desirability of the chosen functions. Of course, we prefer a method with a small value of ASEL and a large value of AD.

Figure 8.9 provides the surfaces of the desirability function for Goal 1 using the two true mean functions (shown in Equations 8.2 and 8.3) for the varying degrees of model misspecification. Similarly, Figure 8.10 provides the surfaces of the desirability function for Goal 2. All these surfaces have only one big "mountain", which means that there will be only a single optimal solution in terms of the desirability function for each level of γ.

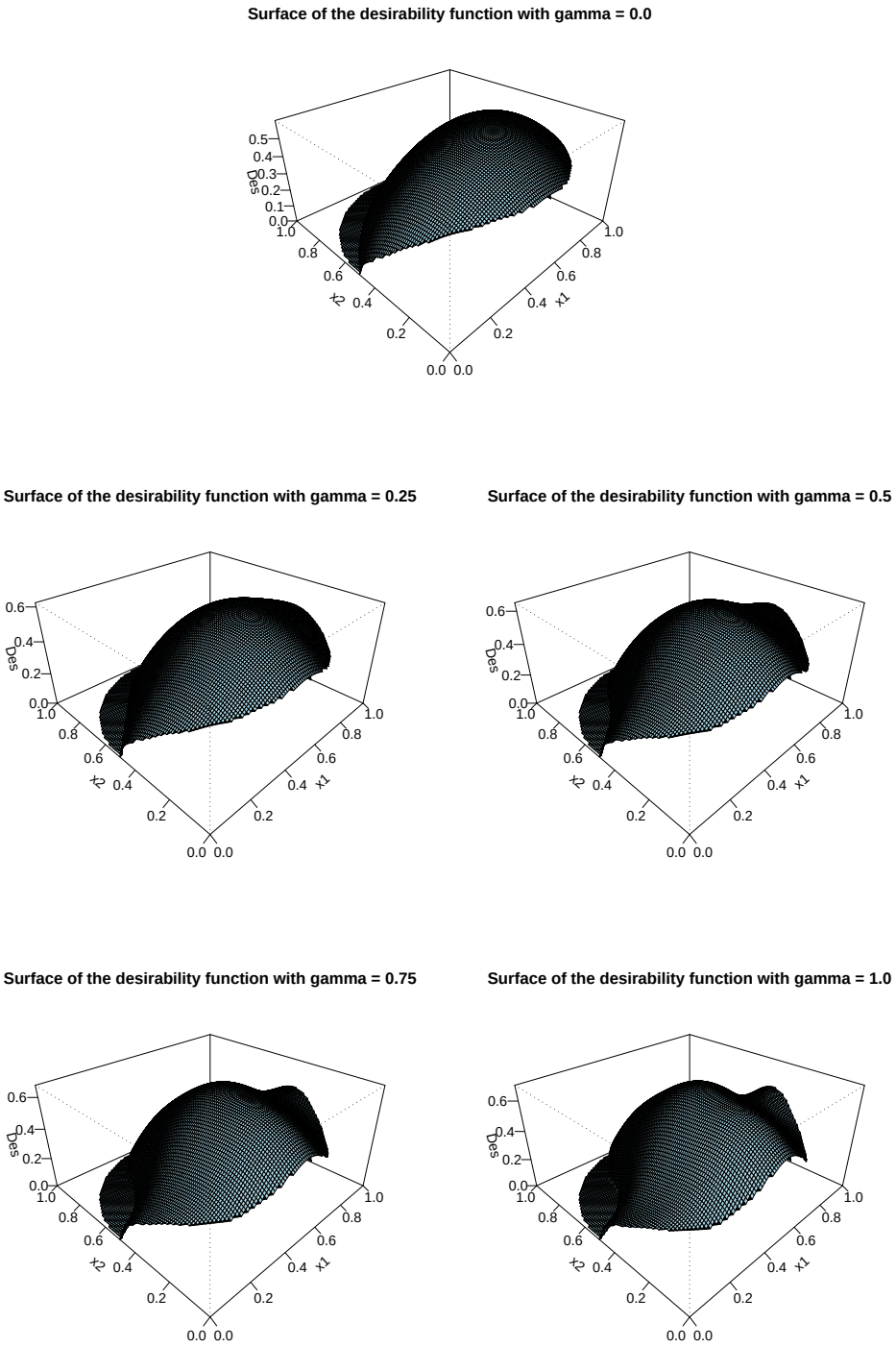

Figure 8.9: Surfaces of the desirability function for Goal 1 using the two true mean functions (aa shown in Equations 8.2 and 8.3) when $\gamma = 0.00$ (top one), 0.25 (middle left), 0.50 (middle right), 0.75 (bottom left), and 1.00 (bottom right), respectively.

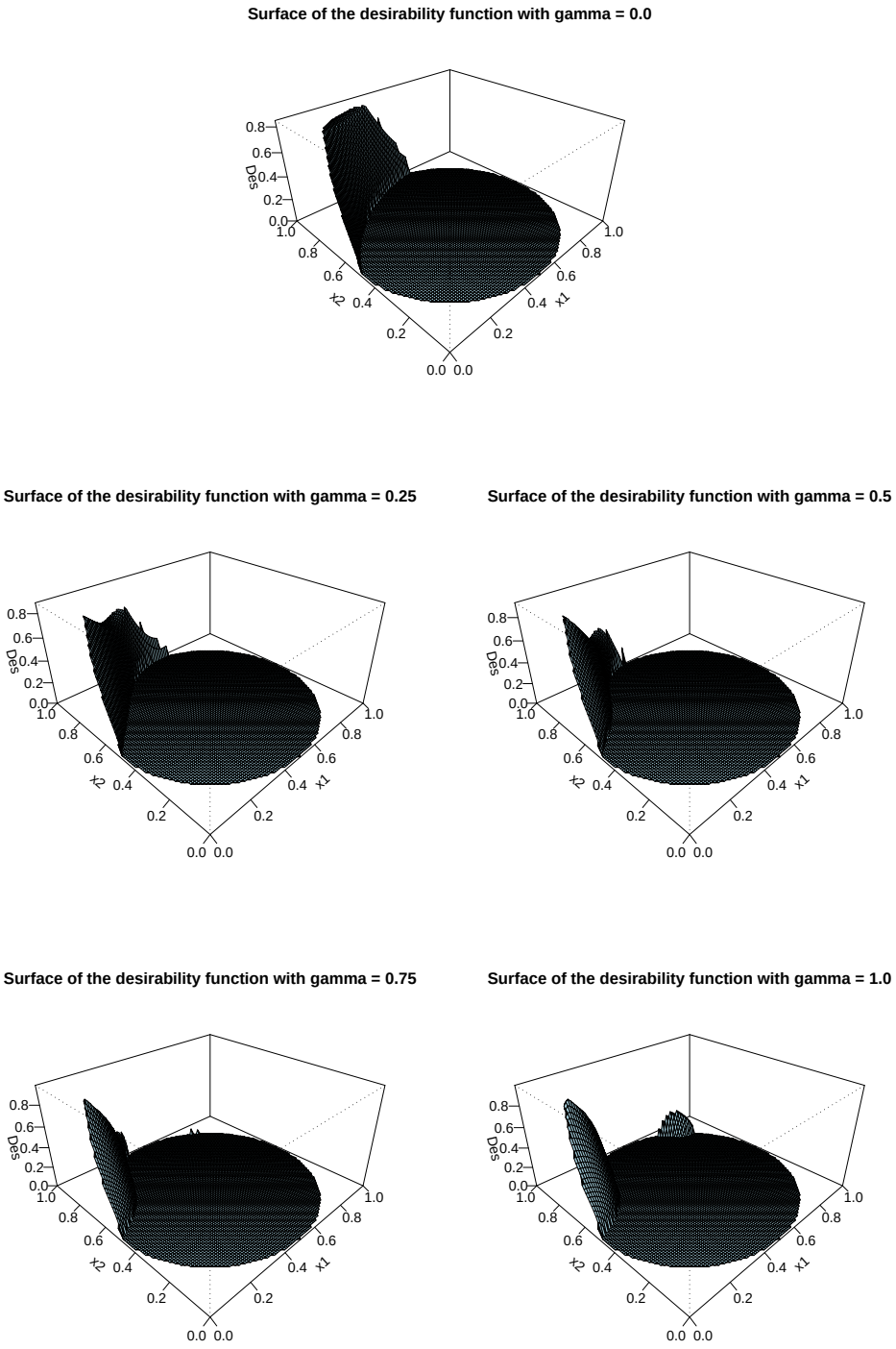

Figure 8.10: Surfaces of the desirability function for Goal 2 using the two true mean functions (as shown in Equations 8.2 and 8.3) when $\gamma = 0.00$ (top one), 0.25 (middle left), 0.50 (middle right), 0.75 (bottom left), and 1.00 (bottom right), respectively.

Table 8.4 provides the true optimal solutions for Goal 1, using the true mean functions, for each level of γ. In the table, Columns 2-3 represent the true optimal location, $\mathbf{x}^{*'}_{opt} = (x^*_{opt1}, x^*_{opt2})$. Columns 4-5 represent the values of the true mean functions of y_1 and y_2 at $\mathbf{x}^*_{opt}$, respectively. The last column is the value of the desirability function, $D^*_{opt} = (d_1(\mathbf{x}^*_{opt})d_2(\mathbf{x}^*_{opt}))^{1/2}$. Similar to Table 8.4, Table 8.5 provides the true optimal solutions for Goal 2. It is easy to see that the optimization results in Tables 8.4 and 8.5 match well with Figures 8.9 and 8.10, respectively.

Table 8.4: True optimal solutions for Goal 1 for the varying degrees of model misspecification using the true mean functions.

γ	x^*_{opt1}	x^*_{opt2}	$\mu_1(\mathbf{x}^*_{opt})$	$\mu_2(\mathbf{x}^*_{opt})$	D^*_{opt}
0.00	0.6103	0.3639	74.0754	62.9948	0.5967
0.25	0.5602	0.3638	75.4653	63.6494	0.6231
0.50	0.5456	0.3795	76.8226	64.1142	0.6485
0.75	0.5427	0.4160	77.8638	64.2434	0.6725
1.00	0.5429	0.4566	78.3741	63.8511	0.6981

Table 8.5: True optimal solutions for Goal 2 for the varying degrees of model misspecification using the true mean functions.

γ	x^*_{opt1}	x^*_{opt1}	$\mu_1(\mathbf{x}^*_{opt})$	$\mu_2(\mathbf{x}^*_{opt})$	D^*_{opt}
0.00	0.0759	0.7649	61.3133	75.1137	0.848873
0.25	0.0651	0.7459	61.0107	75.0002	0.893217
0.50	0.0619	0.7407	60.5261	75.0007	0.945863
0.75	0.0601	0.7378	60.0171	75.0007	0.998213
1.00	0.0803	0.7337	60.0000	75.0001	0.999989

8.2.4 Simulation Results During The Modeling Stage

Table 8.6 provides a comparison of the OLS, LLR, MRR2$_{\lambda 1}$, and MRR2$_{\lambda 2}$ based on the SIMSE values for the varying degrees of model misspecification in the simulations based on the CCD as shown in Table 8.3. Table 8.6 also includes the Monte Carlo errors of the SIMSE

values. For the scenario in which the researchers correctly specify the form of the underlying models (i.e., $\gamma = 0.00$), we would expect the parametric approach to be superior. The first row of Table 8.6 shows that the parametric approach performs the best as it yields a SIMSE value of 0.6330. The semiparametric approaches, $\text{MRR2}_{lambda1}$ and $\text{MRR2}_{lambda2}$, are a close second and third with values of 0.6668 and 0.6720, respectively, whereas the nonparametric fit is much worse with a SIMSE value of 5.2171.

