CONVEX VERSIONS OF MULTIVARIATE ADAPTIVE REGRESSION SPLINES AND IMPLEMENTATIONS FOR COMPLEX OPTIMIZATION PROBLEMS

by

DACHUAN THOMAS SHIH

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2006

To my mother Mei-Yueh Chuang Shih and my father Wen-Sheng Shih

who set the example and who made me who I am.

ACKNOWLEDGEMENTS

I would like to thank my supervising professors Dr. Victoria Chen and Dr. Seoung Bum Kim for constantly motivating and encouraging me, and also for their invaluable advice during the course of my doctoral studies. I wish to thank my academic advisors, Dr. Shiek Imrhan, Dr. Chien-Pai Han, and Dr. Jay Rosenberger, for their interest in my research and for taking time to serve in my dissertation committee.

I would also like to extend my appreciation to my sister, Alice S. F. Shih, and my brother-in-law, Ben J. P. Chen, for providing financial support for my doctoral studies. In particular, without the support from Alice, I could not have pursued my Ph.D. here in Texas.

I am grateful to all the teachers who taught me during the years I spent in school, first in Taiwan, then in the Unites States. I would like to thank Dr. Su-Fen Yang, for encouraging and inspiring me to pursue graduate studies. I would also like to thank Dallas-Fort Worth International Airport for supporting my graduate research assistantship in 2006.

Finally, I would like to express my deep gratitude to my wife, Chao-ping Feng, who have encouraged and inspired me and sponsored my graduate studies. I am extremely fortunate to be so blessed. I am also extremely grateful to my parents and sisters for their sacrifice, encouragement and patience. I also thank several of my friends and

COSMOS lab, Sung, Punnakitikashem, Sundaramoorthi, Pilla, Siddappa, Visoldilokpun, Wen, Suebvisai, Fan, Hwang, Temiyasathit, and Tarun who have helped me throughout my studies here.

October 24, 2006

ABSTRACT

CONVEX VERSIONS OF MULTIVARIATE ADAPTIVE REGRESSION
SPLINES AND IMPLEMENTATIONS FOR COMPLEX
OPTIMIZATION PROBLEMS

Publication No. ______

DACHUAN THOMAS SHIH, Ph.D.

The University of Texas at Arlington, 2006

Supervising Professors: Victoria C. P. Chen and Seoung Bum Kim

Multivariate Adpative Regression Splines (MARS) provide a flexible statistical modeling method that employs forward and backward search algorithms to identify the combination of basis functions that best fits the data. In optimization, MARS has been used successfully to estimate the value function in stochastic dynamic programming, and MARS could be potentially useful in many real world optimization problems where objective (or other) functions need to be estimated from data, such as in simulation optimization. Many optimization methods depend on convexity, but a nonconvex MARS approximation is inherently possible because interaction terms are products of univariate terms. In this dissertation, convex versions of MARS are proposed. In order to ensure MARS convexity, two major modifications are made: (1) coefficients are constrained such that pairs of basis functions are guaranteed to jointly form convex functions; (2)

The form of interaction terms is appropriately changed. Finally, MARS convexity can be achieved by the fact that the sum of convex functions is convex.

The implementation of MARS for approximating complex optimization functions can involve hundreds to thousands of state or decision variables. In particular, this research studies application to an inventory forecasting stochastic dynamic programming problem and an airline fleet assignment problem. Although one can simply attempt a MARS approximation over all the variables, prior research on the fleet assignment application indicates that many variables have little effect on the objective. Thus, a data mining step to conduct variable selection is needed. This step separates potentially critical variables from clearly redundant ones. In this dissertation, variants of two data mining tools are explored separately and in combination for variable selection : regression trees and multiple testing procedures based on false discovery rate.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

1.1 Motivation

The central objective of operations research is optimization [21], i.e., “to do things best under the given circumstances.” This general concept has a great many applications, for instance, in airline fleet assignment [41], distribution of goods and resources [3], emergency and rescue operations [2], environmental management [58], financial planning [39], inventory control [15], manpower and resource allocation [12], manufacturing of goods [9], production process control [42], risk management [36], sequencing and scheduling of tasks [37], telecommunications [29], and traffic control [48]. The well-known problem of maximizing a linear function over a convex polyhedron is known as linear programming [22, 35], and the general problem of convex optimization is to find the minimum of a convex (or quasiconvex) function on a finite-dimensional convex body [50].

Computer modeling is having a profound effect on scientific research. Many processes are so complex that physical experimentation is too time consuming or too expensive; or, as in the case of weather modeling, physical experiments may simply be impossible. As a result, experiments have increasingly turned to mathematical models to simulate these complex systems. Advances in computational power have allowed both greater complexity and more extensive use of such models. Virtually every area of science, engineering, and technology is affected. A computer experiment [43] is a number of runs of the code with various inputs. Often, the codes are computationally expensive to run, and a common objective of an experiment is to obtain a computationally-efficient response surface approximation (a.k.a., metamodel) of the output. The purpose of design and analysis of computer experiments (DACE) is to provide methods for conducting computer experiments to build a metamodel that can be efficiently employed to improve the performance of a complex system. The computer experiment is typically a simulation model [?], but in this dissertation, the computer experiments are all optimization models. Recent reviews of DACE methods are given in Chen, Tsui, Barton and Allen [19] and Chen, Tsui, Barton and Meckesheimer [20].

Multivariate adaptive regression splines (MARS) have been applied in DACE-based approaches for continuous-state stochastic dynamic programming (SDP, [13, 15, 16, 18, 53, 52, 57]), Markov decision processes (MDP, [17, 47]), and, most recently, two-stage stochastic programming (SP, [41]). The DACE-based SDP and MDP approaches used an experimental design to discretize the continuous (or near-continuous) state space, and then used MARS to approximate the continuous value function over the state space. In particular, the MDP application studied an airline revenue management problem with the objective of more accurately estimating the fair market value of a seat over time. The two-stage SP problem studied an airline fleet assignment model that seeks an assignment of aircraft in the first stage that swapping of crew-compatible aircraft can be achieved in the second stage to maximize expected revenue. The DACE approach for SP was used to create a MARS approximation of the first-stage expected profit objective function, so as to speed up the first-stage optimization.

Under the assumption that an optimization function f is convex, it is desired that the response surface model $\hat{f}$ that estimates f be convex as well. For example, in the above-mentioned SDP, MDP, and SP problems, the underlying function is theoretically convex. Convexity is not a typical assumption of statistical modeling methods, and a specialized approach must be developed. There are several options for DACE metamodeling, including polynomial response surface models [7], spatial correlation models, a.k.a., kriging [43], MARS, regression trees [11, 26], and artifical neural networks [31]. However, of these, only parametric polynomial models and MARS can be adapted to accommodate the convexity constraint. Because parametric polynomial models are known to be insufficient for complex approximations, MARS is the best option. Convex-MARS uses the modification of both the MARS basis functions and algorithms to build a sum of convex functions; therefore, the final approximation will be convex.

In practice, large-scale complex optimization can involve hundreds or thousands of variables. Although in a DACE-based approach one can simply attempt a MARS approximation over all these variables, many of the variables may have little effect on the performancce of objective. Thus, in order to be computationally-efficient, a data mining step to conduct variable selection is needed. This step separates potentially critical variables from clearly redundant ones. Data mining is the process of exploration and analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns and rules (Berry and Linoff [5]). Statistical data mining is exploratory data analysis with little or no human interaction using computationally feasible techniques, i.e., the attempt to find unknown interesting structure (Wegman [56]).

The problem of variable selection or subset selection arises in data mining when one wants to explain a response variable of interest by means of a subset of the candidate explanatory variables, but there is uncertainty about which subset to use. Variable selection is particularly of interest when the number of candidate explanatory variables is large, and many redundant or irrelevant variables are thought to be present. Variable selection methods have been developed in regression problems, where the number of observations (n) is larger than the number of variables (p). Since in a DACE-based approach, each observation requires a computationally-expensive computer experiment run, variable selection should ideally be conducted after a small number of computer experiment runs, likely fewer runs (n) than the number of variables (p). In order to address this “large p and small n” problem, this dissertation studies a multiple testing procedure based on the false discovery rate (FDR,[4]) and regression trees [11].

1.2 Research framework

This dissertation addresses two critical components for optimization computer experiments:

1. Identification of potentially important input explanatory variables via data mining variable selection for the “large p and small n” problem.
2. Convexity of a MARS response surface model fit to computer experiment data.

The approaches proposed in this dissertation and existing methods explored are listed in Table 1.1 and described briefly below. Although variable selection is conducted prior to fitting Convex-MARS, the latter will be presented first in this dissertation.

Table 1.1.Summary of the related methodologies in this dissertation

	Proposed Methodologies	Existing Methodlogies
Statistical Data Mining	FDR/CART	CART
	Inverse FDR	FDR
Statistical Modeling	Convex-MARS-I	MARS
	Convex-MARS-II	ASR-MARS
	Convex-MARS-II-T	

Convexity of the MARS approximation requires re-structuring and constraining the basis functions that comprise MARS. Two versions of re-structuring of the MARS interaction basis functions are developed, where the Convex-MARS-I is simpler, but Convex-MARS-II is more flexible. MARS uses a forward search to select basis functions and a backward search to “prune” basis functions. At each iteration, the method of least squares, which has a closed form solution, is used to fit the coefficients of the basis functions. Rather than creating a constrained least squares optimization problem that would require a numerical optimization method, the methods in this dissertation modify the forward and backward algorithms to specifically check that the least squares estimates satisfy the constraints for convexity. A variant of Convex-MARS-II (Convex-MARS-II-T) is developed that sets a conservative threshold on satisfaction of the constraints to speed up the search algorithms. Ultimately, convexity is guaranteed by constructing an approximation that is a sum of convex terms.

To address the "large p and small n" variable selection problem, this dissertation studies regression trees [11] and a multiple testing procedure based on the FDR [4]. The algorithm of $\underline{c}$lassification $\underline{a}$nd $\underline{r}$egression $\underline{t}$rees (CART) uses binary recursive partitioning to separate the explanatory variable space into rectangular regions based on similarity of the response (performance measure) values. Variables that are used to partition the space are considered to be important. FDR uses the p-values from separate hypothesis tests on each of the explanatory variables, and the key is identifying the threshold p-value that separates the important variables from the unimportant ones. However, FDR procedures were traditionally developed for classification, and not regression, so two new versions are developed to handle the continuous or near-continuous nature of our optimization functions that form the underlying response surfaces. The first FDR version uses CART to categorize the the performance measure observations into a finite set of groups (FDR/CART). The second version develops Inverse FDR, in which the groupings are based on the explanatory variables, instead of on the responses. Further discussion about statistical modeling methods for DACE and data mining approaches are given in Chapter 2. Background on the MARS algorithm will also be provided in Chapter 2. Convex versions of MARS will be described in Chapter 3 with computational results using an inventory forecasting SDP problem [18, 15]. Chapter 4 presented the variable selection approaches and computational results using a two-stage SP airline fleet assignment model [41]. Finally, the conclusions and future work are given in Chapter 5.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Statistical Modeling Methods for Regression

Many references on statistical modeling methods for computer experiments and data mining may be found in Chen, Tsui, Barton and Allen [19], Chen, Tsui, Barton and Meckesheimer [20] and Tsui, Chen, Jiang and Aslandogan [54]. A brief explanation of candidate methods for DACE-based optimization is given below.

2.1.1 Polynomial Response Surface Models

Box and Wilson [8] proposed response surface modeling. The main goal was to identify the levels of the explanatory variables $\boldsymbol{x}$ over the experimental region $\mathcal{D}$ that optimized the mean response η. Steepest ascent with first-order models were used to move towards the region of the optimum, and then second-order models were used to find a stationary point that defined a maximum, minimum, or saddle point. The general form of a response surface (RS) model is a polynomial of degree d:

$$\hat{f}(\boldsymbol{x};\boldsymbol{\beta}) = \beta_0 + \sum_j \beta_j x_j + \sum_j \sum_{k>j} \beta_{jk} x_j x_k + \sum_j \beta_{jj} x_j^2$$
$$\sum_j \sum_{k>j} \sum_{l>k} \beta_{jkl} x_j x_k x_l + \ldots + \sum_j \beta_{j,j,\ldots,j} x_j^d.$$

RS models are relatively easy to use, since they are linear models, and the linear model tools may be applied. The disadvantage is that if the selected polynomial form does not match the true response surface, then the model is inadequate. More references for RS models can be found in Box and Draper [7] and Myers and Montgomery [40].

2.1.2 Spatial Correlation Models

Spatial correlation or "kriging," models first appeared in the field of geostatistics [38]. These models are now prevalent in the area of spatial statistics [33]. Kriging for computer experiments uses a Bayesian perspective (e.g., [43]) to model each computer experiment observation as a realization from a stochastic process, even though the computer experiment is deterministic. The general form models the "stochastic" response Y as a function of the explanatory variables $\boldsymbol{x}$:

$$Y(\boldsymbol{x}) = \textstyle\sum_{m=1}^{M} \beta_m h_m(\boldsymbol{x}) + Z(\boldsymbol{x}),$$

where the linear model component consists of known functions $h_m(\boldsymbol{x})$ with unknown coefficients β_m, and the stochastic component $Z(\boldsymbol{x})$ is a random process, commonly assumed to be Gaussian, with mean zero and covariance functions (covariogram) decayed with distance.