Table 8.6: Simulated integrated mean squared error (SIMSE) values by OLS, LLR, $\text{MRR2}_{\lambda 1}$, and $\text{MRR2}_{\lambda 2}$ in the simulations based on CCD and the estimated Monte Carlo (MC) error of SIMSE. Best values in bold.

	SIMSE				MC error(SIMSE)			
γ	OLS	LLR	$\text{MRR2}_{\lambda 1}$	$\text{MRR2}_{\lambda 2}$	OLS	LLR	$\text{MRR2}_{\lambda 1}$	$\text{MRR2}_{\lambda 2}$
0.00	**0.6330**	5.2171	0.6668	0.6720	0.0120	0.0491	0.0123	0.0123
0.25	1.4435	5.6521	**1.4097**	1.4161	0.0181	0.0472	0.0183	0.0184
0.50	3.8424	7.4220	3.5061	**3.4809**	0.0293	0.0482	0.0299	0.0302
0.75	7.8296	10.2053	6.9378	**6.8348**	0.0417	0.0551	0.0426	0.0430
1.00	13.4051	14.0769	11.7393	**11.5172**	0.0545	0.0578	0.0555	0.0558

The remaining rows of Table 8.6 provide the SIMSE values for the scenario in which the researchers misspecify the models (i.e., $\gamma > 0$). Both $\text{MRR2}_{\lambda 1}$ and $\text{MRR2}_{\lambda 2}$ perform better than OLS and LLR with smaller SIMSE values through each non-zero degree of misspecification. The poor performance of the nonparametric method is most likely due to the sparsity of the data and the small sample size. Thus, we conclude that the semiparametric approaches (both $\text{MRR2}_{\lambda 1}$ and $\text{MRR2}_{\lambda 2}$) are highly competitive to the parametric approach when no model misspecification, and always superior to both the parametric and nonparametric approaches when there exists some moderate model misspecification.

Table 8.6 also shows that $\text{MRR2}_{\lambda 1}$ and $\text{MRR2}_{\lambda 2}$ have close SIMSE values across all levels of γ. It means that the semiparametric approach with the data-driven method (PRESS**) is almost equivalent to the semiparametric approach with the estimated asymptotically optimal data driven method, even if the sample size is small (n=13 with two factors). Therefore, we will only use the $\text{MRR2}_{\lambda 2}$ fit for optimization, not the $\text{MRR2}_{\lambda 1}$, because the estimated asymptotically optimal data driven method (for $\text{MRR2}_{\lambda 2}$) is more computationally efficient than the data-driven method using PRESS** (for $\text{MRR2}_{\lambda 1}$).

8.2.5 Simulation Results During The Optimization Stage

Table 8.7 provides a comparison of OLS, LLR, and MRR2$_{\lambda 2}$ in terms of the ASEL and AD values for the varying degrees of model misspecification for Goal 1 in the simulations based on CCD during the optimization stage. When $\gamma = 0.00$ or 0.25, OLS performs the best because it has the smallest ASEL values and the biggest AD values than LLR and MRR2$_{\lambda 2}$. When γ is 0.50 or 0.75, MRR2$_{\lambda 2}$ performs the best because it has the smallest ASEL values and the biggest AD values. The ASEL values (or the AD values) by MRR2$_{\lambda 2}$ may be not smaller (or larger) than the ones by the other two methods by a least two Monte Carlo error. When γ is 1.00, LLR performs the best.

Table 8.7: Average squared error loss (ASEL) and averaged desirability function (AD) values by OLS, LLR, and $MRR2_{\lambda 2}$ for Goal 1 in the simulations based on CCD, with the ranges of the estimated Monte Carlo errors of ASEL and AD values (0.0017, 0.0200) and (6.5×10^{-5}, 8.4×10^{-4}), respectively. Best values in bold.

	ASEL			AD		
γ	OLS	LLR	MRR2$_{\lambda 2}$	OLS	LLR	MRR2$_{\lambda 2}$
0.00	**10.2591**	10.4696	10.2643	**0.5954**	0.5868	0.5952
0.25	**9.5246**	9.7425	9.5286	**0.6192**	0.6103	0.6190
0.50	8.8613	9.0983	**8.8470**	0.6419	0.6322	**0.6426**
0.75	8.3301	8.3231	**8.2930**	0.6627	0.6629	**0.6645**
1.00	7.9774	**7.5698**	7.9352	0.6815	**0.6952**	0.6840

The three modeling methods actually all have close ASEL and AD values as shown in Table 8.7. It seems some relationship across methods between SIMSE from the modeling stage and ASEL and AD from the optimization stage: the smaller value of SIMSE yields the smaller value of ASEL and the larger value of AD in most cases. One reason for this weak relationship is that however poor the fit (say, the LLR fit) may be, if a location obtained based on the fit is closer to the true location which achieves the best compromise, then the location can make the corresponding values of ASEL smaller and AD larger. This reason can be seen in Figures 8.11 and 8.12.

Figure 8.11 provides the plots of y_1 vs. x_2 at $x_1 = 0.25$, 0.5, and 0.75, by OLS, LLR, and MRR2$_{\lambda 2}$, and the true mean function of y_1, respectively, where the response data of y_1 come

from the true mean function (8.2) with $\gamma = 1.00$, based on CCD. Recall that Goal 1 is to maximize y_1 and minimize y_2 simultaneously. Figure 8.11 shows that the locations by OLS, LLR, and MRR2$_{\lambda 2}$, which achieve the maximum of y_1, are close to each other, although the LLR fit is far away from the curve of the true mean function of y_1. Similar to Figure 8.11, Figure 8.12 provide the plots of y_2 vs. x_2 at $x_1 = 0.25$, 0.5, and 0.75, by OLS, LLR, and MRR2$_{\lambda 2}$, and the true mean function of y_2, respectively, given $\gamma = 1.00$. It shows that the locations by OLS, LLR, and MRR2$_{\lambda 2}$, which achieve the minimum of y_2, are close to each other, although the LLR fit is very far away from the curve of the true mean function of y_2.

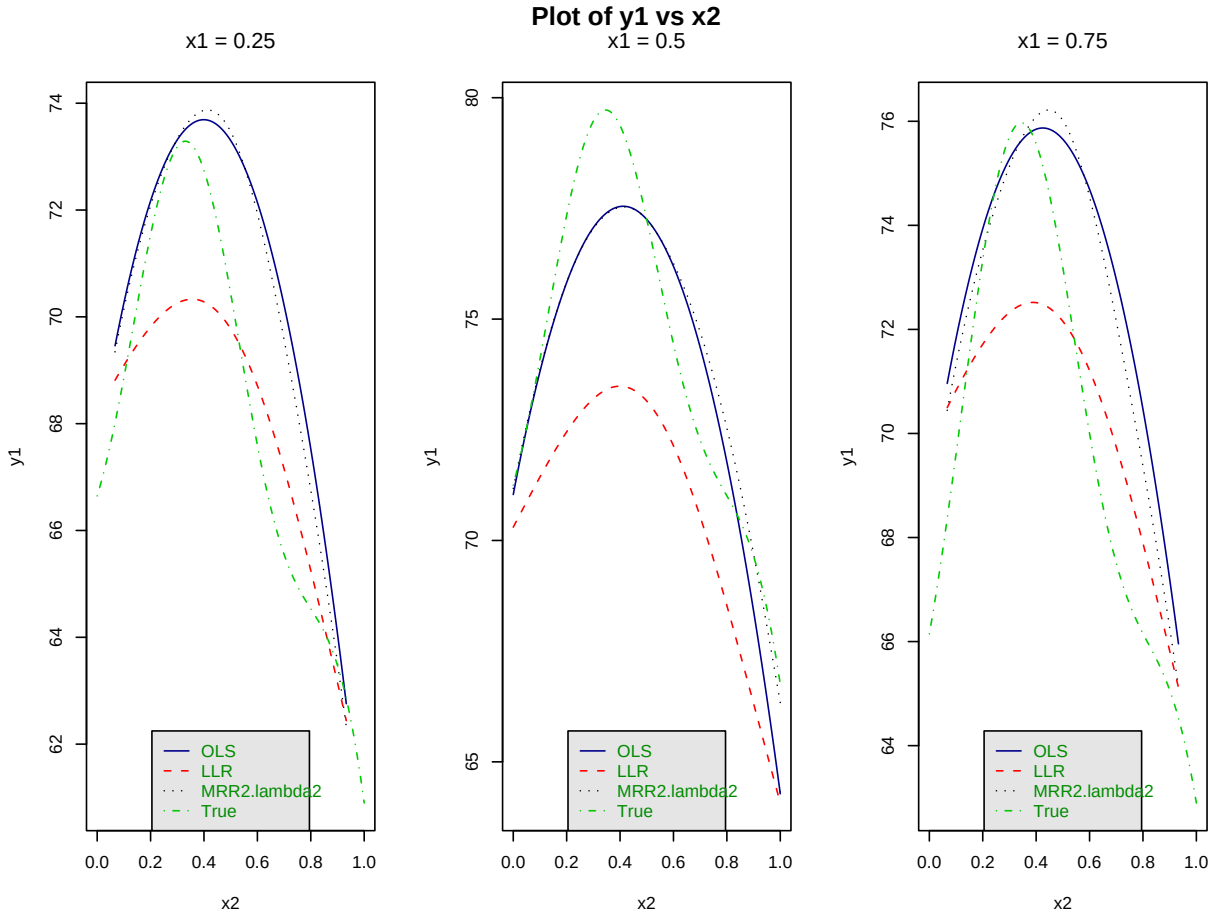

Figure 8.11: Comparison of plots of y_1 vs. x_2 by OLS, LLR, and MRR2$_{\lambda 2}$, and the true mean function of y_1, respectively, where the response data of y_1 come from the true mean function (8.2) with $\gamma = 1.00$ based on CCD: left: x_1= 0.25; center: $x_1 = 0.5$; right: $x_1 = 0.75$.

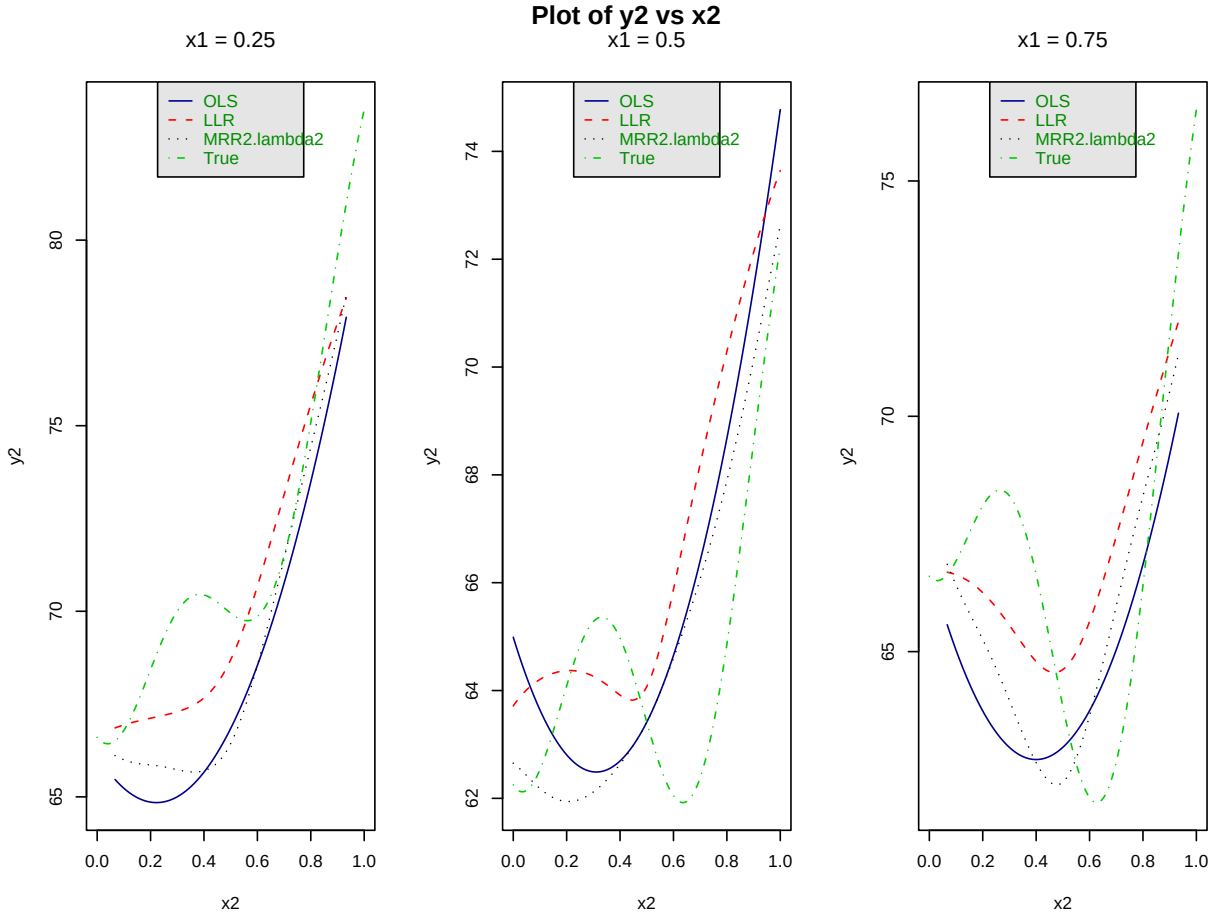

Figure 8.12: Comparison of plots of y_2 vs. x_2 by OLS, LLR, and MRR2$_{\lambda 2}$, and the true mean function of y_2, respectively, where the response data of y_2 come from the true mean function (8.3) with $\gamma = 1.00$ based on CCD: left: x_1= 0.25; center: $x_1 = 0.5$; right: $x_1 = 0.75$.

Similar to Table 8.7, Table 8.8 provides a comparison of the three modeling methods in terms of the ASEL and AD values for the varying degrees of model misspecification for Goal 2. When $\gamma = 0.00$, LLR performs better than OLS and MRR2$_{\lambda 2}$ since it has the smallest ASEL value and the biggest AD value. But LLR only performs slightly better, since actually, all of the three methods have nearly equal values of ASEL and AD. When $\gamma > 0.00$, however, MRR2$_{\lambda 2}$ performs better than the other two, since it has smaller ASEL values and bigger AD values. The ASEL values (or the AD values) by MRR2$_{\lambda 2}$ may be not smaller (or larger) than the ones by the other two methods by a least two Monte Carlo error. As the value of γ increases, the advantage of MRR2$_{\lambda 2}$ increases over the other two. Like Table 8.7, Table 8.8 shows some relationship across methods between SIMSE and ASEL and AD as in general: the smaller value of SIMSE yields the smaller value of ASEL and the larger value of AD, except for LLR at $\gamma = 0$.

Table 8.8: ASEL and AD values by OLS, LLR, and MRR2$_{\lambda 2}$ for Goal 2 in the simulations based on CCD, with the ranges of the Monte Carlo errors of ASEL and AD values (0.0164, 0.0758) and (0.0136, 0.0021), respectively. Best values in bold.

	ASEL			AD		
γ	OLS	LLR	MRR2$_{\lambda 2}$	OLS	LLR	MRR2$_{\lambda 2}$
0.00	1.5741	**1.5327**	1.5637	0.7964	**0.7996**	0.7975
0.25	2.0139	2.0071	**1.9387**	0.7275	0.7298	**0.7387**
0.50	2.4658	2.3030	**2.1151**	0.6434	0.6745	**0.7017**
0.75	3.2945	3.0537	**2.5123**	0.5028	0.5458	**0.6441**
1.00	4.0907	3.9122	**2.9282**	0.4209	0.4357	**0.6063**

For Goal 1, as shown in Table 8.7, the optimization results by the semiparametric fit are highly competitive to the results by the other two methods when there is no or low model misspecification, and superior or highly competitive to the results by the other two methods when there are moderate model misspecification. For Goal 2, as shown in Table 8.8, the optimization results by the semiparametric fit are highly competitive to the results by the other two methods when there is no model misspecification, and always superior to the results by the other two methods when there exists model misspecification with different degrees. Thus, we can conclude that the optimization results by the semiparametric approach are more reliable than the ones by the parametric and nonparametric approaches in general.

8.2.6 Some Further Discussion

Although Table 8.6 shows that MRR2 is highly competitive or superior to OLS and LLR over the entire range of γ values, MRR2 does not enjoy a large advantage over OLS in terms of SIMSE because a CCD is designed to have several nice properties when fitting a second-order polynomial model by OLS. A CCD space is too sparse to capture the important features of a surface of interest for models of the type considered here when γ is large. A second design will now be considered, a design that places points in the interior of the CCD design space. This "space-filling design" (SFD) may not be any optimal design, but simply spreads out the five central runs within the interior of the CCD to capture more structure of the response surface. Table 8.9 provides the 13 design points of the SFD. It shows that only the last four rows (design points) are different from the original CCD. Figure 8.13 shows the 13 design points in the experimental space of the SFD.

Table 8.9: Design points of a space-filling design (SFD) modified from the CCD in this study

i	x_1	x_2
1	0.8536	0.8536
2	0.1464	0.8536
3	0.8536	0.1464
4	0.1464	0.1464
5	1.0000	0.5000
6	0.0000	0.5000
7	0.5000	1.0000
8	0.5000	0.0000
9	0.5000	0.5000
10	0.3232	0.3232
11	0.3232	0.6768
12	0.6768	0.3232
13	0.6768	0.6768

Based on the SFD, we repeat the simulation studies of Section 8.2.1. There are also five Monte Carlo simulations with respect to the five levels of γ, in each of which 500 simulated data sets are generated. All of the data sets are also generated, based on the two underlying

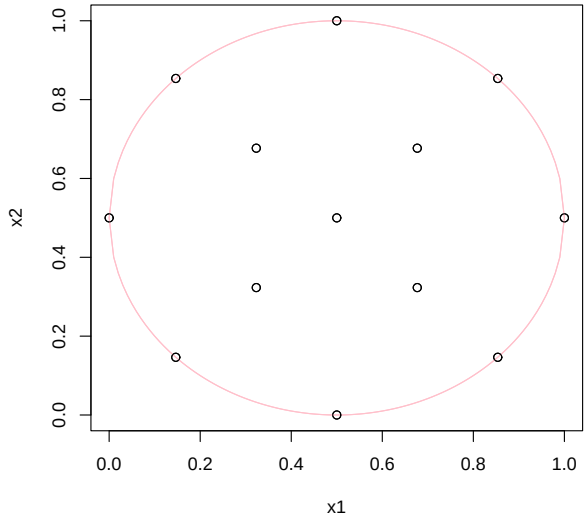

Figure 8.13: Design points in the experimental space of a space-filling design (SFD) modified from the CCD in this study.

models given in Equations 8.2 and 8.3. Each data set is also fitted by OLS, LLR, and $\text{MRR}_{\lambda 2}$. The three simulation criteria, SIMSE, ASEL, and AD, are also used to compare the three methods.

Similar to Table 8.6, Table 8.10 provides a comparison of OLS, LLR, and $\text{MRR2}_{\lambda 2}$ in terms of SIMSE for the varying degrees of model misspecification in the simulations based on the SFD. When $\gamma = 0$, OLS is the best with the smallest SIMSE, slightly smaller than the SIMSE value for $\text{MRR2}_{\lambda 2}$. When $\gamma > 0$, $\text{MRR2}_{\lambda 2}$ is always the best with the smallest SIMSE values across all non-zero degrees of misspecification. The results in Table 8.10 described above are quite similar to the ones in Table 8.6. The difference between these two tables is that MRR2 makes a much bigger improvement over the other two methods when there exists some model misspecification, especially when the value of γ is relatively large. This result implies that $\text{MRR2}_{\lambda 2}$ obtains greater benefits from the SFD than from the CCD. However, LLR in Table 8.10 is worse than LLR in Table 8.6 in terms of SIMSE, when the value of γ is relatively large. The reason is that the sample size is 13 in both designs and LLR fits are highly variable when the sample size is small.

Table 8.10: SIMSE values by OLS, LLR, and $MRR2_{\lambda 2}$ in the simulations based on SFD and the estimated Monte Carlo (MC) errors of the SIMSE values. Best values in bold.

	SIMSE			MC error(SIMSE)		
γ	OLS	LLR	$MRR2_{\lambda 2}$	OLS	LLR	$MRR2_{\lambda 2}$
0.00	**0.6489**	4.9717	0.7252	0.0127	0.0471	0.0131
0.25	1.2530	6.4235	**1.1364**	0.0147	0.0497	0.0153
0.50	3.0467	9.0253	**2.3309**	0.0190	0.0529	0.0229
0.75	6.0294	12.9662	**4.2767**	0.0245	0.0575	0.0314
1.00	10.2012	18.4711	**6.9408**	0.0304	0.0667	0.0373

Similar to Tables 8.7 and 8.8, Tables 8.11 and 8.12 provide a comparison of OLS, LLR, and $MRR2_{\lambda 2}$ in terms of ASEL and AD in the simulations based on the SFD for Goals 1 and 2, respectively. Like Table 8.7, Table 8.11 shows that OLS and $MRR2_{\lambda 2}$ are quite competitive to each other since they have nearly equal values of ASEL and AD for Goal 1. But unlike Table 8.7, Table 8.11 shows that LLR is much worse than OLS and $MRR2_{\lambda 2}$ since it has much larger ASEL values and much smaller AD values than the other two methods. Table 8.12 has similar pattern to Table 8.8 for Goal 2 in terms of ASEL and AD as follows. When $\gamma = 0.00$, LLR performs slightly better than the other two since it has the smallest ASEL value and the largest AD value, but actually, OLS and $MRR2_{\lambda 2}$ are quite competitive. When $\gamma > 0.00$, $MRR2_{\lambda 2}$ always performs much better than the other two.

Table 8.11: ASEL and AD values by OLS, LLR, and $MRR2_{\lambda 2}$ for Goal 1 in the simulations based on SFD, with the ranges of the estimated Monte Carlo errors of ASEL and AD values (0.0018, 0.0787) and (6.9×10^{-5}, 4.1×10^{-4}), respectively. Best values in bold.