2.1.3 Tree-based Models

CART, developed by Breiman et al. [11], has been a popular data mining tool for supervised learning in data mining. It is essentially an approach utilizing recursive partitioning (binary splits), with a forward stepwise procedure that adds model terms and backward procedure for pruning. The model terms partition the explanatory variable $\boldsymbol{x}$-space into disjoint hyper-rectangular regions via indicator functions:

$$b^{+}(x;k) = 1\{x > k\}, \quad b^{-}(x;k) = 1\{x \leq k\}, \tag{2.1}$$

where the "split-point" k defines the separation of the regions. The model terms B_m are the basis functions of the approximation with unknown coefficients. A univariate basis function is a single indicator function; while an interaction basis function is a product of univariate basis functions. The CART model form is then

$$f(\boldsymbol{x};\boldsymbol{\beta}) = \beta_0 + \sum_{m=1}^{M} \beta_m B_m(\boldsymbol{x}). \tag{2.2}$$

The partitioning of the $\boldsymbol{x}$-space does not keep the parent model terms since they are redundant. For instance, assume the current set has the model term:

$$B_a(\boldsymbol{x}) = 1\{x_3 > 6\} \cdot 1\{x_4 \leq 8\}, \tag{2.3}$$

and the forward stepwise algorithm chooses to add

$$B_b(\boldsymbol{x}) = B_a(\boldsymbol{x})\{x_7 > 11\} = 1\{x_3 > 6\} \cdot 1\{x_4 \leq 8\} \cdot 1\{x_7 > 11\}. \tag{2.4}$$

Then the model term, in this case a parent basis function, $B_a(\boldsymbol{x})$ is dropped from the current set. Thus, the recursive partitioning algorithm produces a binary tree with the current set of model terms $B_m(\boldsymbol{x})$ consisting of the M leaves or terminal nodes of the tree, each of which corresponds to a different region R_m. In particular, the final regression tree approximation tends to consist of higher-order interactions instead of lower-order terms, which is the major drawback of CART. For the case of regression, CART minimizes the squared error loss function, and the approximation is a piecewise-constant function. In the classification regression tree, each region R_m is classified to one of the C classes. The algorithm is exactly the same as for regression, but with a different loss function.

Friedman et al. [27] introduced the concepts of boosting (Freund and Schapire [24]; Schapire [44]) and bagging (Breiman [10]). The bagging approach fits the decision tree model several times on boostrap subsamples and uses the average of the decision trees. The idea of boosting is to improve the accuracy of a given learning algorithm. The boosted trees or multiple adaptive regression trees (MART, Friedman [26]) consist of lower-order interaction terms. More detail can be found in Hastie et al. [30] and Chen et al. [19].

2.1.4 Artificial Neural Network (ANN)

Artificial neural network (ANN) models are very popular in engineering (see Lippmann [34] or Haykin [31] for details). Like many non-linear data modeling approaches, ANNs can be used to model complex relationships between response and explanatory variables. In general, an ANN model is represented by a diagram of nodes in various layers with weighted connections between nodes in different layers. This structure can be used to represent a variety of statistical model forms, including multiple linear regression and projection pursuit regression (Hastie et al. [30]). Nodes in the input layer represent the explanatory variables, and nodes in the output layer represent the response variable(s). For a linear regression model, there is no hidden layer, but more complex models require at least one "hidden" layer in between the input and output layers. Transformations between layers (e.g., input to hidden) are defined by activation functions, of which there are many forms (Haykin [31]).

2.1.5 Multivariate Adaptive Regression Splines (MARS)

Friedman [25] developed MARS as a statistical method for high-dimensional modeling with interactions. The MARS model is a linear statistical model with a forward stepwise algorithm to select model terms followed by a backward procedure to find the best subset of the model terms. Prior to MARS, Friedman and Silverman [28] developed a univariate(additive) version. A MARS approximation bends to model curvature at various "knot" locations, and one of the objectives of the forward stepwise algorithm is to select appropriate knots. MARS is flexible yet can be easily implemented, with the computational effort dependent in part on the number of basis functions added to the model. The MARS approximation has the general form

$$\hat{f}_M(\boldsymbol{x};\beta) = \beta_0 + \sum_{m=1}^{M} \beta_m B_m(\boldsymbol{x}), \tag{2.5}$$

where $B_m(\boldsymbol{x})$ initially is a basis function of the form described below in Equation (2.6) that later can be smoothed, M is the number of linearly independent basis functions, and β_m is the unknown coefficient for the m-th basis function. In the forward stepwise algorithm, univariate basis functions are truncated linear functions,

$$b^{+}(x;k) = [+(x-k)]_{+}, \quad b^{-}(x;k) = [-(x-k)]_{+}, \tag{2.6}$$

where $[q]_{+} = \max\{0, q\}$ and k is a univariate knot. Basis functions in pairs corresponding to the two forms above are added in the algorithm. Each explanatory variable is separately assigned a set of eligible knots, and the knots are chosen to coincide with the input levels represented in the data. Interaction basis functions are created by multiplying an existing basis function with the two forms of the truncated linear function involving a new variable. Both the existing basis function and the newly created interaction basis function are considered in the MARS approximation. Thus, the form of the m-th basis function can be written as

$$B_m(\boldsymbol{x}) = \prod_{l=1}^{L_m} [s_{l,m} \cdot (x_{v(l,m)} - k_{l,m})]_{+} \ , \tag{2.7}$$

where $x_{v(l,m)}$ is the explanatory variable corresponding to the l-th truncated linear function in the m-th basis function, $k_{l,m}$ is the knot value corresponding to $x_{v(l,m)}$, and $s_{l,m}$ is $+1$ or -1. The search for new basis functions can be restricted to interactions of a maximum order (e.g., $L_m \leq 3$ permits up through three-factor interactions). Using a generalized cross-validation selection criterion, basis functions are added a pair at a time; i.e., both forms of Equation (2.6). The algorithm stops when $M_{\max}$ basis functions have been selected, where $M_{\max}$ is user-specified. In most cases, two-way or three-way interactions should be sufficient. More details on MARS will be covered in Section 2.2.

2.1.6 Support Vector Machines for Regression

Vapnik, Steven Golwich, and Alex Smola [55] proposed a version of a support vector machines for regression, called support vector regression (SVR) in 1997, which is a supervised learning method. The idea of SVR is to simultaneously minimize the empirical classification error rate and maximize the geometric margin. The model produced by SVR only depends on a subset of the training data, because the cost function for building the model ignores any training data that is in the neighborhood of the model prediction. Further details on SVR can be referred in Collobert and Sengio [23] and Hastie et al. [30].

2.2 Versions of MARS

Since this dissertation focuses on MARS, further details are provided in this section. Section 2.2.1 describes the smoothed version of MARS. Then parallelized versions of MARS [1][51] are introduced in Section 2.2.2. Finally, Section 2.2.3 presents flexible implementations of MARS [52].

2.2.1 Smoothing MARS

The DACE-based SDP problems call for a nonlinear minimization method because the future value functions and their MARS approximations are both nonlinear. One of the most efficient ways to find the optimum is using first and second derivatives. Clearly, a piecewise-linear approximation like MARS, does not possess continuous derivatives. Friedman's MARS provides a continuous first derivative and a continuous second derivative everywhere except at the side knots by replacing the truncated linear basis functions $[\pm(x-k)]_+$ in the forward and backward stepwise algorithms with cubic functions. To give MARS continuity and a continuous second (including first) derivative at the side knots, quintic functions derived by Chen et al. [14] are used in place of Friedman's cubic functions. Chen [14] discovered that when the center knot k is too far from the midpoint between k_- and $k+$, nonconvexities are possible in the cubic and quintic basis functions. Specifically, when we have

$$\begin{aligned} \frac{k_+ - k}{k_+ - k_-} &< \frac{2}{5} \quad \forall s = 1, \quad \text{or} \\ \frac{k - k_-}{k_+ - k_-} &< \frac{2}{5} \quad \forall s = -1. \end{aligned} \tag{2.8}$$

2.2.2 Parallel Computing for MARS

The idea for parallel computing is based on the fact that the task of solving a problem usually can be divided into a number of smaller jobs, which may be carried out simultaneously. In general, parallel computing refers to solving intensive computational problems via parallel computers in order to obtain the solution faster. A parallel computer is a set of processors working jointly to solve a computational problem. Two versions of MARS parallel computing algorithms have been developed. One of them uses a B-splines version of MARS [1], developed for numerical stability based on pure piecewise-linear basis functions. The other was developed by Tsai [51] and was utilized within a decision-making framework (DMF) based on Chen et al. [15].

2.2.3 Flexible Implementations of MARS

Two flexible implementations of MARS were constructed by Tsai [51]: (i) Automatic Stopping Rules for MARS (ASR-MARS) and (ii) Robust MARS. Instead of using the original MARS stopping rule that depends on a user-specified $M_{\max}$ in the MARS forward stepwise algorithm, ASR-MARS stops automatically based on the usage of the coefficient of determination or adjusted coefficient of determination. Robust MARS aims to obtain a MARS approximation that is less sensitive to extreme points through selecting the lower-order terms over the high-order ones given the contributions or fits are comparable.

(i) MARS with Automatic Stopping Rules

Two automatic stopping rules are established based on the changes in the (adjusted) coefficient of determination. The first automatic stopping rule (ASR-I) computes the difference between the previous R^2 (or R_a^2) and the current R^2 (or R_a^2). The second automatic stopping rule (ASR-II) is similar to ASR-I but the proportion of the change in R^2 and R_a^2 is considered as the basis for comparison. The accepted tolerance for the difference or the proportion of the change are user-specified.

(ii) Robust MARS

Robust MARS essentially creates an approximation function with lower-order interactions that has comparably good fit to the data as the original MARS approximation. It is considered "robust" because lower-order interactions are less sensitive to extreme points.

2.3 Data Mining

In a multiple hypothesis test, evaluating the number of false positives is essential since a pure use of individual inference procedures leads to a significant number of false positives (Benjamini and Hochberg [4]). Table 2.1 shows the possible outcomes from m hypothesis tests. Family-Wise Error Rate (FWER), traditionally used as a family error rate in the scope of multiple hypothesis testing, is defined as the probability of creating one or more false rejections, i.e.,

$$\mathrm{FWER} = \Pr\left[V \geq 1\right], \tag{2.9}$$

where V is the number of rejected hypotheses given the null hypothesis is true.

Table 2.1.Outcomes from the multiple hypothesis tests of size m

	Accept H_{0i}	Reject H_{0i}	Total
True H_{0i}	U	V	m_0
False H_{0i}	T	S	m_1
Total	W	R_j	m

Shaffer [45] summarized a number of methods controlling the FWER. One of the most extensively used methods is the Bonferroni method, which rejects H_{0i} if $p_i \leq \alpha_i$, where p_i is the p-value of the i-th hypothesis (i.e., H_{0i}). Typically, α_i is set identically for all hypotheses (e.g., $\alpha_i = \frac{\alpha}{m}$). Thus, the overall FWER is less than α. Other family-wise methods were developed to enhance the power of the Bonferroni method. However, they tend to be overly conservative in terms of detecting false hypotheses. Otherwise stated, they can barely reject the null hypothesis given it is false.

Benjamini and Hochberg [4] introduced FDR, defined as the expected proportion of false positives out of all rejected null hypotheses. The advantage of FDR is to identify as many significant hypotheses as possible while maintaining a small number of false positives (Storey and Tibshirani [49]; Kim et al. [32]). With a large family of hypotheses, the advantages over FWER are substantial. In Table 2.1, R_j is the number of rejected null hypotheses, and V is the number of falsely rejected null hypotheses. Then FDR is defined as

$$E\left[\frac{V}{V+S}\right] = E\left[\frac{V}{R_j}\right]. \tag{2.10}$$

Benjamini and Hochberg [4] proved that an ordered p-value controls the specified FDR. The general FDR-procedure to determine significant variables is as follows: Consider a set of hypotheses, p-values, and ordered p-values, denoted H_{0i}, p_i, and $p_{(i)}$, respectively.

- Select a constant α, where $0 \leq \alpha \leq 1$.

- Find $\hat{i} = \max\left[i : p_{(i)} \leq \frac{i}{m} \cdot \frac{\alpha}{\pi_0}\right]$, where $\pi_0 (= \frac{m_0}{m})$ denotes the proportion of true H_{0i}.
- If $\hat{i} \geq 1$, $\Omega = \{\text{All rejected } \mathrm{H}_{0i} \text{ with } p_i \leq p_{(\hat{i})}\}$ with $\mathrm{FDR}(\Omega) \leq \alpha$; elseif $\hat{i} = 0$, do not reject any hypothesis since $\Omega = \emptyset$.

Kim et al. [32] described an approach to statistical inference of independence of categories in each cell in the large-scale contingency table within the FDR framework and applied their procedure to identify a number of amino acids involved in β-sheet bridges in protein.

CHAPTER 3

CONVEX MARS

This chapter focuses on the development of Convex-MARS. A detailed discussion for convex univariate basis functions is presented in Section 3.1 followed by Section 3.2, which explains in detail the necessity of building a convex form of interaction terms. The concepts of convex sets and convex functions are particularly important in optimization, and derivative calculations are often employed for efficient optimization. To guarantee MARS convexity, two major modifications are made: (1) coefficients are constrained such that pairs of univariate basis functions are guaranteed to jointly form convex functions; (2) The form of interaction basis functions is appropriately changed. A convexity proof is provided for the pairs of univariate basis functions in quintic form in Section 3.3, and in Section 3.4.2 this proof is extended for the new convex form of the interaction basis functions. Two versions of Convex-MARS are derived. Section 3.4 describes Convex-MARS-I, and Section 3.5 describes Convex-MARS-II. The primary difference is the convex form of the interaction basis functions, where Convex-MARS-I is simpler, but Convex-MARS-II is more flexible. In addition, Section 3.5.5 describes Convex-MARS-II-T, a variant for Convex-MARS-II.