	ASEL			AD		
γ	OLS	LLR	$MRR2_{\lambda 2}$	OLS	LLR	$MRR2_{\lambda 2}$
0.00	**10.2587**	10.9483	10.2886	**0.5954**	0.5668	0.5941
0.25	**9.5508**	10.6943	9.5592	**0.6181**	0.5721	0.6178
0.50	8.9430	10.7227	**8.9184**	0.6385	0.5678	**0.6397**
0.75	8.4901	11.0299	**8.4198**	0.6562	0.5537	**0.6591**
1.00	8.2363	11.7881	**8.0901**	0.6711	0.5221	**0.6762**

Table 8.12: ASEL and AD values by OLS, LLR, and MRR2$_{\lambda 2}$ in Goal 2 in the simulations based on SFD, with the ranges of the estimated Monte Carlo errors of ASEL and AD values (0.0167, 0.0898) and (0.0022, 0.0145), respectively. Best values in bold.

	ASEL			AD		
γ	OLS	LLR	MRR2$_{\lambda 2}$	OLS	LLR	MRR2$_{\lambda 2}$
0.00	1.5597	**1.5285**	1.5792	0.7977	**0.7988**	0.7959
0.25	2.0745	2.0121	**1.9137**	0.7203	0.7288	**0.7419**
0.50	2.7866	2.3681	**2.0561**	0.5974	0.6714	**0.7114**
0.75	3.8317	2.8032	**2.3270**	0.3869	0.5853	**0.6757**
1.00	4.9048	3.4404	**2.9258**	0.2729	0.4809	**0.6040**

8.3 Conclusion

RSM has utilized parametric regression techniques to study products and processes since its inception. One drawback, however, is that optimization depends too heavily on the assumption of well-estimated models for the responses of interest, and it is often the case that the user's specified parametric models are not flexible enough to adequately model the process. Nonparametric smoothing has been considered when the user is unable to specify the explicit form for the underlying function. However, in small sample settings, which are customary for response surface experiments, the nonparametric approach often produces estimates that are highly variable. Therefore, we suggest MRR2, a semiparametric approach, which combines the OLS, a parametric method, and the LLR, a nonparametric method. This combination combines the advantages from both the parametric and nonparametric methods and reduce some of their disadvantages (high bias mainly from the OLS, and large variance mainly from the LLR).

In the minced fish quality example of the MRO problem, the results show that MRR2 is superior to OLS and LLR in terms of the seven model comparison criteria through all of the responses. During the optimization stage, the models by the OLS, LLR, and MRR2 are assumed to be correct respectively, and the desirability function method has been utilized to find the optimal solutions with the best compromise among the responses. Although the optimal solutions by the three modeling methods are incomparable directly, the optimization results by MRR2 is more reliable, because the MRR2 model appears to have less misspec-

ification (i.e., both lower bias and variances) than both the parametric and nonparametric models based on the model comparison results.

Simulation studies based on a CCD with different degrees of model misspecification were conducted to compare the three approaches more generally. If the user correctly specifies the model for each response of interest, the parametric approach yields the best fit in terms of SIMSE and its corresponding optimization results are the best for Goal 1 and highly competitive to the results by the other methods for Goal 2 in terms of ASEL and AD. If there exists some moderate model misspecification, the semiparametric approach always yields the best fit in terms of SIMSE and its corresponding optimization results are superior or highly competitive to the ones by the other two methods for Goal 1, and are always the best for Goal 2. Thus, we can conclude that the semiparametric approach consistently performs well in general and its corresponding optimization results are more reliable than the other two.

Although the semiparametric approach performs the best in terms of SIMSE when there exists some model misspecification, its advantage is not great over the parametric approach because a CCD is designed to work well for a second-order polynomial model when using OLS, our parametric method. The CCD space is too sparse to capture well the surface structure of each response. A space-filling design (SFD) is modified from the CCD in the study to capture more of this structure.

Simulation studies based on the SFD with different degrees of model misspecification were conducted to compare the three approaches under the same conditions as the simulation studies based on the CCD. The model fitting and optimization results are quite similar to the results in the simulations based on the CCD. The only big difference is that the semiparametric approach performs much better than the other two in terms of SIMSE.

Since, in practice, one never knows if the form of the underlying model for each response of interest has been correctly specified, we advocate the semiparametric method as it is the only one which consistently performs well over all degrees of potential misspecification and its corresponding optimization results are more reliable. We also suggest use of a SFD in those cases where the user is unsure of the appropriateness of a parametric model. However, the optimal SFD for a specific problem is unknown and is certainly a viable topic for future research.

Chapter 9

Summary and Future Research

The multi-response optimization (MRO) problem in response surface methodology (RSM) is quite common in industry and in many other areas of science. Before the optimization stage in MRO, appropriate fitted models for each response are required. Traditional RSM modeling parametric methods, which are not flexible, are likely to lead to biased estimates and result in miscalculating optimal solutions, when the user's model is incorrectly specified. Nonparametric methods have been suggested as an alternative, yet they often result in highly variable estimates, especially for sparse data with small sample size which are the typical properties of traditional RSM experiments. Therefore, in this research, we have proposed the use of semi-parametric methods to combine the advantages from each of the parametric and nonparametric methods and at the same time to avoid the disadvantages inherent in each.

During the optimization stage in MRO, the desirability function method, one of the most flexible and popular MRO approaches and which has been utilized in this research, is a highly nonlinear function. Therefore, we have proposed the use of the genetic algorithm (GA), which is a global optimization tool, to help solve the MRO problem.

This chapter summarizes the work from the previous chapters and then proposes areas for future research.

9.1 Summary and Future Work on a MGA

A GA is a very powerful optimization tool, but it has computational efficiency problem. Thus, we developed an improved GA, the MGA, with four different versions, as presented in Chapter 5. The main idea in our modification is to incorporate a local directional search into the GA process. The local directional searches utilized in this study to develop our four MGAs include using SD, NR, DFDS, and the method that combines SD with DFDS. MGA_{SD} and MGA_4 both require the first derivative of f, MGA_{NR} requires calculating the Hessian matrix with the second derivative of f and its inverse matrix, while MGA_3 requires no derivative calculations.

Several examples, including a case study of a chemical process, are used to facilitate comparisons of GA, MGA_{SD}, MGA_3, MGA_4, and MGA_{NR} under a variety of combinations using different levels of GA operations. Numerical and graphic comparison results in all of the examples show that the new MGAs procedures perform better than the traditional GA procedure, not only in computational efficiency (by stopping rule 2), but also in accuracy (by stopping rule 1), in most cases.

Several issues remain for further study. For example, the three derivative-free directions defined in MGA_3 may not be optimal. Additionally, the derivative-based directions defined in MGA_{SD} and MGA_{NR} may also not be optimal. Perhaps, there are other directions better than the four we have chosen in this study. Another issue concerns the appropriate moving distance, once the directions are chosen. The size of an appropriate moving distance, arbitrarily chosen by us, may greatly affect the efficiency of the MGAs. The last issue is on the optimal setting of the GA operations. In this study, type of replacement, the number of crossover points, the mutation rate, the three main GA operations, have been studied. However, there may be some other operations affecting the GA performance, such as population size and parent/offspring ratio. We plan to study these issues in future work.

9.2 Summary and Future Work on Finding the Feasible Region of a Desirability Function

Finding all possible feasible regions is usually preferred by practitioners, instead of finding only one or several optimal/feasible solutions. The reason is that some feasible regions may be more desirable than others based on practical considerations. In Chapter 6, using the stochastic property of a GA/MGA, we have presented a procedure of using a MGA to obtain all possible feasible regions of a desirability function. This procedure is not limited by the number of factors. A case study has been employed to illustrate that our procedure can successfully define all feasible regions.

This procedure should be easily extended to other nonlinear objective functions such as generalized distance measure function and weighted squared error loss function mentioned in Chapter 3. We plan to study these issues in future work.

9.3 Summary and Future Work on a Semiparametric Approach to MRO

MRR2, a semiparametric modeling method, combines the OLS, a parametric method, and the LLR, a nonparametric method. This semiparametric approach combines the advantages from both the parametric and nonparametric methods and reduces some of their disadvantages (high bias mainly from the OLS, and large variance mainly from the LLR).

In the minced fish quality example on the MRO problem in Chapter 8, the results show that MRR2 is superior to OLS and LLR in terms of the seven model comparison criteria through all of the responses. During the optimization stage, the models by OLS, LLR, and MRR2 are assumed to be correct respectively, and the desirability function method has been utilized to find the optimal solutions with the best compromise among the responses. Although the optimal solutions by the three modeling methods are incomparable directly, the optimization results by the MRR2 is more reliable, because the semiparametric models appears to have less model misspecification than both the parametric and nonparametric models based on the model comparison results.

Simulation studies based on a CCD with different degrees of model misspecification were conducted to compare the three approaches more generally. If the user correctly specifies the model for each response of interest, the parametric approach yields the best fit in terms of SIMSE and its corresponding optimization results are the best for Goal 1 and highly competitive to the results by the other methods for Goal 2 in terms of ASEL and AD. If there exists some moderate model misspecification, the semiparametric approach always yields the best fit in terms of SIMSE and its corresponding optimization results are superior or highly competitive to the ones by the other two methods for Goal 1, and are always the best for Goal 2. Thus, we can conclude that the semiparametric approach consistently performs well in general and its corresponding optimization results are more reliable than the other two.

Although the semiparametric approach performs the best in terms of SIMSE when there exists some model misspecification, its advantage is not great over the parametric approach because a CCD is designed to work well for a second-order polynomial model when using OLS, our parametric approach. The CCD space is too sparse to capture well the surface structure of each response. A space-filling design (SFD) is modified from the CCD in the study to capture more of this structure.

Simulation studies based on the SFD with different degrees of model misspecification were conducted to compare the three approaches under the same conditions as the simulation studies based on the CCD. The model fitting and optimization results are quite similar to the results in the simulations based on the CCD. The only big difference is that the semiparametric approach performs much better than the other two in terms of SIMSE.

Since, in practice, one never knows if the form of the underlying model for each response of interest has been correctly specified, we advocate the semiparametric method as it is the only one which consistently performs well over all degrees of potential misspecification and its corresponding optimization results are more reliable.

Several issues remain for further study. In this research, PRESS** is utilized to select an appropriate bandwidth for LLR fit, including the LLR method to obtain a fit to the OLS residual in MRR2. As mentioned in Chapter 2, Mays and Birch (1998, 2002) compared PRESS** with PRESS* and some other popular bandwidth selectors such as the generalized cross-validation (GCV) and Akaike's Information criterion (AIC). Their examples and

simulation results show that, when using MRR2, PRESS** is the best choice in terms of minimizing integrated mean squared error of fit across a broad variety of data scenarios. But their examples and simulation studies do not include data resulting from a typical RSM problem. That is, designs where the design points are sparse with most of the design points on the edge of the design space and where the sample size is small. Therefore, we propose simulating RSM data and comparing PRESS**, PRESS* with other popular bandwidth selection methods such as GCV, and AIC (two of the most popular bandwidth selectors), and determining if PRESS** still outperforms the others. This issue will be left for our future research.

Another issue is related to outliers in the RSM data. It is well-known that outliers frequently occur in the RSM data. The MRR2 techniques originally designed are robust to model misspecification, but not robust to outliers. Especially the MRR2 technique we have utilized in this research is not robust to outliers, because it combines OLS (parametric part) with LLR (nonparametric part), and both OLS and LLR are sensitive to outliers.

One way to deal with the outlier problem is to modify the MRR2 technique to combine a robust parametric method, such as m-estimators with LOWESS, one of resistant smoothing nonparametric techniques. LOWESS denotes LOcally WEighted Scatter plot Smoothing, introduced by Cleveland (1979). The idea of LOWESS is to start with a local polynomial least squares fit and then to "robustify" it. More details on LOWESS can be seen, for example, Hardle (1990). Assaid (1997) studied this problem using MRR2 combining m-estimators and a robust version of LLR (a method related to LOWESS) for the non-RSM problem and found excellent results favoring MRR2 over either m-estimation or robust LLR. This issue on the outlier problem will be left for our future research.

9.4 Other Future Work

In this dissertation, the only MRO technique focused on is the desirability function. As discussed in Chapter 3, the generalized distance method and the weighted squared error loss method are popular due to both taking into account the variance-covariance structure of the responses. However, the desirability function does not consider the variance-covariance structure of the responses. Thus, comparing the desirability function with these two methods

via Monte Carlo simulation would be interesting. This will be left in future work.

We adopted the MRR2 modeling technique to the MRO problem. For simplification of the application of MRR2, we assume that the variance of each response is constant across all responses. When the constant variance assumption is violated, we prefer to apply the dual response approach to the MRO problem with the MRR2 technique for both the mean and variance models. This problem will be left for our future work.

As mentioned in Chapter 1 of this research, we assume that the data have already been collected and we focused on the latter stages modeling and optimization. That is, the traditional RSM designs such as CCD are assumed to be conducted in a some region of interest so that the lower-polynomial parametric method is suitable. However, suppose it is known that a simple polynomial model in the region of interest cannot adequately describe the response before an experiment is conducted. That is, the traditional RSM designs are unsuitable.

Myers et al. (2004) suggest conducting a space-filling design covering the entire experimental region, not just a small region of interest, so that the nonparametric or semiparametric modeling methods could be better utilized than if a traditional design where used. Of course, MRR2 should work well even if the parametric function is nonlinear. Actually, the nonparametric part of MRR2 should be more efficient if the data were collected through the space-filling design rather than through a sparse data design such as a CCD. In this situation, comparing MRR2 with one of the nonparametric techniques would be interesting. This will also be left for future work.

Appendix A

Computational Details on a Directional Search in a MGA and Some Related Functions

A.1 Mathematical Representation of the Three Directions in MGA_3

We first introduce our notation. Parent 1 (P1) is given by $\mathbf{x}_{P1} = [x_{P11}, ..., x_{P1k}]'$, where $\mathbf{x}$ is a vector of size $k \times 1$ where k is the number of factors or the number of dimensions. Similarly, Parent 2 (P2) is given by $\mathbf{x}_{P2} = [x_{P21}, ..., x_{P2k}]'$, and their offspring (O) is expressed as $\mathbf{x}_O = [x_{O1}, ..., x_{Ok}]'$. The Parent 1 direction (from P1 to O) is expressed as δ_{P1O} and the Parent 2 direction (from P2 to O) is as δ_{P2O}. And the common direction is simply denoted as δ. The new points after the first step along the three directions are expressed as $\mathbf{x}_{New1} = [x_{New11}, ..., x_{New1k}]'$, $\mathbf{x}_{New2} = [x_{New21}, ..., x_{New2k}]'$, and $\mathbf{x}_{New} = [x_{New1}, ..., x_{Newk}]'$, corresponding to Parent 1, Parent 2, and their common direction respectively. The appropriate moving distance on each axis in each moving step is expressed as d.

The parent 1 direction, which essentially is the different distances on each dimension between points P1 and O, is expressed as

$$\delta_{P1O} = \mathbf{x}_O - \mathbf{x}_{P1} = [\delta_{11}, \delta_{12}, ..., \delta_{1k}]'. \tag{A.1}$$

Similarly, the parent 2 direction is expressed as

$$\delta_{P2O} = \mathbf{x}_O - \mathbf{x}_{P2} = [\delta_{21}, \delta_{22}, ..., \delta_{2k}]'. \tag{A.2}$$

To keep the same directions and move along the three paths, the moving distance on each axis should be in constant proportion to each other, as in the method of steepest ascent/descent in response surface methodology (RSM). In RSM, the constant proportion on the i^{th} dimension is defined as $\hat{\beta}_i/\hat{\beta}^*$, where the $\hat{\beta}_i$ is the ith estimated coefficient in the estimated first-order model and the $\hat{\beta}^*$ is the largest coefficient in magnitude among the k estimated coefficients, that is, $\hat{\beta}^* = \max_{i=1,...,k} |\hat{\beta}_i|$. From this ratio, we can see that the proportion only depends on the β_i, the ith coefficient. The moving distance on the ith dimension is defined as $(\hat{\beta}_i/\hat{\beta}^*) * \rho$, where the ρ is an appropriate fixed distance. (For more details, please see Myers and Montgomery (2002) in page 205-207.)

In our GA application, the main idea in moving along the parent 1 path is the same as that in the method of steepest ascent/descent. That is, to keep the constant proportion in each dimension and move some appropriate fixed distance (which is d in our case) along the parent 1 path. But the difference between our GA case and RSM is the starting point. In the GA case, the starting point is P1, not O. That is, the first step has already been completed. So the next moving step starts at O. The largest moving distance in the first step is also not d, but $\max_{i=1,...,k} |\delta_{1i}|$, where the δ_{1i} is the moving distance on ith axis in Equation (A.1). Let δ_1^* denote $\max_{i=1,...,k} |\delta_{1i}|$. In our study, if $\delta_1^* < d$, then the moving distant in the next step will be δ_1^*. Otherwise, the distance in the next step will be d. The distance d is obviously utilized to control the next moving distance.

The procedure of moving along the parent 1 direction is as following.

1. Calculate δ_{P1O} and then find $\delta_1^* = \max_{i=1,...,k} |\delta_{1i}|$, the largest distance in the first moving step.

2. If $\delta_1^* < d$, then the next new position on the ith axis, $i = 1, ..., k$, is defined as $x_{New1i} = x_{Oi} + (\delta_{1i}/\delta_1^*) \times d$. Otherwise, the new position is $x_{New1i} = x_{Oi} + \delta_{1i}$.

3. Check the region of the new point $\mathbf{x}_{New1} = [x_{New11}, ..., x_{New1k}]'$. If x_{New1i} is greater than its upper bound (which is the largest value in the ith domain), then let it be the upper bound . Similarly, if it is less than its lower bound (which is the lowest value in the ith domain), then let it be the lower bound. (Usually, the upper bounds and lower bounds have been given through defining the objective function.)

4. Evaluate the new point $\mathbf{x}_{New1}$ by the objective function. If the new point performs worse than the point $\mathbf{x}_O$, then the process of moving along the parent 1 direction is halted. If the new point performs better than the $\mathbf{x}_O$, then replace the point $\mathbf{x}_{New1}$ by the next new point $\mathbf{x}_{New1} + \mathbf{\Delta}_{N1O}$, where $\mathbf{\Delta}_{N1O} = \mathbf{x}_{New1} - \mathbf{x}_O$. (The "N1O" means "New point from Parent 1" to "Offspring".) Then return to Step 3.

The procedure for moving along the parent 2 direction is the same as that for the parent 1 direction. However, the procedure for the common direction is slightly different from them, due to the different starting points. The starting points from the parents directions are P1 or P2, while the starting point in the common direction is O.

As mentioned earlier, building the common direction depends on whether both parent directions are consistent or not. If they are consistent on ith axis (either both positive or both negative), then move the same direction on the ith axis as the parent directions. Otherwise, stay on that axis without any movement, due to inconsistent directions. There is a special case: one of the moving distances on an axis in the parent directions is zero and the other is nonzero. In this case, we recommend movement in the same direction with the parent direction with nonzero moving distance on the axis.

The procedure for movement along the common direction is as follows:

1. Calculate δ_{P1O} and δ_{P2O} as Equation (A.1) and (A.2).
2. The next new point is defined as $\mathbf{x}_{New} = [x_{New1}, ..., x_{Newk}]'$ along the path from the common direction. To establish the common direction, three situations on each axis/dimension are possible: (a) the $\delta_{1i} \times \delta_{2i} > 0$ which means that there is a common direction on the ith axis; (b) The $\delta_{1i} \times \delta_{2i} < 0$ which means that there is not a common direction on the ith axis; and (c) the $\delta_{1i} \times \delta_{2i} = 0$ which means that at least one of δ_{1i} and δ_{2i} equals zero.
    - 2.1. If the situation is (a), then the new point position on the ith axis is given by $x_{Newi} = x_{Oi} + min(|\delta_{1i}|, |\delta_{2i}|, d)$ if both δ_{1i} and δ_{2i} are positive, or $x_{Newi} = x_{Oi} - min(|\delta_{1i}|, |\delta_{2i}|, d)$ if both δ_{1i} and δ_{2i} are negative.
    - 2.2. If the situation is (b), the new point position on the ith axis is given by $x_{Newi} = x_{Oi}$ (no movement on the ith axis in this situation).
    - 2.3. If the situation is (c), there are three subcases: (1) $\delta_{1i} = 0$ and $\delta_{2i} \neq 0$; (2) $\delta_{1i} \neq 0$ and $\delta_{2i} = 0$; and (3) $\delta_{1i} = 0$ and $\delta_{2i} = 0$.
        - 2.3.1. For case (1), if $|\delta_{2i}| \geq d$, then $x_{Newi} = x_{Oi} + d$ (when $\delta_{2i} > 0$) or $x_{Newi} = x_{Oi} - d$ (when $\delta_{2i} < 0$). Otherwise, $x_{Newi} = x_{Oi} + \delta_{2i}$.

2.3.2. For case (2), similar to case (1), if $|\delta_{1i}| \geq d$, then $x_{Newi} = x_{Oi} \pm d$. Otherwise $x_{Newi} = x_{Oi} + \delta_{1i}$.

2.3.3. For case (3), $x_{Newi} = x_{Oi}$.

3. Check the range of the new point $\mathbf{x}_{New}$.

4. Evaluate the point $\mathbf{x}_{New}$. If the new point performs worse than the point $\mathbf{x}_O$, then the process for moving along the common direction is stopped. If the new point is better than $\mathbf{x}_O$, then replace the point $\mathbf{x}_{New}$ by the next new point $\mathbf{x}_{New} + \mathbf{\Delta}_{NCO}$, where $\mathbf{\Delta}_{NCO} = \mathbf{x}_{New} - \mathbf{x}_O$. (The "NCO" means "New from Common directions" and "Offspring"). Return to Step 3.

A.2 Computational Details on A Derivative-based Directional Search by SD

In this appendix, we focus on how to implement SD into the GA process. Suppose that in the i^{th} iteration, the best offspring, which is the best among both the current parent and offspring populations, denoted by $\mathbf{x}_O = [x_{O1}, ..., x_{Ok}]'$, is found. Then the MGA procedure will implement a direction determined by SD into the GA process.