Modifications of the MARS algorithms are needed to fit Convex-MARS. Friedman's MARS Forward Stepwise Algorithm [25], used to adaptively select basis functions based on lack-of-fit, is shown in Algorithm 2. Friedman refers to the explanatory variables as *covariates*, so this terminology is used in this chapter. Both Convex-MARS-I and Convex-MARS-II require the following algorithms:

- Convex Interaction Transformation Algorithm (CIT) (Algorithms 1 and 5).
- Forward Coefficient Restriction Algorithm (FCR) (Algorithms 3, 6, and 8).
- Backward Iteration of Pruning and Refitting (BIPR) (Algorithms 4 and 7).

The CIT algorithms create the convex forms of the interaction basis functions. The FCR algorithms are modifications of Friedman's Forward Stepwise Algorithm that incorporates convexity restrictions on the model coefficients while selecting basis functions. Following this, the BIPR algorithms check for nonconvexities and eliminate them. In the original implementation, the Convex-MARS algorithm could build the approximation by iterating between the FCR and BIPR algorithms. However, it was discovered that setting a stricter (more conservative) threshold on the convexity restrictions in the FCR algorithm might yield better approximation faster, and this led to Convex-MARS-II-T that uses Algorithm 8. Finally, Section 3.6.1 presents computational results using a four-dimensional and a nine-dimensional inventory forecasting stochastic dynamic program studied by Chen et al. [18]. The value function is known to be convex, and the quality of fit produced by Convex-MARS is compared to that of ASR-MARS [51].

3.1 Convex Univariate Terms

A univariate basis function is either unpaired or one of a pair added corresponding to the two forms in Equation (2.6). An unpaired univariate basis function takes on only one of the forms in Equation (2.6). In this case, it will only form a convex term in the MARS approximation if its coefficient is nonnegative. In the case of a pair of univariate basis functions, the coefficients of a pair are considered together. For example, the top two plots in Figure 3.1 display two forms in Equation (2.6) with $k = 0$. The lower left plot in Figure 3.1 shows the sum of a pair of univariate terms that yields a convex function while the lower right plot shows the sum of a pair that yields a concave function. The key is the sum of the coefficients for the pair. For the convex function, the two coefficients are 1.0 and −0.5, which sums to 0.5. However, for the concave function, the two coefficients are 1.0 and −1.5, which sums to −0.5. It can be seen that the critical value of the sum is zero, so a convex function can be guaranteed if that sum is nonnegative.

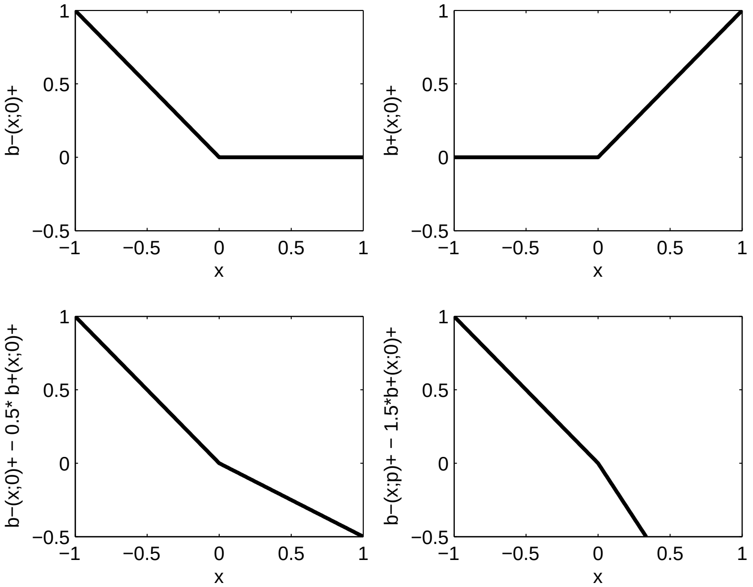


Figure 3.1. Pair of basis functions.

3.2 Form of Interaction Basis Functions

A nonconvex MARS approximation is inherently possible because interaction terms are products of univariate terms. In this case, a convex form of interaction basis functions for MARS must be properly developed, in order to ensure MARS convexity. In other words, not only the coefficient for the interaction basis function must be constrained, but also a new convex form is needed to successfully construct Convex MARS. In particular, original MARS utilizes a simple routine for smoothing each basis function to achieve continuous derivatives, and ideally the new convex interaction basis functions would utilize the same smoothing routine. Thus, the Convex-MARS versions in this dissertation are constructed so that the smoothing in Section 2.2.1 can be applied. The major difference is a transformation of the interaction basis functions must be performed to achieve a convex form prior to using the smoothing routine. This transformation depends on what form of convex interaction basis functions is used. One alternative for the transformation is rotation of the covariate axes, such that the direction as well as the magnitude for the interaction basis functions can be determined. This transforms the multiple covariates in an interaction to a new one-dimensional variable. In order to smooth the interaction basis functions, a cubic function, as in Friedman [25], or a quintic function, as in Chen et al. [14], can be applied to the one-dimensional variable in the same manner as it is applied to univariate basis functions. Different versions of this approach are developed for Convex-MARS-I and Convex-MARS-II in Sections 3.4.1 and 3.5.1, with further details on smoothing presented in Sections 3.4.5 and 3.5.6.

3.3 Convexity Proof

In this section, a convexity proof is provided for the pairs of MARS univariate terms in quintic form. Without loss of generality, a center knot of zero is specified. First, define $\Delta = k_+ - k_-$, $\Delta_1 = k_+ - k$, $\Delta_2 = k - k_-$, where k is the center knot, and k_+ and k_- are the side knots. Then the quintic functions are:

$$Q(x; s = +1, k = 0, k_+, k_-) =$$
$$\frac{[6\Delta_1 - 4\Delta_2][(x - k_-)^3]}{\Delta^3} + \frac{[-8\Delta_1 + 7\Delta_2][(x - k_-)^4]}{\Delta^4} + \frac{[3\Delta_1 - 3\Delta_2][(x - k_-)^5]}{\Delta^5},$$

and

$$-Q(x; s = -1, k = 0, k_+, k_-) =$$
$$\frac{[4\Delta_1 - 6\Delta_2][(x - k_+)^3]}{\Delta^3} + \frac{[8\Delta_2 - 7\Delta_1][(x - k_+)^4]}{\Delta^4} + \frac{[-3\Delta_1 + 3\Delta_2][(x - k_+)^5]}{\Delta^5}.$$

The goal is to prove the combined pair of univariate terms:

$$Q = \beta_1 Q(x; s = +1, k = 0, k_+, k_-) + \beta_2 Q(x; s = -1, k = 0, k_+, k_-) \tag{3.1}$$

is a convex function on $[k_-, k_+]$, where β_1 and β_2 are coefficients of basis functions. As illustrated in Figure 3.1, convexity appears assured if $\beta_1 + \beta_2 \geq 0$. Without loss of generality, we consider the case where $\beta_1 + \beta_2 \geq 0$, in which $\beta_1 > 0$ and $\beta_2 < 0$. Hence, the problem can be simplified as:

$$Q = \beta_1 [Q(x; s = +1, k = 0, k_+, k_-) + \frac{\beta_2}{\beta_1} Q(x; s = -1, k = 0, k_+, k_-)], \tag{3.2}$$

given $\beta_1 > 0$ and $\beta_2 < 0$. In particular, $\frac{\beta_2}{\beta_1} \geq -1$ is equivalent to $\beta_1 + \beta_2 \geq 0$. Considering the lower bound of $\frac{\beta_2}{\beta_1}$, this problem can be further simplified as

$$Q = Q(x; s = +1, k = 0, k_+, k_-) - Q(x; s = -1, k = 0, k_+, k_-). \tag{3.3}$$

To prove $Q = Q(x; s = +1, k = 0, k_+, k_-) - Q(x; s = -1, k = 0, k_+, k_-)$ is a convex function on $[k_-, k_+]$, we take the second derivative of Q with respect to x, and then we have the following equation:

$$Q'' = \frac{6[6\Delta_1 - 4\Delta_2][(x - k_-)]}{\Delta^3} + \frac{12[-8\Delta_1 + 7\Delta_2][(x - k_-)^2]}{\Delta^4} + \frac{20[3\Delta_1 - 3\Delta_2][(x - k_-)^3]}{\Delta^5} + \frac{6[4\Delta_1 - 6\Delta_2][(x - k_+)]}{\Delta^3} + \frac{12[-7\Delta_1 + 8\Delta_2][(x - k_+)^2]}{\Delta^4} + \frac{20[-3\Delta_1 + 3\Delta_2][(x - k_+)^3]}{\Delta^5},$$

$$\frac{\Delta_1}{\Delta} < \frac{2}{5} \quad \forall s = 1, \text{or } \frac{\Delta_2}{\Delta} < \frac{2}{5} \quad \forall s = -1. \tag{3.4}$$

Nonconvexities are produced in the cubic and quintic basis functions when the center knot k is not close enough to the midpoint between k_- and k_+ [14]. Given $\frac{\Delta_1}{\Delta} = \frac{2}{5}$ is used, it follows that $\frac{\Delta_2}{\Delta} = \frac{3}{5}$. Substitute those two values into Equation (3.4), which can be simplified as:

$$12\frac{x - k_-}{\Delta^3} - 12\frac{(x - k_-)^3}{\Delta^4} - 12\frac{(x - k_+)}{\Delta^2} + 24\frac{(x - k_+)^2}{\Delta^3} + 12\frac{(x - k_+)^3}{\Delta^4}. \tag{3.5}$$

Equation (3.5) can be decomposed as the sum of the following two components:

$$12/(\Delta^3)(x - k_-)^2[1 - \frac{x - k_-}{\Delta}] \tag{3.6}$$

and

$$12/(\Delta^2)(k_+ - x)[1 - 2\frac{x - k_+}{\Delta} - (\frac{x - k_+}{\Delta})^2]. \tag{3.7}$$

With $0 < \frac{x-k_-}{\Delta} < 1$, Equation (3.6) can be shown nonnegative. Similarly, with $0 < \frac{x-k_+}{\Delta} < 1$, Equation (3.7) can be proved nonnegative. Since, both Equation (3.6) and Equation (3.7) are nonnegative, Q is convex on $[k_-, k_+]$, given $\frac{2}{5} <= \frac{\Delta_1}{\Delta} <= \frac{3}{5}$.

3.4 Convex-MARS-I

3.4.1 Convex Form of Interaction Terms

To solve the problem of the inherently nonconvex interaction terms in the original MARS algorithm, the convex form of the m-th interaction basis function for Convex-MARS-I is proposed as follows:

$$B_m(\boldsymbol{x}) = [\sum_{l=1}^{L_m} \{s_{l,m} \cdot (x_{v(l,m)} - k_{l,m})/(1 - s_{l,m}k_{l,m})\}]_+ \ \ , \tag{3.8}$$

where the notation is the same as defined in Section 2.1.5. Consider the set of covariates $x_{v(l,m)}$ for an interaction term and corresponding knots $k_{l,m}$ and signs $s_{l,m}$, define

$$\omega_0(\boldsymbol{x}) = \sum_{l=1}^{L_m - 1} \{s_{l,m} \cdot (x_{v(l,m)} - k_{l,m})/(1 - s_{l,m}k_{l,m})\} \ \ , \tag{3.9}$$

$$\omega_1(\boldsymbol{x}; s_{L_m,m}) = s_{L_m,m} \cdot (x_{v(L_m,m)} - k_{L_m,m})/(1 - s_{L_m,m}\ k_{L_m,m}) \ \ . \tag{3.10}$$

In fact, the parent term defined in Equation (3.9) and split term defined in Equation (3.10) can be expressed as the following equations:

$$\omega_0(\boldsymbol{x}) = a_0 + \sum_{l=1}^{L_m - 1} a_l\ x_{v(l,m)} \ \ , \tag{3.11}$$

where

$$a_0 = k_{l,m} s_{l,m} / (s_{l,m} k_{l,m} - 1) \;\; ; \;\; a_l = s_{l,m} / (1 - s_{l,m} k_{l,m}) \;\; , \tag{3.12}$$

and

$$\omega_1(\boldsymbol{x}; s_{L_m,m}) = s_{L_m,m} k_{L_m,m} / (s_{L_m,m}\, k_{L_m,m} - 1) + s_{L_m,m} / (1 - s_{L_m,m} k_{L_m,m}) \cdot x_{v(L_m,m)} \;\; . \tag{3.13}$$

The covariates for the interaction are transformed into a new corresponding one-dimensional variable:

$$z^-(\boldsymbol{x}) = \omega_0(\boldsymbol{x}) + \omega_1(\boldsymbol{x}; s_{L_m,m} = -1) \;\; ; \;\; z^+(\boldsymbol{x}) = \omega_0(\boldsymbol{x}) + \omega_1(\boldsymbol{x}; s_{L_m,m} = +1) \;\; . \tag{3.14}$$

The two candidate basis functions of interactons for C-MARS-I are as follows:

$$b^+(z^-; \tau) = [(z^- - \tau)]_+ \;\; ; \;\; b^+(z^+; \tau) = [(z^+ - \tau)]_+ \;\; , \tag{3.15}$$

where the multivariate knot $\boldsymbol{k}$ in the original covariates $\boldsymbol{x}$ is transformed to the knot $\tau = 0$ in the dimension of z^- or z^+. Plugging in $\omega_0(\boldsymbol{x})$ and $\omega_1(\boldsymbol{x}; s_{L_m,m})$, the general form of a Convex-MARS-I interaction basis function is:

$$B_m(\boldsymbol{x}) = [\{\omega_0(\boldsymbol{x}) + \omega_1(\boldsymbol{x})\}]_+ \;\; . \tag{3.16}$$

The convex interaction transformation algorithm I (CIT-I) that conducts this transformation is shown in Algorithm 1.