Based on formula (5.1), the procedure of building a derivative-based directional search by SD into the GA process between the i^{th} and $(i+1)^{th}$ steps is as follows:

1. The first new point is defined as $\mathbf{x}_1 = \mathbf{x}_O - d\nabla f(\mathbf{x}_O)$, where d is the size of a moving distance in each step and $\mathbf{x}_O$ is the best offspring. If $f(\mathbf{x}_1) < f(\mathbf{x}_O)$ in the case of finding a minimum of f, or $f(\mathbf{x}_1) > f(\mathbf{x}_O)$ in the case of finding a maximum, then go to Step 2. Otherwise, the procedure is halted.

2. Compute $\mathbf{x}_{j+1} = \mathbf{x}_j - d\nabla f(\mathbf{x}_j)$, where the iteration index $j = 1, 2, ...$ If $f(\mathbf{x}_{j+1}) < f(\mathbf{x}_j)$ for minimization, then repeat Step 2 by letting $j = j + 1$. Otherwise, the procedure is halted.

The procedure shows us that the algorithm starts at the best offspring, $\mathbf{x}_O$, on the surface of the objective function f and minimizes along the direction of its gradient. This procedure can be improved by using fractional increments (Myers, 1990, pp. 429) to allow the procedure itself to adjust the moving distance in magnitude to the surface of the objective function. In

our study, the strategy on fractional increments implemented into each step of the procedure of building a direction by SD is as follows:

1. Let $l = 1$.
2. Let $\mathbf{x}_{j+1} = \mathbf{x}_j - d\gamma^{l-1}\nabla f(\mathbf{x}_j)$, where γ is a constant value within (0, 1), and $\gamma = 0.5$ in our study.
3. If $f(\mathbf{x}_{j+1}) < f(\mathbf{x}_j)$ for minimization, then the procedure is completed.
4. If $f(\mathbf{x}_{j+1}) > f(\mathbf{x}_j)$, then let $l = l + 1$ and go back to Step 2.
5. If $l > a$ (where a is some constant integer and $a = 5$ in our study), then the procedure is halted.

A.3 Computational Details on A Derivative-based Directional Search by NR

Implementation by the NR method into a GA process is quite similar to implementation of a search by SD. When the best offspring, which is also the best over the parent population, is found at the i^{th} iteration, implement a directional search by NR into the GA process between the i^{th} and $(i+1)^{th}$ steps as the following procedure, based on formula (5.3):

1. The first new point is defined as $\mathbf{x}_1 = \mathbf{x}_O - \mathbf{H}_O^{-1}\nabla f(\mathbf{x}_O)$, where similar to the Hessian matrix in Equation 5.3, $\mathbf{H}_O$ is the Hessian matrix evaluated at location $\mathbf{x}_O$. If the point $\mathbf{x}_1$ is better than $\mathbf{x}_O$ in terms of f, then go to Step 2. Otherwise, the procedure is halted.
2. Compute $\mathbf{x}_{j+1} = \mathbf{x}_j - \mathbf{H}_j^{-1}\nabla f(\mathbf{x}_j)$, where the iteration index $j = 1, 2, ...$ If $\mathbf{x}_{j+1}$ is better than $\mathbf{x}_j$ in terms of f, then repeat Step 2 by letting $j = j + 1$. Otherwise, the procedure is halted.

This procedure can also be improved by using fractional increments as the procedure by SD. The following is the strategy on fractional increments implemented into each step of the procedure of NR.

1. Let $l = 1$.

2. Let $\mathbf{x}_{j+1} = \mathbf{x}_j - \gamma^{l-1}\mathbf{H}_j^{-1}\nabla f(\mathbf{x}_j)$, where γ is a constant value within (0, 1), and $\gamma = 0.5$ in our study.

3. If $\mathbf{x}_{j+1}$ is better than $\mathbf{x}_j$ in terms of f, then the procedure is completed.

4. If $\mathbf{x}_{j+1}$ is worse than $\mathbf{x}_j$ in terms of f, then let $l = l + 1$ and go back to Step 2.

5. If $l > a$ (where a is some constant integer and $a = 5$ in our study), then the procedure is halted.

A.4 Sphere Model and Schwefel's Function

The sphere model (Schwefel, 1995; Back, 1996; and Haupt and Haupt, 2004) is given by

$$f(\mathbf{x}) = \sum_{i=1}^{k} x_i^2,$$

where the k is the number of dimensions of the function. We chose $k = 2$ in this study and the range is set to $-40 \leq x_i \leq 60$ as in Back (1996). The goal is to find its minimal value and its corresponding location. Obviously, the minimum value is 0 and its location is $(0, 0)$.

A generalized Schwefel's problem 2.26 from Schwefel (1995), is given by

$$\sum_{i=1}^{k} -x_i \sin(\sqrt{|x_i|}), where - 500 \leq x_i \leq 500,$$

where k is the number of dimensions of the function. The minimum of the objective function is given by

$$\min(f(\mathbf{x})) = f(420.9687, ..., 420.9687).$$

The minimum is dependent on k, the number of dimensions. For example, if k =5, then the minimum value is -2094.9144. If k=20, then the minimum value is -8379.6577. Figure A.1 shows its 1- and 2-dimensional surfaces.

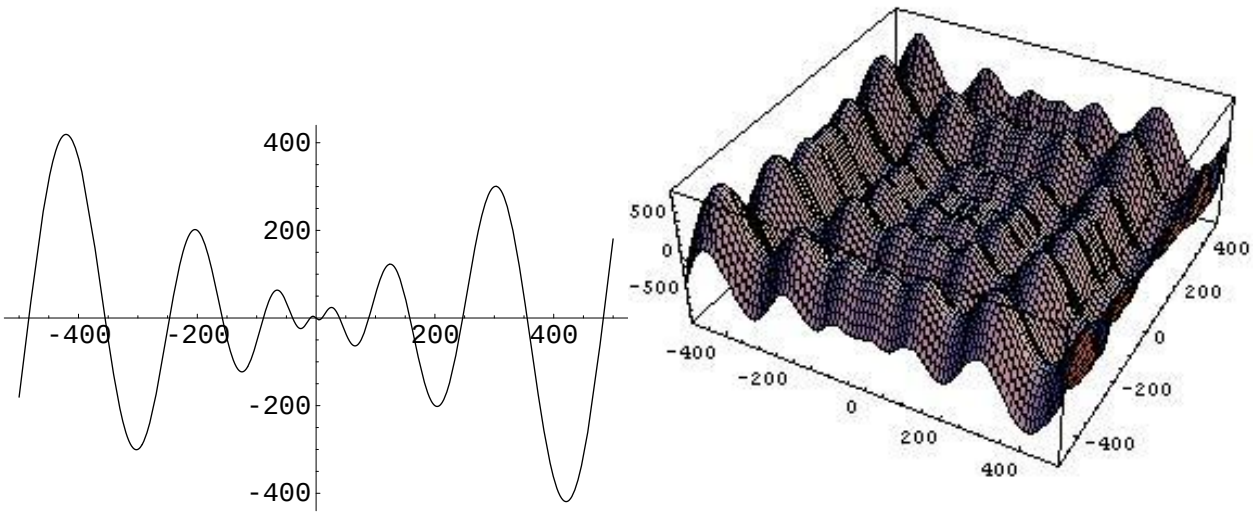

Figure A.1: Surface of Schwefel's function. Left: 1-dimension; right: 2-dimension.

Appendix B

Some Relationships Among the OLS, LLR, and MRR2 Fits

In this study, our MRR2 procedure, a semiparametric method, combines the parametric fit to the raw data obtained by the OLS method with a nonparametric fit to the residuals from the parametric fit obtained by the LLR method via a mixing parameter λ. Under some special conditions, several relationships exist among the OLS, LLR, and MRR2 fits, as stated the two following theorems.

Theorem 1 *If the bandwidth b goes to infinity, then the MRR2 fit at location* $\mathbf{x}_0$ *is equal to the OLS fit at* $\mathbf{x}_0$ *for all values of* λ.

Proof. According to Equations 2.5-2.7, the OLS fit at location $\mathbf{x'}_0 = (x_{10}, x_{20}, ..., x_{k0})$ for the full second-order model is given by

$$\hat{y}_0^{(OLS)} = \mathbf{h}_0^{(OLS)'}\mathbf{y} = \breve{\mathbf{x}}'_0(\mathbf{X'X})^{-1}\mathbf{X'y}, \tag{B.1}$$

where $\mathbf{h}_0^{(OLS)'} = \breve{\mathbf{x}}'_0(\mathbf{X'X})^{-1}\mathbf{X'}$, $\breve{\mathbf{x}}'_0 = (1\ x_{10}\ x_{20}\ ...\ x_{q0})$, including the k first-order terms, the k second-order terms, and the $\begin{pmatrix} k \\ 2 \end{pmatrix}$ interaction terms, $q = 2k + \begin{pmatrix} k \\ 2 \end{pmatrix}$, and $\mathbf{X}$ is the model matrix, the same as in Equation 2.3. However, according to Equations 2.20-2.21, the MRR2 fit at $\mathbf{x}_0$ can be expressed as

$$\hat{y}_0^{(MRR2)} = \mathbf{h}_0^{(MRR2)'}\mathbf{y} = \mathbf{h}_0^{(OLS)'}\mathbf{y} + \lambda\mathbf{h}_{\mathbf{r}0}^{(LLR)'}\mathbf{r} \tag{B.2}$$

$$= [\breve{\mathbf{x}}'_0(\mathbf{X'X})^{-1}\mathbf{X'y} + \lambda\tilde{\mathbf{x}}'_0(\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}0}(\mathbf{I} - \mathbf{X}(\mathbf{X'X})^{-1}\mathbf{X'})]\mathbf{y}, \tag{B.3}$$

where $\mathbf{h}_{\mathbf{r}0}^{(LLR)'} = \tilde{\mathbf{x}}_0'(\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}0}$, $\tilde{\mathbf{x}}_0'$ and $\tilde{\mathbf{X}}$ both include the intercept and the k first-order terms, as shown in Equation 2.16, the local weight matrix $\mathbf{W}_{\mathbf{r}0} = \left\langle h_{\mathbf{r}0j}^{(KER)} \right\rangle$ is the similar to the local weight matrix in Equation 2.16 by considering the residuals from the parametric fits as the response, and similarly, the kernel weight $h_{\mathbf{r}0j}^{(KER)}$ is similar to the kernel weight in Equation 2.16.

When the bandwidth b goes to infinity, the local weight matrix becomes a constant weight matrix across all observations. That is, $\mathbf{W}_{\mathbf{r}0} = \mathbf{I}$. Thus,

$$\tilde{\mathbf{x}}_0'(\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}0}(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') = \tilde{\mathbf{x}}_0'(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'). \tag{B.4}$$

$\mathbf{X}$ can be partitioned into $[\ \tilde{\mathbf{X}}\ \ \tilde{\mathbf{X}}_2\ ]$, where $\tilde{\mathbf{X}}_2$ includes the k second-order terms and the $\begin{pmatrix} k \\ 2 \end{pmatrix}$ interaction terms. That is, $\tilde{\mathbf{X}}$ is a subset of $\mathbf{X}$. According to $\tilde{\mathbf{X}}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' = \tilde{\mathbf{X}}'$ (Rencher, 2000),

$$\tilde{\mathbf{x}}_0'(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') = \tilde{\mathbf{x}}_0'(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}(\tilde{\mathbf{X}}' - \tilde{\mathbf{X}}') = 0. \tag{B.5}$$

Therefore, $\hat{y}_0^{(MRR2)} = \hat{y}_0^{(OLS)}$ when the bandwidth goes to infinity. □

Theorem 2 *If the parametric fit of MRR2 is the first order polynomial fit to the raw data and $\lambda = 1$, then the MRR2 fit at location $\mathbf{x}_0$ is equal to the LLR fit at $\mathbf{x}_0$.*

Proof. The LLR fit at $\mathbf{x}'_0 = (x_{10}, x_{20}, ..., x_{k0})$ is given by

$$\hat{y}_0^{(LLR)} = \mathbf{h}_0^{(LLR)'}\mathbf{y} = \tilde{\mathbf{x}}_0'(\tilde{\mathbf{X}}'\mathbf{W}_0\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_0\mathbf{y}, \tag{B.6}$$

the same as Equation 2.16. However, the MRR2 fit at $\mathbf{x}_0$ can be expressed as in (B.2) and (B.3).