Figure 3.2 illustrates two-way interaction terms for both original MARS and Convex-MARS. The nonconvexity of the interaction terms of original MARS is clearly visible while our proposed modification eliminates this issue. Finally, in addition to modifying the form of the interaction term, its coefficient must be constrained to be nonnegative because a negative coefficient would yield a concave interaction function.

Algorithm 1 Convex Interaction Transformation Algorithm I (CIT-I)

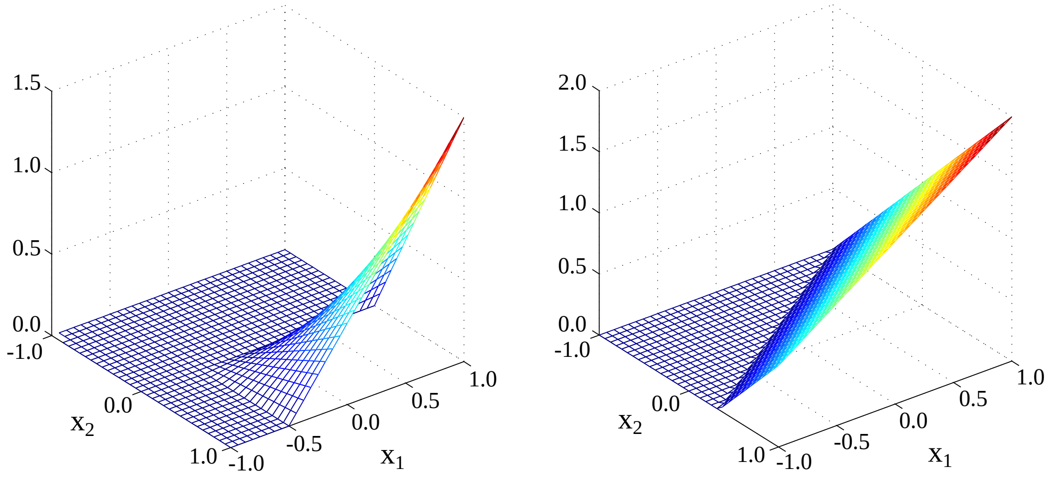
```
z = 0.
for all ($l$=1,2, …,$L_m$) do
    if $s_{l,m}$==1 then
        z += $(x_{v(l,m)} - k_{l,m})/(1 - k_{l,m})$.
    else if $s_{l,m}$==−1 then
        z −= $(x_{v(l,m)} - k_{l,m})/(1 + k_{l,m})$.
    end if
end for
return z.
```

3.4.2 Extension of Convexity Proof

In Convex-MARS-I, the multiple covariates of an interaction are reduced to the one-dimensional variable in Equation (3.14). The proof of convexity of a Convex-MARS-I interaction term in quintic form is similar to the convexity proof in Section 3.3, except only a single interaction term is considered instead of a pair of interaction terms.

3.4.3 Forward Coefficient Restriction Algorithm I

One could develop a full optimization search to identify the set of convex functions to form a convex MARS approximation. However, as with original MARS, we chose to employ stepwise procedures to conduct a more efficient, although suboptimal, search. In Convex-MARS, the forward stepwise procedure of original MARS is modified to check the coefficients of newly added basis functions according to the criteria described in Section 3.1 and Section 3.2. Nevertheless, because the basis functions are not necessarily orthogonal, the coefficients can change during forward selection. In theory, if the true underlying function is convex, then basis functions that introduce potential nonconvexity



(a) (b)

Figure 3.2. Comparison of MARS interaction basis functions: (a) Original MARS, (b) Convex-MARS.

should not be needed. Forward coefficient restriction algorithm I (FCR-I, Algorithm 3) searches for possible candidate basis functions. In essence, Algorithm 3 (FCR-I) is modified from Algorithm 2 (Friedman [25]). Algorithm 3 (FCR-I) constrains the coefficients for basis functions throughout the searching process. Whenever there are basis functions being added to the current set of basis functions, either a univariate pair or one basis function (an interaction term or one univariate term) is possible. In the first case, the sum of the two coefficients are constrained to be nonnegative if they come from a univariate pair. In the latter case, the coefficient is restricted to be nonnegative if it comes from a unpaired basis function.

3.4.4 Backward Iteration of Pruning and Refitting Algorithm I

For computational reasons, the coefficients for the previously added basis functions are not checked in Algorithm 3 (FCR-I). Since the convex form of the basis functions are not necessarily orthogonal, the coefficients might change each time any basis function is added in the model. Thus, it becomes necessary to re-check the coefficients at the end.

Algorithm 2 Friedman's MARS forward algorithm

```
Initialize $M = 1$; maxIA = maximum # covariates in an interation.
while $(m < M_{\max})$ do
  LOF $= \infty$.
  for all $m = 0, \ldots, M - 1$ do
    if basis function $m$ involves fewer than maxIA covariates then
      for all $v = 1$ to $n$ do
        if $v \notin$ basis function $m$ then
          for all $k = 1$ to $K$ do
            if basis function $m$ is nonzero at $k$ then
              Split basis function at knot $k$ into 2 new basis functions.
              Calculate lack-of-fit LOF.
              if LOF < LOF* then
                LOF = LOF*; save $m*, v*, k*$.
              end if
            end if
          end for $k$
        end if
      end for $v$
    end if
  end for $m$
  Add basis functions$(m*, v*, k*)$; $M+ = 2$.
  Orthnormalize new basis functions.
end while
```

Algorithm 3 Forward Coefficient Restriction Algorithm I (FCR-I)

```
Initialize $M$=1; maxIA = maximum # covariates in an interation.
while ($m < M_{\max}$) do
  LOF* = $\infty$.
  for all $m = 0, \ldots, M-1$ do
    if basis function $m$ has fewer than maxIA splits then
      for all $v = 1$ to $n$ do
        if $v \notin$ basis function $m$ then
          for all $k = 1$ to $K$ do
            if basis function $m$ is nonzero at $k$ then
              Split basis function at knot $k$ into 2 new basis functions.
              if (nonnegative coefficient from a unpaired basis function) $\cup$ (nonnegative
              sum of coefficients from a univariate pair of basis functions) then
                Calculate lack-of-fit LOF.
                if LOF < LOF* then
                  LOF* = LOF; save $m*, v*, k*$.
                end if
              end if nonnegative
            end if basis
          end for $k$
        end if $v$
      end for $v$
    end if
  end for $m$
  Add basis functions($m*, v*, k*$); $M$+=2.
  Orthnormalize new basis functions.
end while
```

Backward iteration of pruning and refitting algorithm I (BIPR-I, Algorithm 4) re-checks the coefficients after running Algorithm 3 (FCR-I). Algorithm 4 (BIPR-I) searches for convexity violations and removes them. In the case of an interaction basis function, if its coefficient is negative, then the related basis function will be dropped. If any basis function is dropped, Convex-MARS-I has to be refit with the remaining basis functions.

Algorithm 4 Backward Iteration of Pruning and Refitting Algorithm I (BIPR-I)

```
Initialize the full set of $m$ basis functions.
while not convex do
  for all basis functions $\subseteq$ current set $(i = m, m-1, \ldots, 1)$ do
    if (negative coefficient) $\cap$ (not from a univariate pair) then
      Drop $i$-th basis function.
      $m = m - 1$.
    else if negative sum of coefficients for a univariate pair then
      Drop the pair of basis function ($i$-th and $i+1$-th ).
      $m = m - 2$.
    end if
    Refit Convex-MARS-I model if any basis functions have been dropped.
  end for
end while
```

Suppose Algorithm 3 (FCR-I) yields M nonconstant basis functions. A basis function is either by itself or one of a univariate pair. Starting from the last basis function that was added to Convex-MARS-I model, Algorithm 4 (BIPR-I) examines the coefficient of each basis function. If a basis function is not one of a univariate pair, then it is either an unpaired univariate basis function or an interaction basis function. Either way, the coefficient of such a basis function must be nonnegative, and if a negative coefficient is found, the basis function is dropped. If a basis function is one of a univariate pair, then the sum of the coefficients for the pair must be nonnegative. If a violation is found, then the pair of basis functions is dropped. Once a pass through all the basis functions is completed, and the violations have been dropped, the coefficients are refit for the remaining basis functions. This process repeats until no more violations are found. It is then that the approximation can be guaranteed to be convex.

3.4.5 Smoothing Routine (Degree of Continuity)

The smoothing routine for a univariate term in Convex-MARS-I is exactly the same as original MARS in Section 2.2.1. However, for the interaction terms, Algorithm 1 (CIT-I) transforms the covariates for an interaction term into a new corresponding one-dimensional variable z. In the dimension of variable z, the transformed center knot is always zero, the side knots can be set at κ and $-\kappa$, where κ is a smoothing factor. The larger the smoothing factor is, the smoother the approximation function will be. Typically, κ is between 0 and 1. One possible choice for κ is 0.5. Thus, variable z and side knots can be smoothed using the same smoothing routine as for a univariate basis function.

3.5 Convex-MARS-II

3.5.1 Convex Form of Interaction Terms

To solve the problem due to inherently nonconvex interaction terms in the original MARS algorithm and enhance the flexibility of Convex-MARS, a version of the convex form of the interaction basis function is developed that allows them to be added in *pairs*. Instead of requiring each interaction term to have a nonnegative coefficient, the coefficient restriction would be the same as univariate pairs, for which the sum of their coefficients for the pair must be nonnegative. As in Convex-MARS-I, the convex form of interaction basis functions transforms the multiple covariates in the interaction to a one-dimensional variable that is a linear combination of the covariates. The following uses the notation defined in Section 2.1.5. Given the set of covariates $x_{v(l,m)}$ for an interaction term and corresponding knots $k_{l,m}$ and signs $s_{l,m}$, define

$$\omega_0(\boldsymbol{x}) = \sum_{l=1}^{L_m-1} \{s_{l,m} \cdot (x_{v(l,m)} - k_{l,m})/(1 - s_{l,m}k_{l,m})\} \ , \tag{3.17}$$

$$\omega_1(\boldsymbol{x}; \phi_m) = \phi_m \cdot (x_{v(L_m,m)} - k_{L_m,m})/(1 - \phi_m \ k_{L_m,m}) \ . \tag{3.18}$$

where $\omega_0(\boldsymbol{x})$ represents the components of an existing basis function (parent term), $\omega_1(\boldsymbol{x}; \phi_m)$ represents the split component on variable $x_{v(L_m,m)}$ that creates a new interaction term. Sign ϕ_m (-1 or $+1$) determines two distinct one-dimensional variable directions:

$$z^-(\boldsymbol{x}) = \omega_0(\boldsymbol{x}) + \omega_1(\boldsymbol{x}; \phi_m = -1) \ ; \ z^+(\boldsymbol{x}) = \omega_0(\boldsymbol{x}) + \omega_1(\boldsymbol{x}; \phi_m = +1) \ . \tag{3.19}$$

To show that z^- and z^+ are linear combinations of the covariates, re-write:

$$\omega_0(\boldsymbol{x}) = a_0 + \sum_{l=1}^{L_m-1} a_l \ x_{v(l,m)} \ , \tag{3.20}$$

where

$$a_0 = k_{l,m}s_{l,m}/(s_{l,m}k_{l,m} - 1) \ ; \ a_l = s_{l,m}/(1 - s_{l,m}k_{l,m}) \ , \tag{3.21}$$

and

$$\omega_1(\boldsymbol{x}; \phi_m) = \phi_m k_{L_m,m}/(\phi_m \ k_{L_m,m} - 1) + \phi_m/(1 - \phi_m k_{L_m,m}) \cdot x_{v(L_m,m)} \ . \tag{3.22}$$

The <u>c</u>onvex <u>i</u>nteraction <u>t</u>ransformation algorithm II (CIT-II) that conducts this transformation in Equation (3.19) is shown in Algorithm 5. The major difference between Algorithm 1 (CIT-I) and Algorithm 5 (CIT-II) is that the latter implements a more comprehensive search. In essence, ϕ_m enhances the flexibility of the search loop.

Algorithm 5 Convex Interaction Transformation Algorithm II (CIT-II)

```
z = ($\phi_m$ / (1$-\phi_m$ * $k_{L_m,m}$) * ($x_{v(L_m,m)} - k_{L_m,m}$)).
for all ($l = 1, 2, \ldots, L_m - 1$) do
    if $s_{l,m}$==1 then
        z += ($x_{v(l,m)} - k_{l,m}$)/(1$-k_{l,m}$).
    else if $s_{l,m}$==$-1$ then
        z $-$= ($x_{v(l,m)} - k_{l,m}$)/(1+$k_{l,m}$).
    end if
end for
return z.
```

The m-th pair of interaction basis functions (i.e. m-th basis function with $s_{L_m,m} = -1$, $m+1$-th basis function with $s_{L_m,m} = 1$) can use either z^- or z^+. Thus, the two candidate pairs of interaction basis functions for Convex-MARS-II are as follows:

$$b^-(z^-;\tau) = [-(z^- - \tau)]_+ \;\; ; \;\; b^+(z^-;\tau) = [+(z^- - \tau)]_+ \;\; , \tag{3.23}$$

or

$$b^-(z^+;\tau) = [-(z^+ - \tau)]_+ \;\; ; \;\; b^+(z^+;\tau) = [+(z^+ - \tau)]_+ \;\; . \tag{3.24}$$

where the multivariate knot $\boldsymbol{k}$ in the original covariates $\boldsymbol{x}$ is transformed to the knot $\tau = 0$ in the dimension of z^- or z^+. Both pairs are considered in the forward algorithm search loop to enable better flexibility. Plugging in $\omega_0(\boldsymbol{x})$ and $\omega_1(\boldsymbol{x};\phi_m)$, the general form of a pair of Convex-MARS-II interaction basis functions is:

$$B_m(\boldsymbol{x}) = [-1\{\omega_0(\boldsymbol{x}) + \omega_1(\boldsymbol{x})\}]_+ \;\; ; \;\; B_{m+1}(\boldsymbol{x}) = [+1\{\omega_0(\boldsymbol{x}) + \omega_1(\boldsymbol{x})\}]_+ \;\; . \tag{3.25}$$

To better understand the role of ϕ_m, consider the two-way interaction example shown in Figure 3.3. In Figure 3.3, knots for the two covariates are 0.25 and -0.5 for covariates x_1 and x_2, respectively, and $s_{1,m} = -1$. The upper two contour plots demonstrate the pair of two-way interaction basis functions with $\phi_m = -1$, and the lower two contour plots show the pair of two-way interaction basis functions with $\phi_m = 1$. The upper left plot shows the case when $s_{2,m} = 1$, and the upper right plot displays the case when $s_{2,m} = -1$.

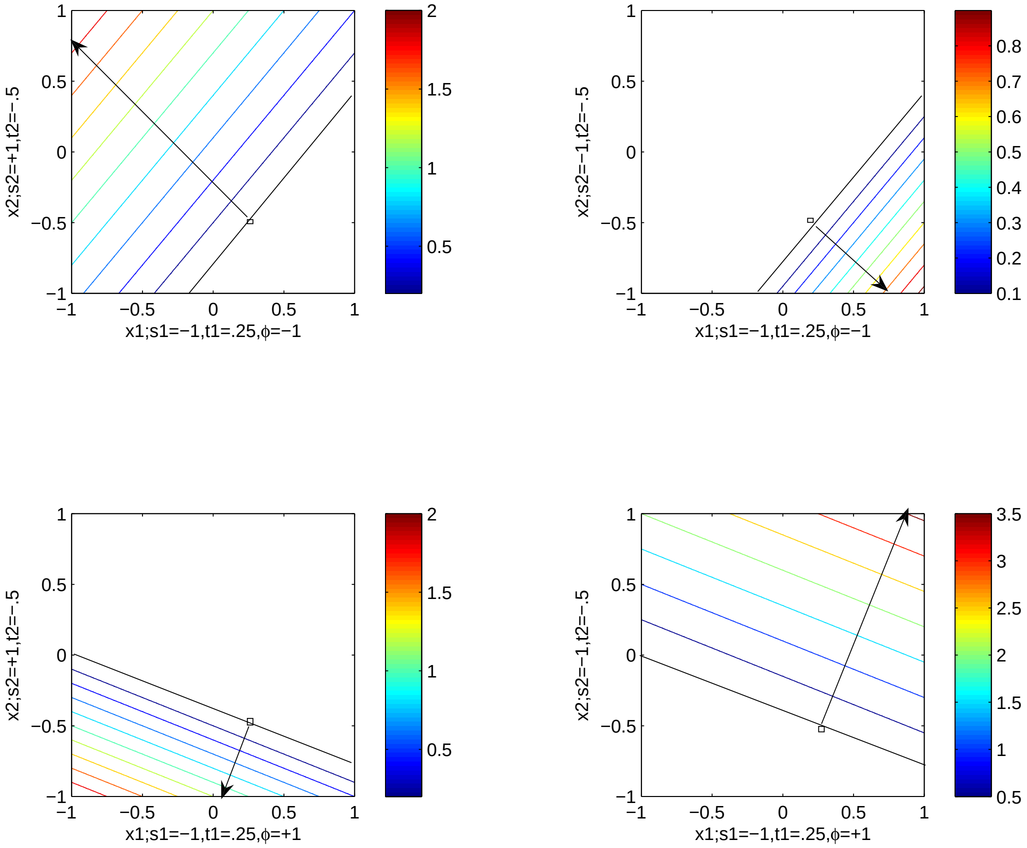


Figure 3.3. Comparison of candidate pairs of Convex-MARS-II interaction basis functions.

3.5.2 Extension of Convexity Proof

In Convex-MARS-II, a pair of multi-covariate interaction basis functions is transformed to a pair of univariate basis functions, as in Equation (3.23) or Equation (3.24). The proof of convexity of a pair of Convex-MARS-II interaction terms in quintic form is identical to the convexity proof in Section 3.3, using the transformed variables z^- or z^+.

3.5.3 Forward Coefficient Restriction Algorithm II

For Convex-MARS-I, interaction basis functions are not added in pairs in Algorithm 3 (FCR-I). This might potentially limit the flexibility of a Convex MARS approximation. In order to improve the fit of Convex MARS-I via the Convex-MARS-II interaction basis functions, Algorithm 3 (FCR-I) must be modified to incorporate a more flexible search for eligible pairs of basis functions. Such flexibility could be achieved through incorporating ϕ_m, in Equation (3.19), into the search loop for interaction terms. Similar to Convex-MARS-I, the sum of the coefficients is restricted to be nonnegative in the case of a pair of basis functions. For Algorithm 3 (FCR-I), only univariate basis functions are considered as pairs; however, for forward coefficient restriction algorithm II (FCR-II, Algorithm 6), all basis functions can be pairs when considering the coefficient criteria.

3.5.4 Backward Iteration of Pruning and Refitting Algorithm II

Similar to Algorithm 4 (BIPR-I), backward iteration pruning and reffiting algorithm II (BIPR-II, Algorithm 7) searches for convexity violations and removes them. In Algorithm 7, one of the pair of basis functions will be dropped if the sum of their coefficients is negative. Specifically, the dropped basis function has the smaller coefficient of the pair. However, if both coefficients corresponding to respective basis functions are

Algorithm 6 Forward Coefficient Restriction Algorithm II (FCR-II)

```
Initialize $M = 1$; maxIA = maximum # covariates in an interation.
while ($m < M_{\max}$) do
  LOF* = $\infty$.
  for all $m = 0, \ldots, M - 1$ do
    if basis function $m$ involves fewer than maxIA covariates then
      for all $v = 1$ to $n$ do
        if $v \notin$ basis function $m$ then
          for all $k = 1$ to $K$ do
            for all $\phi_m = -1$ , $+1$ do
              if basis function $m$ is nonzero at $k$ then
                Split basis function at knot $k$ into 2 new basis functions.
                if (nonnegative coefficient from a unpaired basis function) $\cup$ (nonnegative
                sum of coefficients from a pair of basis functions) then
                  Calculate lack-of-fit LOF.
                  if LOF < LOF* then
                    LOF* = LOF; save $m*, v*, k*, \phi_m*$.
                  end if
                end if nonnegative
              end if basis function $m$
            end for $\phi_m$
          end for $k$
        end if $v$
      end for $v$
    end if
  end for $m$
  Add basis functions($m*, v*, k*, \phi_m*$); $M$+=2.
  Orthnormalize new basis functions.
end while
```

negative, this pair will be dropped completely. Further, if any basis function is dropped, Convex-MARS-II has to be refit with the remaining basis functions. Once a pass through all the basis functions is completed, and the violations have been dropped, the coefficients are refit for the remaining basis functions. This process repeats until no more violations are found. It is then that the approximation can be guaranteed to be convex.

Algorithm 7 Backward Iteration of Pruning and Refitting Algorithm II (BIPR-II)

```
Initialize the full set of $m$ basis functions.
while not convex do
  for all basis functions $\subseteq$ current set $(i = m, m-1, \dots, 1)$ do
    if negative coefficient $\cap$ unpaired then
      Drop $i$-th basis function.
      $m = m - 1$.
    else if  negative sum of coefficients for a pair then
      Drop one of the pair of basis functions ($i$-th and $i+1$-th ).
      $m = m - 1$.
    else if  negative coefficients for each of a pair then
      Drop the pair of basis functions ($i$-th and $i+1$-th ).
      $m = m - 2$.
    end if
    Refit Convex-MARS-II model if any basis functions have been dropped.
  end for
end while
```

3.5.5 Forward Coefficient Restriction Algorithm II-T

As we mentioned in Section 3.4.4, the coefficients for previously added basis functions are not checked in the FCR forward algorithms (Algorithms 3 or 6), potentially permitting coefficient violations. Although, the BIPR backward algorithms (Algorithms 4 or 7) re-check the coefficient criteria at the end of the FCR forward algorithms, many basis functions that violate the coefficient criteria may be dropped. It is possible to re-run the FCR forward algorithm to add more basis functions, but this was not found to improve the fit. Instead, the coefficient criteria are modified using a stricter *threshold*. In particular, a slightly modified algorithm is developed, called the forward coefficient restriction algorithm II-T (FCR-II-T, Algorithm 8). Essentially, the eligible pairs of basis functions from Algorithm 8 (FCR-II-T) is a subset of that from Algorithm 6 (FCR-II). The difference between Algorithm 6 (FCR-II) and Algorithm 8 (FCR-II-T) is that the latter contains a stricter convex subset of basis functions. With the stricter threshold, fewer or possibly no coefficients violate the coefficient criteria at the end of Algorithm 6 (FCR-II).

The major issue is how to build the coefficient criteria to select a subset from Algorithm 6 (FCR-II). For Algorithm 6 (FCR-II), the sum of the coefficients for a pair of basis functions is constrained to be nonnegative. To find a subset from Algorithm 6 (FCR-II), the *threshold* for the coefficient criteria is now a nonnegative number, instead of just zero. The challenge, however, lies in selecting the *threshold*, i.e. how to decide the positive number, or equivalently, the *threshold*. One appropriate approach is to run a multiple linear regression (MLR) on the data set prior to determining the *threshold*. The maximum absolute coefficient can be obtained by taking the absolute values for the coefficients from the MLR model accordingly. Alternatively, if MARS model is already available, the maximum absolute coefficient can be obtained from its result. In general, the rule of thumb is $2 \sim 20\%$ of the absolute value of the maximum coefficient could be a reasonable candidate for the *threshold*. If the *threshold* is set too high, then this can reduce flexibility since fewer basis functions are considered eligible in the searching process. If the selected *threshold* is too small, then will be little difference between FCR-II (Algorithm 6) and FCR-II-T (Algorithm 8).

3.5.6 Smoothing Routine (Degree of Continuity)

As with Convex-MARS-I, the smoothing routine for a univariate term in Convex-MARS-II is exactly the same as original MARS. For the interaction terms, Algorithm 5 (CIT-II) transforms the covariates for an interaction term into a new corresponding one-dimensional variable z^- or z^+. Since the transformed center knot is always zero for the interaction terms of Convex-MARS-II, similar to Algorithm CIT-I in Section 3.4.5, the corresponding side knots can be set at κ and $-\kappa$, where κ is the smoothing factor. The larger the smoothing factor is, the smoother the approximation function will be. One possible choice for κ is 0.5. Therefore, the transformed basis function in the one-dimensional variable z^- or z^+, given knots and the sign $s_{L_m,m}$, can be smoothed using the same smoothing routine as for a univariate basis function.

3.6 Inventory Forecasting Problem

In this section, Convex-MARS is tested on four-dimensional and nine-dimensional inventory forecasting SDP problems studied by Chen et al. [18]. The goal of the inventory forecasting problem is to minimize the inventory holding and backorder costs. The optimal value function, known to be theoretically convex, specifies the minimum expected cost to operate the system. The state of the system is represented by the inventory levels for the products and their demand forecasts. The versions of MARS are fit to the data for the last time period of the three-period inventory forecasting SDP.

Algorithm 8 Forward Coefficient Restriction Algorithm with Threshold (FCR-II-T)

```
Initialize $M = 1$; maxIA = maximum # covariates in an interation.
while ($m < M_{\max}$) do
  LOF* = $\infty$.
  for all $m$ =0,...,$M - 1$ do
    if basis function $m$ involves fewer than maxIA covariates then
      for all $v = 1$ to $n$ do
        if $v \notin$ basis function $m$ then
          for all $k = 1$ to $K$ do
            if basis function $m$ is nonzero at $k$ then
              for all $\phi_m = -1$ , $+1$ do
                Split basis function at knot $k$ into 2 new basis functions.
                if (nonnegative coefficient from a unpaired basis function) $\cup$ (nonnegative sum of coefficients from a pair of basis functions) then
                  Calculate lack-of-fit LOF.
                  if LOF < LOF* then
                    LOF* = LOF; save $m*$, $v*$, $k*$, $\phi_m*$.
                  end if
                end if nonnegative
              end for $\phi_m$
            end if basis function $m$
          end for $k$
        end if $v$
      end for $v$
    end if
  end for $m$
  Add basis functions($m*, v*, k*, \phi_m*$); $M$+=2.
  Orthnormalize new basis functions.
end while
```

3.6.1 Four-dimensional Inventory Forecasting Problem

In this inventory forecasting DP, there are two products, each with one demand forecast. MARS was fit to a data set of 125 points from the last period. A set of 100 validation data points is used to compare the three different models, and boxplots of the absolute errors, computed using the formula $|y - \hat{f}|$, are shown in Figure 3.4, where y is the actual cost of the system, and $\hat{f}$ is the MARS/Convex-MARS approximation for the actual cost. A boxplot of the actual costs of the inventory model, for each of the 100 validation points, is displayed in Figure 3.5 for reference. The performance of both versions of Convex-MARS are comparable to original MARS. Further, because versions of Convex-MARS guarantee convexity, optimization methods that depend on this assumption can be reliably employed.