When $\lambda = 1$ and the parametric part of the MRR2 fit is the first order polynomial, $\breve{\mathbf{x}}_0$ becomes $\tilde{\mathbf{x}}_0$ and $\mathbf{X}$ becomes $\tilde{\mathbf{X}}$ and thus,

$$\begin{aligned} \mathbf{h}_0^{(MRR2)'} &= \tilde{\mathbf{x}}_0'(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}' + \tilde{\mathbf{x}}_0'(\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}0}(\mathbf{I} - \tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}') \\ &= \tilde{\mathbf{x}}_0'(\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}0}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}0}. \end{aligned} \tag{B.7}$$

To show $\hat{y}_0^{(LLR)} = \hat{y}_0^{(MRR2)}$ it must be shown that $\mathbf{W}_0 = \mathbf{W}_{\mathbf{r}0}$. This is equivalent to proving that PRESS** by LLR and PRESS** by MRR2 are equal to each other because the local

weight matrices only depend on the value of the bandwidth chosen by PRESS** in this study.

The PRESS** by LLR is given by

$$PRESS^{**(LLR)}(b) = \frac{\sum (y_i - \hat{y}_{i,-i}^{(LLR)}(b))^2}{n - trace(\mathbf{H}^{(LLR)}(b)) + (n-k-1)\frac{SSE_{\max} - SSE_b}{SSE_{\max}}}, \tag{B.8}$$

while the PRESS** by MRR2 is

$$PRESS^{**(MRR2)}(b) = \frac{\sum (e_i - \hat{e}_{i,-i}^{(MRR2)}(b))^2}{n - trace(\mathbf{H}_{\mathbf{r}}^{(LLR)}(b)) + (n-k-1)\frac{SSE_{\mathbf{r}\max} - SSE_{\mathbf{r}b}}{SSE_{\mathbf{r}\max}}}, \tag{B.9}$$

where e_i is the ith residual from the parametric fit, considered as a response, and $\hat{e}_{i,-i}^{(LLR)}$ is the LLR fit at location $\mathbf{x_i}$ with the i^{th} observation left out. The PRESS** by LLR in (B.8) is exactly the same as the PRESS** in (2.24), while the PRESS** by MRR2 in (B.9) is similar to (B.8) and (2.24), except for considering the residuals $\mathbf{r}$ from the parametric fit as a response. Therefore, the terms $\mathbf{H}^{(LLR)}(b)$, $SSE_{\max}$, and SSE_b in (B.9) are marked with a subscript $\mathbf{r}$.

According to the formula $\hat{y}_{i,-i} = \frac{\hat{y}_i - h_{ii}y_i}{1-h_{ii}}$ (Myers, 1990), the numerator of the PRESS** by LLR becomes

$$\sum (y_i - \hat{y}_{i,-i}^{(LLR)})^2 = \sum \left(\frac{y_i - \hat{y}_i^{(LLR)}}{1 - h_{ii}^{(LLR)}} \right)^2, \tag{B.10}$$

while the numerator of the PRESS** by MRR2 becomes

$$\sum (e_i - \hat{e}_{i,-i}^{(LLR)})^2 = \sum \left(\frac{e_i - \hat{e}_i^{(LLR)}}{1 - h_{ii}^{(LLR)}} \right)^2. \tag{B.11}$$

Since $e_i = y_i - \hat{y}_i^{(OLS)}$ and

$$\begin{aligned}
\hat{e}_i^{(LLR)} &= \tilde{\mathbf{x}}_i'(\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}i}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}i}\mathbf{r} \\
&= \tilde{\mathbf{x}}_i'(\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}i}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}i}(\mathbf{y} - \hat{\mathbf{y}}^{(OLS)}) \\
&= \tilde{\mathbf{x}}_i'(\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}i}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}i}\mathbf{y} - \tilde{\mathbf{x}}_i'(\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}i}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}i}\tilde{\mathbf{X}}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{y} \\
&= \hat{y}_i^{(LLR)} - \hat{y}_i^{(OLS)},
\end{aligned} \tag{B.12}$$

the numerator of PRESS** by MRR2 (shown in (B.9) and (B.11) becomes

$$\sum (e_i - \hat{e}_{i,-i}^{(LLR)})^2 = \sum \left(\frac{y_i - \hat{y}_i^{(LLR)}}{1 - h_{ii}^{(LLR)}} \right)^2, \tag{B.13}$$

which is equal to the numerator of PRESS** by LLR (shown in (B.8) and (B.10).

Now, it must be shown that the denominators in (B.8) and (B.9) are equal. The $\mathbf{H}^{(LLR)}(b)$ in (B.8) is given by

$$\mathbf{H}^{(LLR)}(b) = \begin{bmatrix} \tilde{\mathbf{x}}'(\tilde{\mathbf{X}}'\mathbf{W}_1\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_1 \\ \vdots \\ \tilde{\mathbf{x}}'(\tilde{\mathbf{X}}'\mathbf{W}_n\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_n \end{bmatrix}, \tag{B.14}$$

while the $\mathbf{H}^{(MRR2)}(b)$ in (B.9) is by

$$\mathbf{H}^{(MRR2)}(b) = \begin{bmatrix} \tilde{\mathbf{x}}'(\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}1}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}1} \\ \vdots \\ \tilde{\mathbf{x}}'(\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}n}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{W}_{\mathbf{r}n} \end{bmatrix}. \tag{B.15}$$

For equality of $\mathbf{H}^{(LLR)}(b)$ and $\mathbf{H}^{(MRR2)}(b)$, it must be shown that $\mathbf{W}_i = \mathbf{W}_{\mathbf{r}i}$. $\mathbf{W}_i$ and $\mathbf{W}_{\mathbf{r}i}$ both depend only on the $\tilde{\mathbf{X}}$ values, not the responses. Thus, at the same value of b, $\mathbf{W}_i = \mathbf{W}_{\mathbf{r}i}$.

$SSE_{\max}$, the largest sum of square error over all possible bandwidth values, essentially, is the parametric SSE by OLS that results when b goes to infinity, as mentioned in Section 2.4.1. Thus, in this study, $SSE_{\max} = \sum_{i=1}^{n}(y_i - \hat{y}_i^{(OLS)})^2 = \sum_{i=1}^{n} e_i^2$ (where $\hat{y}_i^{(OLS)}$ is the i^{th} first order polynomial fit), while $SSE_{\mathbf{r}\max} = \sum_{i=1}^{n}(e_i - \hat{e}_i^{(OLS)})^2$ (where $\hat{e}_i^{(OLS)}$ is the i^{th} first order polynomial fit by considering the residuals as a response). It is easy to see that $\hat{e}_i^{(OLS)} = 0$, for all i, since $\hat{y}_i^{(OLS)}$ is the result of a first order polynomial fit and $e_i = y_i - \hat{y}_i^{(OLS)}$ cannot obtain further a first order polynomial fit. That is, if $\hat{y}_i^{(OLS)} = \mathbf{H}^{(OLS)}\mathbf{y}$, then $\hat{\mathbf{e}}^{(OLS)} = \mathbf{H}^{(OLS)}\mathbf{e} = \mathbf{H}^{(OLS)}(\mathbf{I} - \mathbf{H}^{(OLS)})\mathbf{y} = \mathbf{0}$. Therefore, $SSE_{\max} = SSE_{\mathbf{r}\max}$.

SSE_b in the denominator of the PRESS** by LLR is given by

$$SSE_b = \sum_{i=1}^{n}(y_i - \hat{y}_i^{(LLR)}(b))^2, \tag{B.16}$$

while $SSE_{\mathbf{r}b}$ in the denominator of the PRESS** by MRR2 is given by

$$SSE_{\mathbf{r}b} = \sum_{i=1}^{n}(e_i - \hat{e}_i^{(LLR)}(b))^2. \tag{B.17}$$

(B.16) and (B.17) are equal to each other, since they are the numerators of (B.10) and (B.11), respectively, and it has been proven that (B.10) and (B.11) are equal to each other.

Thus, $PRESS^{**(LLR)}(b) = PRESS^{**(MRR2)}(b)$ with the same numerators and the same denominators. Therefore, $PRESS^{**(LLR)}(b)$ and $PRESS^{**(MRR2)}(b)$ pick up the same value of bandwidth, which achieves the minimum of PRESS**. Therefore, $\hat{y}_0^{(LLR)} = \hat{y}_0^{(MRR2)}$. □

A model general version of Theorem 2 can be stated in the following theorem:

Theorem 3 *If the model matrix of the parametric fit of MRR2 is the same as the model matrix used by the nonparametric fit and* $\lambda = 1$*, then the MRR2 fit at location* $\mathbf{x}_0$ *is equal to the nonparametric fit at* $\mathbf{x}_0$*.*

Proof. It is easy to prove Theorem 3 based on the work of the proof of Theorem 2. □

References

1. Ames, A. E., Mattucci, N., MacDonald, S., Szonyi, G. and Hawkins, D. M. (1997). "Quality Loss Functions for Optimization Across Multiple Response Surfaces". *Journal of Quality Technology* 29, pp. 339-346.

2. Araujo, A. and Assis, F. M. (2000). "An Improved Genetic Algorithm Performance with Benchmark Functions". *Neural Networks, 2000. Proceedings. Sixth Brazilian Symposium* pp. 292.

3. Anderson-Cook, C. M. and Prewitt, K. (2005). "Some Guidelines for Using Non-parametric Methods for Modeling Data from Response Surface Designs". *Journal of Modern Applied Statistical Methods* 4, pp. 106-119.

4. Assaid, C. (1997). "Outlier Resistant Model Robust Regression". Ph.D. Dissertation. Department of Statistics, Virginia Polytechnic Institute & State University, Blacksburg, VA.

5. Back (1996). *Evolutionary Algorithms in Theory and Practice* Oxford University Press, Oxford, New York.

6. Borkowski, J. J. (2003). "Using a Genetic Algorithm to Generate Small Exact Response Surface Designs". *Journal of Probability and Statistical Science* 1(1), pp. 65-88.

7. Carlyle, W. M., Montgomery, D. C. and Runger, G. C. (2000). "Optimization Problems and Methods in Quality Control and Improvement". *Journal of Quality Technology* 32(1), pp. 1-17.

8. Ch'ng, C. K., Quah, S. H. and Low, H. C. (2005). "Index C^*_{pm} in Multiple Response Optimization". *Quality Engineering* 17, pp. 165-171.