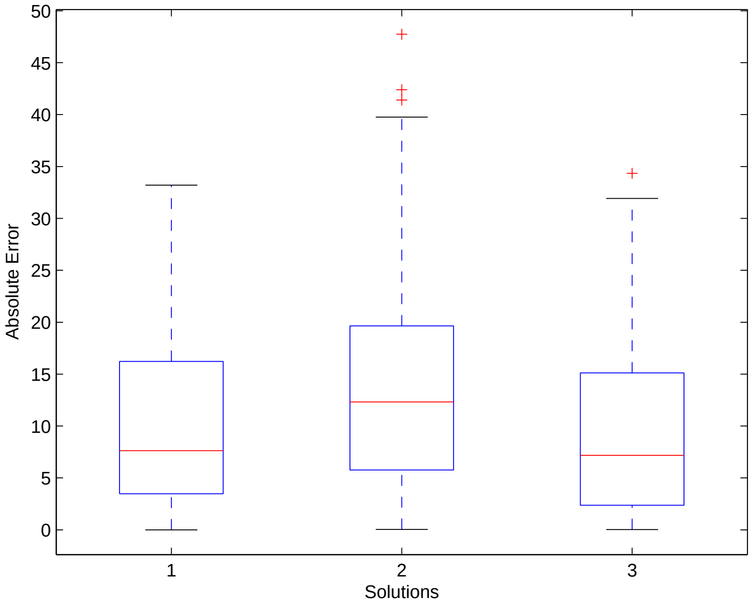


Figure 3.4. Comparison of boxplots (four-dimensional inventory forecasting problem) based on a validation set of 100 points: (1) Origianl MARS, (2) Convex-MARS-I, (3) Convex-MARS-II-T.

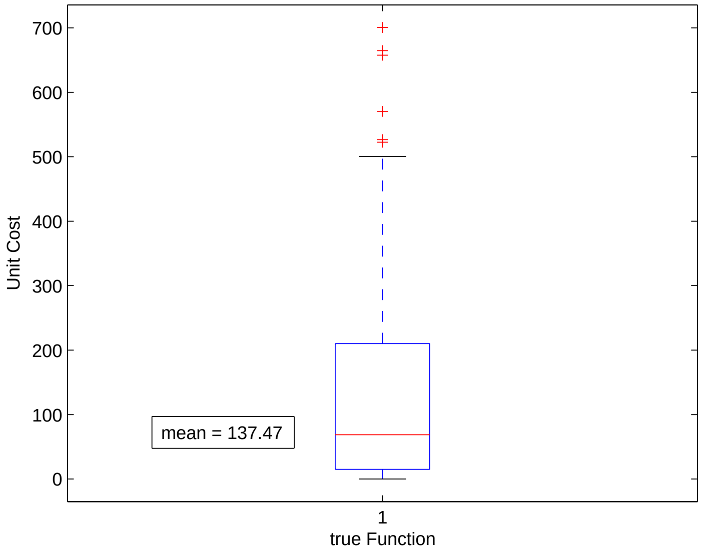


Figure 3.5. Boxplot of the actual cost for the four-dimensional inventory forecasting problem.

Table 3.1 summarizes the parameter settings for original MARS, Convex-MARS-I and Convex-MARS-II-T. In particular, the threshold for Convex MARS-II-T (Convex version of MARS with coefficient *threshold* constraint), 23.34, is determined by multiplying 5.0% by the absolute value of maximum coefficient, 490.729, in original MARS model. The MARS algorithms were written in C on and executed on a Dual 2.6-GHz Athlon Workstation. In terms of CPU time, all three versions of MARS requires low computational effort (Table 3.2). However, convexity must be assured to obtain

Table 3.1. Setting of algorithms on the three versions of MARS based on the four-dimensional inventory forecasting problem

	Original MARS	Convex-MARS-I	Convex-MARS-II-T
Convex: threshold	N/A	N/A	23.34
ASR :	N/A	ASR-I:diff .002 ,	ASR-I:diff .002
Common Setting:	points: 125,	konts:3, int.:3 ,	M_{max}:100.

the global optimum for the four-dimensional inventory forecasting DP problem. In the four-dimensional inventory forecasting problem, versions of Convex-MARS guarantee the convexity of their approximations.

Table 3.2. Comparison of CPU time for each version of MARS based on the four-dimensional inventory forecasting problem

original MARS	Convex-MARS-I	Convex-MARS-II-T
.074 sec	.521 sec	.028 sec

3.6.2 Nine-dimensional Inventory Forecasting Problem

In this inventory forecasting DP, there are three products, each with two demand forecasts (for next time period and the one after). MARS was fit to a data set of 1331 points from the last period. A set of 1000 validation data points is used to compare the three different models, and boxplots of the absolute errors, computed using the formula $|y - \hat{f}|$, are shown in Figure 3.4, where y is the actual cost of the system, and $\hat{f}$ is the MARS/Convex-MARS approximation for the actual cost. A boxplot of the actual cost of the inventory model, for each of the 1000 validation points, is displayed in Figure 3.5 for reference. The performance of both versions of Convex-MARS are comparable to ASR-MARS. Further, because versions of Convex-MARS guarantee convexity, optimization methods that depend on this assumption can be reliably employed.

Table 3.3 summarizes the parameter settings for ASR-MARS, Convex-MARS-I and Convex-MARS-II. The MARS algorithms were written in C on and executed on a Dual 2.6-GHz Athlon Workstation. In terms of CPU time, all three versions of MARS requires low computational effort (Table 3.4). However, convexity must be assured to obtain the global optimum for the nine-dimensional inventory forecasting DP problem. In

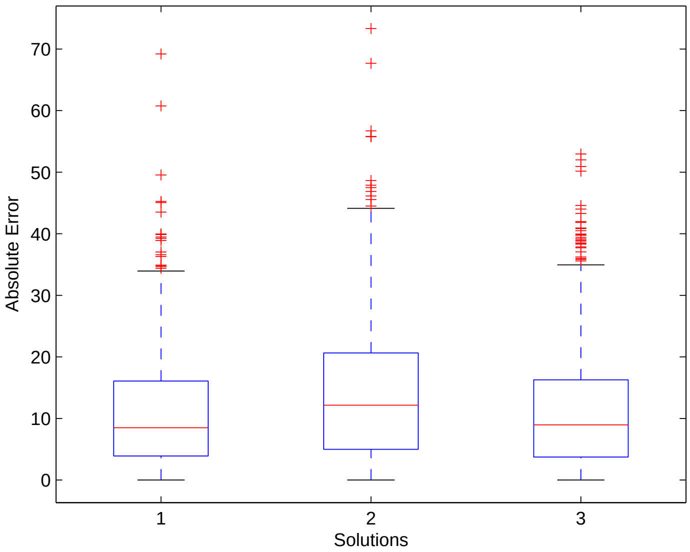


Figure 3.6. Comparison of boxplots (nine-dimensional inventory forecasting problem) based on a validation set of 1000 points: (1) ASR-MARS, (2) Convex-MARS-I, (3) Convex-MARS-II.

Table 3.3. Setting of algorithms on the three versions of MARS based on the nine-dimensional inventory forecasting problem

	ASR-MARS	Convex-MARS-I	Convex-MARS-II
Robust :	Robust-I: tolerance: .3	N/A	N/A
Common Setting :	ASR-II: diff .02 ,	interaction:3,	
	points: 1,331,	konts:45,	M_{max}:300.

the nine-dimensional inventory forecasting problem, versions of Convex-MARS guarantee the convexity of their approximations.

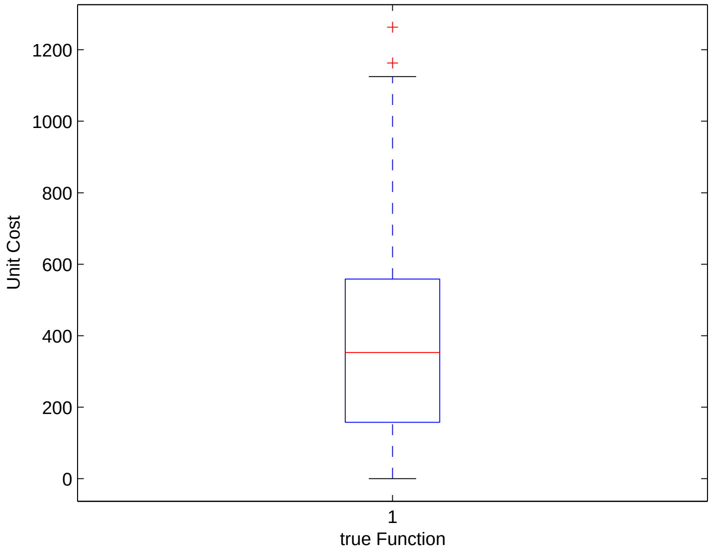


Figure 3.7. Boxplot of the actual cost for the nine-dimensional inventory forecasting problem.

Table 3.4. Comparison of CPU time for each version of MARS based on the nine-dimensional inventory forecasting problem

ASR-MARS	Convex-MARS-I	Convex-MARS-II
1.746 sec	7.228 sec	3.858 sec

CHAPTER 4

DATA MINING VARIABLE SELECTION FOR OPTIMIZATION

This chapter focuses on variable selection methods, particularly for the "large p and small n" problem. A two-stage SP airline fleet assignment model (FAM) studied by Pilla [41] is presented as an application. The next section briefly describes SP-FAM. Section 4.2 illustrates the data mining process to reduce a high-dimensional decision variable space, and Section 4.3 presents computational results on SP-FAM.

4.1 Airline Fleet Assignment Problem

The FAM studied by Pilla [41] uses a two-stage SP framework along with the Boeing concept of D^3 to swap crew-compatible aircraft closer to departure, when most of the demand has been realized. Crew-compatible aircraft have identical cockpits, allowing an airline to swap aircraft without swapping crews. The two-stage formulation assigns crew-compatible aircraft in the first stage, about 90 days prior to departure, so as to enhance the demand-capturing potential of swapping in the second stage, about two weeks prior. The stochasticity of the demand is modeled by different demand scenarios in the second stage, and the average over the scenarios estimates the expected profit, which is the performance measure that will be used as the response variable y in the variable selection methods. The explanatory variables $\boldsymbol{x}$ are the crew-compatible allocation (CCA) decision variables that specify which crew-compatible family is assigned to each flight leg. The expected profit function is known to be concave over the CCA space.

Traditional two-stage SP uses a Benders' approach or L-shaped method [6]; however, for large-scale problems, this can be slow to converge. Pilla [41] developed a two-phase DACE approach to reduce the computation involved in conducting the optimization. The DACE phase uses first-stage constraints in a multi-step process to construct an experimental design within the feasible region, then builds a statistical model that approximates the expected profit function in the first stage of the SP. The optimization phase solves the two-stage problem using the DACE expected profit approximation instead of solving many second-stage subproblems in every iteration. This greatly speeds up the optimization, compared to Benders', because the computation of the subproblems is shifted to the DACE phase. However, further speedup may be achieved by conducting variable selection prior to the DACE phase.

For a real airline network with 50 stations and 2358 legs, Pilla's DACE phase reduced the decision space from 6537 to 1264 dimensions, and his multi-step process derived 141 initial feasible CCA region. The second-stage subproblem is then solved for each of these design points. Among the 1264 decision variables, there are still many useless ones that could be identified via variable selection, enabling a much smaller set of design points. In particular, in this dissertation variable selection is conducted for the 1264 CCA variables using the initial 141 initial feasible CCA points; clearly, this is a "large p and small n" problem.

4.2 Data Mining

The implementation of MARS for approximating optimization functions in complex optimization problems can involve thousands of variables. Even though one can simply attempt a MARS approximation over all these variables, many may have little impact on the desired performance objective. Therefore, a data mining step for variable selection is required to adequately exclude noninformative variables.

For the airline SP-FAM problem in this dissertation, we have $p = 1,264$ and $n = 141$. In order to address this "large p and small n " problem, we used a multiple testing procedure based on the FDR and regression trees. The background on CART and FDR have been introduced in Sections 2.1.3 and 2.3. The usage of CART for variable selection is described in Section 4.2.1. With regard to FDR, the tricky issue is that traditional FDR assumes a categorical response variable and numerical explanatory variables. In DACE, the response and explanatory variables are both numerical, and typically continuously-valued. Section 4.2.2 describes a new FDR based method variable selection that uses CART to categorize a continuous response, so that a traditional FDR procedure can be used for multiple hypothesis testing. Section 4.2.3 describes a new FDR based method called Inverse FDR (InvFDR) that maintains the continuous characteristic of response variable. Thus, three variable selection methods are discussed in this dissertation:

1. Variable importance scores from CART.
2. FDR-based variable selection based on regression trees.
3. InvFDR: FDR-based variable selection grouped by predictors.

4.2.1 Variable Importance Scores from CART

CART developed by Breiman et al. [11] is a very popular data mining tool for supervised learning. The CART forward algorithm uses binary recursive partitioning to separate the variable space into rectangular regions based on similarity of the response values. In this dissertation, we utilized CART software from Salford Systems (www.salfordsystems.com). For variable selection, this software provides "relative variable importance scores." The variable that receives a 100 score indicates the most influential variable for prediction, followed by other variables based on their relative importance to the most important one. However, there are some different options for calculating the scores, and selecting the threshold of the scores may be subjective. Further, this method tends to select an overly small number of variables. This motivates the development of an objective and systematic approach for variable selection.