9. Cieniawski, S. E., Eheart, J. W., and Ranjithan, S. (1995). "Using Genetic Algorithms To Solve A Multiobjective Groundwater Monitoring Problem". *Water Resources Research* 31(2), pp. 399-409.

10. Cleveland, W. S. (1979). "Robust locally weighted regression and smoothing scatterplots". *Journal of the American Statistical Association* 74, pp. 829-836.

11. Davis, L. (1991). "Performance enhancements". In L. Davis, (ed.), *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold.

12. Derringer, G. (1994). "A balancing act: Optimizing a product's properties". *Quality Progress* 6, pp. 51C58.

13. Derringer, G. C. and Suich, R. (1980). "Simultaneous Optimization of Several Response Variables". *Journal of Quality Technology* 12, pp. 214-219.

14. Design-Expert (1997). *Stat-Ease Inc.* Version 5. Minneapolis, MN.

15. Einsporn, R. (1987). "HATLINK: A Link Between Least Squares Regression and Nonparametric Curve Estimation". Ph.D. Dissertation, Virginia Polytechnic Institute & State University, Blacksburg, VA.

16. Einsporn, R. and Birch, J. B. (1993). "Model robust regression: using nonparametric regression to improve parametric regression analysis". Technical Report 93-5. Department of Statistics, Virginia Polytechnic Institute & State University, Blacksburg, VA.

17. Eshelman, L. J., Caruna, R. A., and Schaffer, J. D. (1989). "Biases in the crossover landscape". *Proceedings of the 3rd International Conference on Genetic Algorithms and Their Applications*, Morgan Kaufmann Publishers, San Mateo, CA, 1989, pp. 10-19.

18. Fan, J. and Gijbels, I. (1996). *Local Polynomial Modeling and Its Applications*. Chapman and Hall, London.

19. Fan, J. and Gijbels, I. (2000). "Local polynomial fitting". In: Schimek, M.G. (Ed.), *Smoothing and Regression: Approaches, Computation, and Application*. Wiley, New York, pp. 229-276.

20. Francisco Ortiz, Jr., Simpson, J. R., Pignatiello, J. J., Jr, and Heredia-Langner, A. (2004). "A Genetic Algorithm Approach to Multiple-Response Optimization". *Journal of Quality Technology* **36**(4), 432-450.

21. Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning* Reading, MA: Addison-Wesley.

22. Hamada M., Martz, H. F., Reese, C. S. and Wilson, A. G. (2001). " Statistical Practice: Finding Near-Optimal Bayesian Experimental Designs via Genetic Algorithms". *The American Statistician* 55(3), pp. 175-181.

23. Heredia-Langner, A., Carlyle, W. M., Montgomery, D. C., Borror, C. M. and Runger, G. C. (2003). " Genetic Algorithms for the Construction of D-Optimal Designs". *Journal of Quality Technology* 35(1), pp. 28-46.

24. Heredia-Langner, A., Montgomery, D. C., Carlyle, W. M. and Borror, C. M. (2004). " Model-Robust Optimal Designs: A Genetic Algorithm Approach". *Journal of Quality Technology* 36(3), pp. 263-279.

25. Hardle, W. (1990). *Applied Nonparametric Regression.* Cambridge Univ. Press, London.

26. Hardle, W., Muller, M., Sperlich, S. and Werwatz, A. (2004). *Nonparametric and Semiparametric Models.* Springer, Berlin.

27. Haupt, R. L. and Haupt, S. E. (2004). *Practical Genetic Algorithms.* John Wiley and Sons, Inc., New York, NY.

28. Hinkelmann, K. and Kempthorne, O. (1994). *Design and Analysis of Experiments Volume 1: Introduction to Experimental Design.* John Wiley and Sons, Inc., New York, Chichester, Brisbane, Toronto and Singapore.

29. Holland J. H. (1992). *Adaption in Natural and Artificial Systems: an Introduction Analysis with Applications to Biology, Control, and Artificial Intelligence* A Bradford Book: The MIT Press, Cambridge, Massachusetts, London, England.

30. Kim, K. and Lin, D. (2000). "Simultaneous optimization of multiple responses by maximining exponential desirability functions". *Applied Statistics (Journal of the Royal Statistical Society: Series C)* 49 (3), pp. 311-325.

31. Kim, K. and Lin, D. K. J. (2006). "Production, Manufacturing and Logistics Optimization of multiple responses considering both location and dispersion effects". *European Journal of Operational Research* 169, pp. 133-145.

32. Khuri, A. I. (1996). *Multiresponse surface methodology.* In: Ghosh, A., Rao, C.R. (Eds.), *Handbook of Statistics: Design and Analysis of Experiments* 13, pp. 377-406.

33. Khuri, A. I. and Conlon, M. (1981). "Simultaneous Optimization of Multiple Responses Represented by Polynomial Regression-Functions". *Technometrics* 23, pp. 363-375.

34. Kim, K. J. and Lin, D. K. J. (2006). "Optimization of Multiple Responses Considering Both Location and Dispersion Effects". *European Journal of Operational Research* 169, pp. 133-145.

35. Kros, J. F. and Mastrangelo, C. M. (2001). "Comparing Methods for the Multiresponse Design Problem". *Quality and Reliability Engineering International* 17, pp. 323-331.

36. Lind, E. E., Goldin, J. and Hichman, J. B. (1960). "Fitting Yield and Cost Response Surfaces". *Chemical Engineering Progress* 56, pp. 62.

37. Mayer, D. G., Belward, J. A. and Burrage, K. (1996). "Use of advanced techniques to optimize a multi-dimensional dairy model". *Agricultural Systems* 50, pp. 239-253.

38. Mayer, D. G., Belward, J. A. and Burrage, K. (1999a). "Performance of genetic algorithms and simulated annealing in the economic optimization of a herd dynamics model". *Environment International* 25, pp. 899-905.

39. Mayer, D. G., Belward, J. A. and Burrage, K. (1999b). "Survival of the fittest—genetic algorithms versus evoluation strategies in the optimization of systems models". *Agricultural Systems* 60, pp. 113-122.

40. Mayer, D. G., Belward, J. A. and Burrage, K. (2001). "Robust Parameter Settings of Evolutionary Algorithms for the Optimization of Agricultural Systems Models". *Agricultural Systems* 69, pp. 199-213.

41. Mays, J. E. and Birch, J. B. (1998). "Smoothing Considerations in Nonparametric and Semiparametric Regression". Technical Report Number 98-2. Department of Statistics, Virginia Polytechnic Institute & State University, Blacksburg, VA.

42. Mays, J. E. and Birch, J. B. (2002). "Smoothing for Small Samples with Model Misspecification: Nonparametric and Semiparametric Concerns". *Journal of Applied Statistics* 29, pp. 1023-1045.

43. Mays, J. E., Birch, J. B. and Starnes, B. A. (2001). "Model robust regression: combining parametric, nonparametric, and semiparametric methods". *Journal of Nonparametric Statistics* 13, pp. 245-277.

44. Montgomery, D. C. (1999). "Discussion". *Journal of Quality Technology* 31(1), pp. 45-46.

45. Myers, R. H. (1990). *Classical and Modern Regression with Applications.* PWS-Kent, Boston, MA

46. Myers, R. H. (1999). "Response Surface Methodology—Current Status and Future Directions". *Journal of Quality Technology* 31(1), pp. 30-44.

47. Myers, R. H. and Montgomery, D. C. (2002). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments.* John Wiley and Sons, Inc., New York, NY.

48. Myers, R. H., Montgomery, D. C., Vining, G. G., Borror, C.M. and Kowalski, S. M. (2004)."Response Surface Methodology: A Retrospective and Literature Survey". *Journal of Quality Technology* 36 (1), pp. 53-77.

49. Nadaraya, E. (1964). "On estimating regression". *Theory of Probability and Its Applications* 9, pp. 141-142.

50. Peck, C. C. and Dhawan, A. P. (1995)." Genetic algorithms as global random search methods: an alternative perspective". *Evolutionary Computation* 3, pp. 39-80.

51. Pickle, S. M. (2006). "Semiparametric Techniques for Response Surface Methodology". Ph.D. Dissertation. Department of Statistics, Virginia Polytechnic Institute & State University, Blacksburg, VA.

52. Pickle, S. M., Robinson, T. J., Birch, J. B. and Anderson-Cook, C. M. (2006). "A Semi-Parametric Approach to Robust Parameter Design". *Journal of Statistical Planning and Inference*. (to appear in 2007)

53. Pignatiello, J. J., Jr.(1993). "Strategies for Robust Multiresponse Quality Engineering". *IEE Transaction* 25, pp. 5-15.

54. Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (1992). *Numerical Recipes in C* Cambridge, University Press.

55. Radcliff, N. J. (1991). *Forma Analysis and Random Respectful Recombination*. In Proc. 4th Int. Conf. on Genetic Algorithm, San Mateo, CA: Morgan Kauffman.

56. Rencher, A. C. (2002). *Method of multivariate analysis*. John Wiley and Sons, Inc., New York.

57. Shah, K. H., Montgomery, D. C. and Carlyle, W. M. (2004). "Response Surface Modeling and Optimization in Multiresponse Experiments Using Seemingly Unrelated Regressions". *Quality Engineering* 16 (3), pp. 387-397.

58. Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, New York.

59. Simonoff, J. S. (1996). *Smoothing Methods in Statistics*. Springer-Verlag, New York.

60. Takezawa, K. (2006). *Introduction to Nonparameteric Regression*. John Wiley and Sons, Inc., New Jersey.

61. Tseo, C. L., Deng, J. C., Cornell, J. A., Khuri, A. I. and Schmidt, R. H. (1983). "Effect of washing treatment on quality of minced mullet flesh". *Journal of Food Science* 48, pp. 163-167.

62. Vining, G. G. (1998). "A Compromise Approach to Multi-response Optimization". *Journal of Quality Technology* 30, pp. 309-313.

63. Vining, G. G. and Bohn, L. L. (1998). "Response surfaces for the mean and variance using a nonparametric approach". *Journal of Quality Technology* 30, pp. 282-291.

64. Watson, G. (1964). "Smoothing regression analysis". *Sankhya Series A*26, pp. 359-372

65. Wu, S. J. and Chow, P. T. (1995). " Steady-State Genetic Algorithms for Discrete Optimization of Trusses". *Computers and Structures* 56(6), pp. 979-991.

Vita

Wen Wan was born in Nanchang, Jiangxi, China. In 1993, she graduated from Nanchang Number Two High School in Nanchang. In July, 1998, she graduated from Shandong Medical University, Jinan, Shandong, China and received a Bachelor of Medical Degree in Medical Science. She then worked as an Obstetrician and Gynecologist in Maternal and Child Health Hospital of Jiangxi Province, China, for two years. In September, 2000, she entered Sun Yat-sen University of Medical Sciences, Guangzhou, China, for her graduate study in Pharmacological Ophthalmology. During her graduate study, the course Biostatistics started her interest in Statistics. She started her graduate study in Statistics in August, 2002, at Virginia Polytechnic Institute & State University. She received a Master of Science degree in Statistics in December, 2003, and is expected to complete work for a Ph.D. degree in Statistics in October, 2007. She will be joining the Medical Statistics Section, Department of Medicine and Biostatistics and Bioinformatics Unit, Comprehensive Cancer Center, at the University of Alabama at Birmingham, as a research assistant professor, in October, 2007.