4.2.2 FDR-based Variable Selection with CART (FDR/CART)

Generally, a conventional FDR procedure for variable selection requires a categorical response variable that separates the data into C groups, where C is the number of categories. For each predictor variable, we test for differences in the C samples, using a two-sample t-test or F-test. However, because the response variable generated by computer experiments is continuous in most cases, we need to categorize the original response. A mean or median value of the response variable can be used to separate the response variable into two groups, high and low, if the response surface is monotonic. In an air quality application addressed in Shih et al. [46], the effectiveness of this simple grouping strategy was shown. However, if the relationship between the response and the predictors is not monotonic, such that the separation by high and low values does not make sense, then alternate grouping strategies are needed. In order to address this problem, we use binary regression trees (e.g., CART) to partition the response observations into meaningful groups. An algorithm constructing binary regression trees partitions the space into two regions using the predictor variable and splitting-point that achieves the best improvement in fit. This partitioning process is repeated on one or both of these regions until a termination criterion has been reached. Based on the terminal nodes of regression trees, the response values can be separated into a certain number of groups, and an FDR procedure can be applied for variable selection. Note that for three or more groups, an analysis of variance (ANOVA) table is constructed for each predictor variable and its significance is tested using an F-test. This approach simultaneously takes advantage of both regression trees and FDR procedure.

4.2.3 Inverse FDR (InvFDR)

In order to maintain the continuous characteristic of a response variable in an FDR procedure, InvFDR uses the groupings on the explantory variables instead of on the responses. The main purpose is to create a set of new variables possessing the same number of original predictors by grouping the response variable based on the predictors and conducting an FDR procedure on these new variables. This is analogous to the resampling technique because each new variable is re-sampled from the original response based on each predictor variable. InvFDR is similar to the original FDR procedure, except that the hypothesis test is conducted on the continuous response grouped by each predictor variable, as opposed to testing each continuous predictor variable grouped by the response values. The setting and procedure for InvFDR is as follows:

- For each predictor variable, divide the response variable into C groups based on the values of the predictor variable.
- For each predictor variable, conduct a statistical test (e.g., two-sample t-test, ANOVA F-test) on its corresponding set of response variable groups, and record the p-value. There will be one p-value for each predictor variable.
- Use the p-values to conduct an FDR procedure that identifies which predictor variables are statistically significant.

If the response surface is known to be convex or concave, a common occurrence in optimization, then InvFDR with $C = 3$ groups should be sufficient. In general, more complex nonlinear structure can be captured with more groups.

4.3 Computational Results on SP-FAM Application

In this section, the results for the implementation of the three variable selection approaches on SP-FAM [41] are illustrated.

4.3.1 Variable Selection for SP-FAM

The variable selection procedures of all three approaches were implemented on a data set of 141 feasible CCA points from the airline fleet assignment problem sudied by Pilla [41]. It should be noted that InvFDR is especially appropriate for SP-FAM becuase the CCA decision variables $\boldsymbol{x}$ are binary $\{0, 1\}$ in their pure form, although fractional values can exist. The settings and procedures for CART Variable Importance Scores, FDR/CART, and InvFDR are described as follows:

1. Setting for CART Variable Importance Scores: Run CART and select all of those variables with nonzero variable importance scores.
2. Setting and procedure for FDR/CART: Based on the terminal nodes of CART, the response variable can be grouped into three groups. FDR (family $\alpha = 0.0000001$) is implemented based on the three groups.
3. Setting and procedure for InvFDR:
   (a) For each x_v-variable: Divide the y-values into three groups corresponding to x_v-values of 0, fractional, and 1.
   (b) For each x_v-variable: Conduct an ANOVA F-test on the y-values for these three groups, and record the p-values.
   (c) Conduct the FDR procedure controlling FDR at 0.05.

For the fleet assignment problem, Table 4.1 shows the results of the three variable selection approaches discussed in this dissertation. Specifically, CART variable importance scores, FDR/CART, and InvFDR selected 36, 565, and 476 variables, respectively. In other words, variable reductions of 97.15%, 55.30%, and 62.34% were achieved by the three methods.

Table 4.1.Results of the three variable selection approaches for the FAM problem

	Description	number of variable selected
CART	Variable Importance Scores from CART	36
FDR/CART	FDR with $C = 3$ Groups from CART	565
InvFDR	Inverse FDR	476

The above approaches have been applied on a fleet assignment problem with 1,264 decision variables and 141 feasible CCA points. It should be noted that 1,264 variables were reduced to 1,061 prior to implementation of the FDR-based variable selection grouped by predictors. Those 203 (= 1264 − 1061) were dropped because they possess uniform values for almost all 141 observations. In order to understand the covariance structure among the 1,061 variables, the pairwise correlations must be obtained. The $C_2^{1061} = 562,330$ pairwise correlations for the $1,061$ predictors variables are computed, and Fig 4.1 shows the histogram of the pairwise correlation coefficients of those variables. Among them, about $9,988$ pairwise correlations have very strong correlation ($|\rho| > 0.8$), and $31,418$ pairwise correlations have strong correlation ($|\rho| > 0.6$). Or equivalently, about 1.78% of C_2^{1061} pairwise correlations have very strong correlation ($|\rho| > 0.8$), and 5.59% of C_2^{1061} pairwise correlations have strong correlation ($|\rho| > 0.6$). Based on the histogram in Figure 4.1, it suggests that a moderate variable dependency structure exists among the 1,061 variables. This might explain why more than 50% of the variables were reduced by all three variable selection approaches tested. On the other hand, the $C_2^{476} = 113,050$ pairwise correlations for the 476 variables selected by InvFDR were computed. The histogram for the pairwise correlation coefficients of the 476 variables (Figure 4.2) indicates that about 0.00796% of C_2^{476} pairwise correlations have very strong

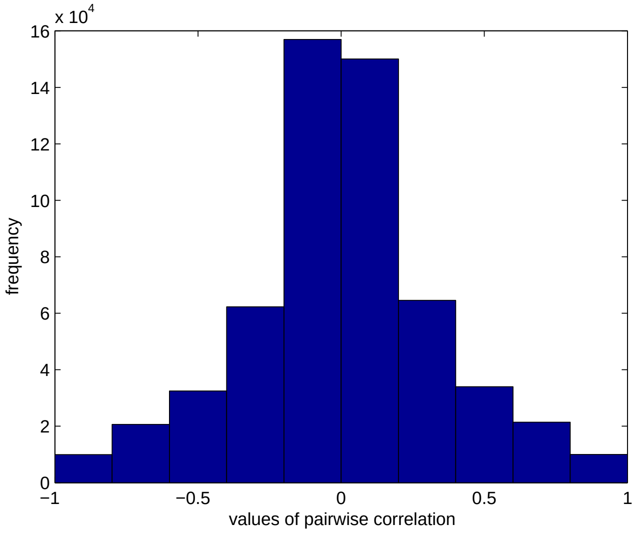


Figure 4.1. Histogram for pairwise correlation of the 1061 predictor variables for the airline SP-FAM.

correlation ($|\rho| > 0.8$), and 0.11% of C_2^{476} pairwise correlations have strong correlation ($|\rho| > 0.6$). These relatively low proportions of highly correlated variables implies that InvFDR was adequately performed.

4.3.2 Convex MARS for SP-FAM after Variable Selection

Using the average profit response values (y) at the feasible CCA points ($\boldsymbol{x}$), Pilla [41] fit an ASR-MARS [52] approximation to a data set of 3,562 points (141 initial feasible CCA points, 1,525 extreme points, and 1,896 interior points) from the airline FAM problem. This MARS fit resulted in 84 basis functions with a coefficient of determination (R^2) of 99.013%. A validation data set of 1600 points was used to test the MARS approximation, and relative errors were computed using the formula $\frac{|y-\hat{f}|}{y}$. In this dissertation, we tested ASR-MARS, Convex-MARS-II and Convex-MARS-II-T in combination with the results from the three variable selection methods. ASR-MARS does not guarantee convexity.

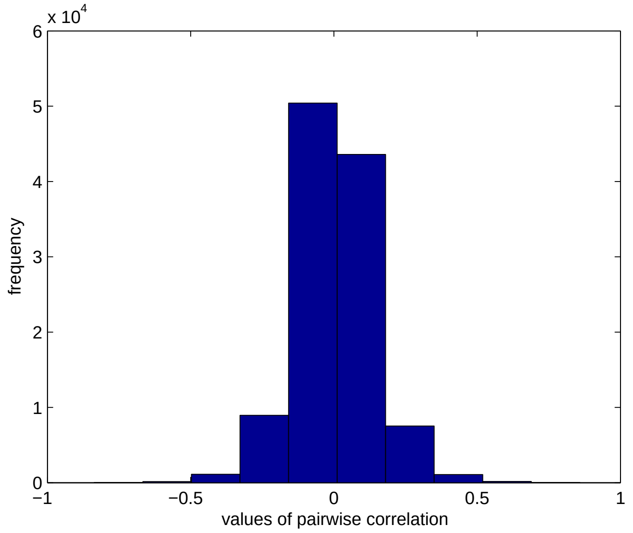


Figure 4.2. Histogram for pairwise correlation of the 476 predictor variables reduced by Inverse FDR for the airline SP-FAM.

Boxplots of the relative errors from the nine models are shown in Figure 4.3. For comparison, Pilla [41] recorded that the maximum relative error of his ASR-MARS model was obtained as 4.6×10^{-5}, and the median relative error was 9×10^{-6}. Further, the tested nine models provide an alternative to approximate the expected profit with reduced CCA decision variable space. The performance of Convex-MARS is comparable with the ASR-MARS. Because Convex-MARS guarantees convexity, optimization methods depending on this convexity assumption can be reliably employed. In terms of accuracy, all nine models appear to be comparable with the result using all 1264 predictor variable in Pilla [41]. Overall, the maximum relative error is less than 1.25×10^{-4}, and the median for the relative error is 1.2×10^{-5}. To further justify the results, 36 variables were randomly selected and tested. This resulted in a maximum relative error of 0.0727 and a median relative error of 0.0693, which are clearly larger than those using the three variable selection methods (Figure 4.3).

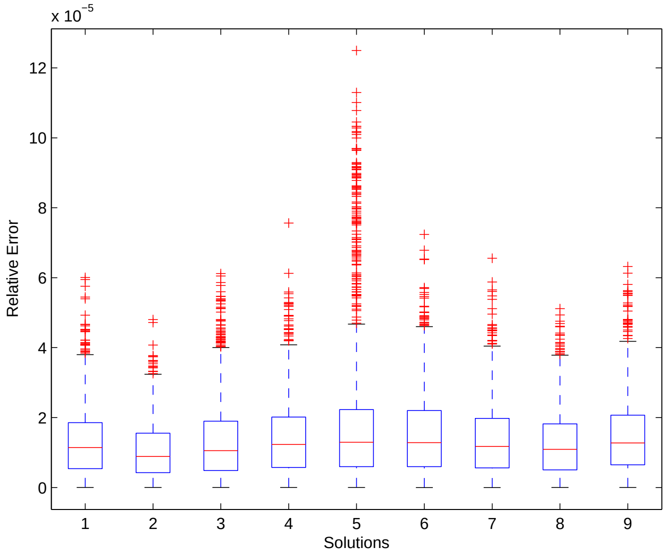


Figure 4.3. For the airline SP-FAM, the boxplots for relative errors based on a validation set of 1600 points from three variable selection methods combined with three versions of MARS {1:CART, 2:FDR/CART, 3:InvFDR}⊗[ASR-MARS], {4:CART, 5:FDR/CART, 6:InvFDR}⊗[Convex-MARS-II-T], and {7:CART, 8:FDR/CART, 9:InvFDR}⊗[Convex-MARS-II].

Table 4.2 summarizes the parameter settings for ASR-MARS and Convex-MARS in conjunction with the three variable selection procedures. In particular, the threshold for Convex MARS-II-T (Convex version of MARS with coefficient *threshold* constraint), 150,000, is determined by multiplying 19.32% by the absolute of maximum coefficient in ASR-MARS model based on the 476 predictors. The MARS algorithms were written in C on and executed on a Dual 2.6-GHz Athlon Workstation. In terms of CPU time, ASR-MARS with selected variables from the three variable selection approaches requires lower computational effort (Table 4.3), but does not guarantee convexity. Pilla [41] showed that 18 of 1000 points randomly chosen to test the convexity of the recourse function proved to be non-convex. However, convexity must be assured to obtain the global optimum for the two-stage SP-FAM problem. To some extent, versions of Convex-MARS guarantee the convexity of their approximations at the expense of run time. It is seen from Table 4.3

Table 4.2. Setting of algorithms on the three versions of MARS based on the FAM problem

	ASR-MARS	Convex-MARS-II-T	Convex-MARS-II
	CART,FDR,InvFDR	CART,FDR,InvFDR	CART,FDR,InvFDR
dim	36 ,565 , 476	36, 565, 476	36, 565, 476
Robust: tolerance	0.02	N/A	N/A
Convex: threshold	N/A	150,000	0
Common Setting:		ASR-I:diff .015 ,	points: 3,562,
		konts:5, int.:2 ,	M_{max}:1,264.

that the maximum difference of CPU time between Convex-MART-T and ASR-MARS for the SP-FAM problem is 18.4 minutes.

Table 4.3. Comparison of CPU time for each version of MARS using the three variable selection approaches based on the FAM problem

	ASR-MARS	Convex-MARS-II-T	Convex-MARS-II
CART Variable Importance Scores	29 sec	18 sec	73 sec
FDR with 3 Groups from CART	548 sec	1654 sec	1295 sec
Inverse FDR	684 sec	936 sec	888 sec

CHAPTER 5

CONCLUSION

5.1 Contributions

One of the major contributions of this research is that convex versions of MARS have been developed and tested on two complex complex optimization applications, an inventory forecasting stochastic dynamic programming problem and a two-stage stochastic programming fleet assignment problem. Convex-MARS essentially modifies both the MARS basis functions and algorithms to build a sum of convex functions; therefore, the final approximation will be convex. Convex versions of MARS proposed in this dissertation include Convex-MARS-I, Convex-MARS-II, and Convex-MARS-II-T. In particular, the basis functions for interaction terms have been developed based on Algorithm 1 (CIT-I) or Algorithm 5 (CIT-II) to avoid the inherent non-convex property from original MARS. Friedman's MARS replaces the truncated linear basis functions $[\pm(x-k)]_+$ in the forward and backward stepwise algorithms with cubic functions, which provides a continuous first derivative and a continuous second derivative everywhere except at the side knots. To give MARS continuity and a continuous first and second derivative at the side knots, quintic functions derived by Chen et al. [14] are utilized in place of Friedman's cubic functions. All these convex versions use the same smoothing algorithm as original MARS since the interaction terms are transformed to new one-dimensional variables with basis function forms that can be treated like univariate basis functions. Since Convex-MARS guarantees convexity, optimization methods depending on this assumption can be reliably employed.

Traditional two-stage stochastic programming uses a Benders' approach or L-shaped method [6]; however, for large-scale problems, this can be slow to converge. Pilla [41] developed a two-phase DACE approach to reduce the computation involved in conducting the optimization. The DACE phase uses first-stage constraints in a multi-step process to construct an experimental design within the feasible region and then builds a statistical model that approximates the expected profit function in the first stage of the stochastic program. The optimization phase solves the two-stage problem using the DACE expected profit approximation instead of solving many second-stage subproblems in every iteration. Although this greatly speeds up the optimization, compared to Benders', the computation of the subproblems is shifted to the DACE phase. In this dissertation, the potential for further speedup is demonstrated by conducting variable selection prior to the DACE phase.

For variable selection, the major contributions are two new approaches for the "large p and small n" problem based on multiple testing procedures using FDR. The first approach is a simple scheme that uses regression trees (CART) to group the response variable observations for the multiple testing procedure. The second approach, called Inverse FDR, switches the roles of response and explanatory variables. Both are designed to handle non-monotonic response surfaces.

5.2 Future Research

Table 5.1 shows the strengths and weaknesses for the methodologies developed in this dissertation. With regard to both computational requirements and quality of solutions, Convex-MARS and ASR-MARS are considered competitive with each other for both inventory forecasting problem and airline fleet assignment problem. Future research in the area of this dissertation includes the following:

1. Develop methods to automatically determine parameters for Convex MARS, e.g., the number of knots (K), or the threshold for Convex-MARS-T. This will enable easier use of the algorithms.
2. Compare versions of Convex-MARS more generally and comprehensively. In this dissertation, two complex optimization problems have been tested in the paradigm of Convex-MARS. Other than these two specific DACE-based applications, versions of Convex-MARS can be implemented for the other optimization DACE-based optimization problems.
3. Develop versions of Convex MARS that are compatible with Robust MARS. Robust MARS seeks to obtain a MARS approximation that is less sensitive to extreme points. The robust entity is not incorporated in current versions of Convex-MARS.
4. Explore potential new variable selection methods such as entropy based approaches, and FDR under dependency. Although, the methods in Section 4.2 provide important subsets of the explanatory variable space, they cannot guarantee the orthogonality of the reduced variable space. An orthogonal subset of the explanatory variable space is easier to handle in both the design of experiments and statistical modeling tasks of DACE-based approach.
5. Develop a parallel version of Convex-MARS to further reduce the time required to generate the Convex-MARS approximation. A parallel version of MARS (Tsai [51]), taking advantage of the parallel computing power, has already been developed. This can be combined with parallel implementations of DACE-based optimization approaches.

Table 5.1.Summary of the proposed methodologies in this dissertation

	Strengths	Weaknesses
FDR/CART	can handle continuous response	response is categorized
Inverse FDR	can handle continuous response	test statistics assumed independent
Convex-MARS-I	convex	moderate accuracy
Convex-MARS-II	convex , flexible	slower run time
Convex-MARS-II-T	convex , flexible	MLR has to be performed to set threshold, slower run time

REFERENCES

[1] Bakin, S., Hegland, M., and Osborne, M. R. Parallel MARS algorithm based on B-splines. *Computational Statistics 15* (2000), 463–484.

[2] Ball, M. O., and Lin, F. L. A reliability model applied to emergency service vehicle location. *Operations Research, Special Issue on Stochastic and Dynamic Models in Transportation 41*, 1 (1993), 18–36.

[3] Beasley, J. E. OR-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society 41*, 11 (1990), 1069–1072.

[4] Benjamini, Y., and Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. B 57* (1995), 289–300.

[5] Berry, M. J. A., and Linoff, G. *Mastering Data Mining: The Art and Science of Customer Relationship Management.* Wiley, New York, 2000.

[6] Birge, J. R., and Louveaux, F. *Introduction to Stochastic Programming.* Springer, New York, New York, 1997.

[7] Box, G. E. P., and Draper, N. R. *Empirical Model Building and Response Surfaces.* Wiley, New York, 1987.

[8] Box, G. E. P., and Wilson, K. B. On the experimental attainment of optimum conditions. *J. Roy. Statist. Soc. Ser. B 13* (1951), 1–45.

[9] Bozdag, C. E., Kahraman, C., and Ruan, D. Fuzzy group decision making for selection among computer integrated manufacturing systems. *Computers In Industry 51* (2003), 13–29.

[10] Breiman, L. Bagging predictors. *Machine Learning 726* (1996), 123–140.

[11] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. *Classification and Regression Trees.* Belmont, Wadsworth, California, 1984.

[12] Bretthauer, K. M., and Shetty, B. A pegging algorithm for the nonlinear resource allocation problem. *Computers and Operations Research 29* (2002), 505–527.

[13] Cervellera, C., Chen, V. C. P., and Wen, A. Neural network and regression spline value function approximations for stochastic dynamic programming. *Computers and Operations Research 34* (2006), 70–90.

[14] Chen, V. C. P. *Applying Experimental Design and Regression Splines to High-Dimensional Continuous-State Stochastic Dynamic Programming.* PhD thesis, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York, U.S.A., 1993.

[15] Chen, V. C. P. Application of MARS and orthogonal arrays to inventory forecasting stochastic dynamic programs. *Computational Statistics and Data Analysis 30* (1999), 317–341.

[16] Chen, V. C. P. Measuring the goodness of orthogonal array discretizations for stochastic programming and stochastic dynamic programming. *SIAM Journal of Optimization 12*, 2 (2001), 322–344.

[17] Chen, V. C. P., Günther, D., and Johnson, E. L. Solving for an optimal airline yield management policy via statistical learning. *Journal of the Royal Statistical Society, Series C 52*, 1 (2003), 1–12.

[18] Chen, V. C. P., Ruppert, D., and Shoemaker, C. A. Applying experimental design and regression splines to high-dimensional continuous-state stochastic dynamic programming. *Operations Research 47* (1999), 38–53.

[19] Chen, V. C. P., Tsui, K.-L., Barton, R. R., and Allen, J. K. *A Review of Design and Modeling in Computer Experiments*, vol. 22 of *In Handbook of Statistics (C. R. Rao and Ravi Khattree, eds.).* Elsevier Science, NY, 2003, pp. 231–261.

[20] Chen, V. C. P., Tsui, K.-L., Barton, R. R., and Meckesheimer, M. Design, modeling, and applications of computer experiments. *IIE Transactions 38* (2006), 273–291.

[21] Chong, E. K. P., and Zak, S. H. *An Introduction to Optimization*, 2nd ed. ed. Wiley, New York, 2001.

[22] Chvatal, V. *Linear Programming.* Freeman, 1983.

[23] Collobert, R., and Bengio, S. SVMTorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research 1*, 2 (2001), 143–160.

[24] Freund, Y., and Schapire, R. *Experiments with a New Boosting Algorithm.* Machine Learning: Proceedings of the Thirteenth International Conference, 1996, pp. 148–156.

[25] Friedman, J. H. Multivariate adaptive regression splines (with discussion. *Annals of Statistics 19* (1991), 1–141.

[26] Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of Statistics 29* (2001), 1189–1232.

[27] Friedman, J. H., Hastie, T., and Tibshirani, R. Additive logistic regression: a statistical view of boosting. *Annals of Statistics 28* (2000), 337–407.

[28] Friedman, J. H., and Silverman, B. W. Flexible parsimonious smoothing and additive modeling. *Technometrics 31* (1989), 3–39.

[29] Frost, V. S., and Melamed, B. Traffic modeling for telecommunications networks. *IEEE Communications Magazine 32* (1994), 70–81.

[30] Hastie, T., Friedman, J. H., and Tibshirani, R. *Elements of Statistical Learning: Data Mining.* Springer-Verlag, New York, 2001.

[31] Haykin, S. S. *Neural Networks: A Comprehensive Foundation*, second edition ed. Prentice Hall, Upper Saddle River, NJ, 1999.

[32] Kim, S. B., Tsui, K. L., and Borodovsky, M. Multiple hypothesis testing in large-scale contingency tables: inferring patterns of pair-wise amino acid association in β-sheets. *Journal of Bioinformatics Research and Applications 2* (2006), 193–217.

[33] Kleijnen, J. P. C. An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research 164*, 2 (2005), 287–300.

[34] Lippmann, R. P. An introduction to computing with neural nets. *IEEE ASSP Magazine* (1987), 4–22.

[35] Luenberger, D. G. *Linear and Nonlinear Programming.* Springer, 2003.

[36] Lyytinen, K., Mathiassen, L., and Ropponen, J. Attention shaping and software risk-A categorical analysis of four classical risk management aproaches. *Information Systems Research 9* (1998), 233–255.

[37] Mastrolilli, M., and Gambardella, L. M. Effective neighbourhood functions for the flexible job shop problem. *Journal of Scheduling 3* (2000), 3–20.

[38] Matheron, G. Principles of geostatistics. *Economic Geology 58* (1963), 1246–1266.

[39] Mulvey, J. M., and Shetty, B. Financial planning via multi-stage stochastic optimization. *Computers and Operations Research 31* (2004), 1–20.

[40] Myers, R. H., and Montgomery, D. C. *Response Surface Methodlogy: Process and Product Optimization Using Designed Experiments.* Wiley, New York, 1995.

[41] Pilla, V. L. *Robust Airline Fleet Assignment.* PhD thesis, University of Texas at Arlington, 2006.

[42] Porteus, E. L. Optimal lot sizing, process quality improvement and setup cost reduction. *Operations Research 34*, 1 (1986), 137–144.

[43] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. Design and analysis of computer experiments (with discussion). *Statistical Science 4* (1989), 409–423.

[44] Schapire, R. The boosting approach to machine learning: An overview, 2001.

[45] Shaffer, J. Multiple hypothesis testing. *Annu. Rev. Psychol. 46* (1995), 561–584.

[46] Shih, D. T., Pilla, V. L., Kim, S. B., Rosenberger, J. M., and Chen, V. C. P. Efficient computer experiment-based optimization through variable selection. In *Proceedings of the 2006 INFORMS Annual Conference* (Pittsburgh, PA, USA, October 2006).

[47] Siddappa, S. *Statistical Modeling Approach to Airline Revenue Management with Overbooking*. PhD thesis, University of Texas at Arlington, 2006.

[48] Smith, B. L., and Demetsky, M. J. Traffic flow forecasting: Comparison of modeling approaches. *Journal of Transportation Engineering 123* (1997), 261–266.

[49] Storey, J. D., and Tibshirani, R. Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci. 100* (2003), 9440–9445.

[50] Tokhomirov, V. M. The evolution of methods of convex optimization. *Amer. Math. Monthly 103* (1996), 65–71.

[51] Tsai, J. C. C. *Statistical Modeling of the Value Function in High-Dimensional, Continuous-State Stochastic Dynamic Programming*. PhD thesis, Georgia Institute of Technology, 2002.

[52] Tsai, J. C. C., and Chen, V. C. P. Flexible and robust implementations of multivariate adaptive regression splines within a wastewater treatment stochastic dynamic program. *Quality and Reliability Engineering International 21* (2005), 689–699.

[53] Tsai, J. C. C., Chen, V. C. P., Chen, J., and Beck, M. B. Stochastic dynamic programming formulation for a wastewater treatment decision-making framework. *Annals of Operations Research, Special Issue on Applied Optimization Under Uncertainty 132* (2004), 207–221.

[54] Tsui, K.-L., Chen, V. C. P., Jiang, W., and Aslandogan, Y. A. 2006. In *Handbook of Engineering Statistics*, H. Pham, Ed., vol. in press. Springer-Verlag, London, 2006.

[55] Vapnik, V., Golowich, S., and Smola, A. *Support vector method for function approximation, regression estimation, and signal processing*, vol. 9. MIT Press, Cambridge, MA, 1997, ch. Advances in Neural Information Processing Systems, pp. 281–287.

[56] Wegman, E. Data mining tutorial. Short Course Notes. Interface 2001 Symposium, 2001.

[57] Wen, A. *Statistics-Based Approaches to Stochastic Optimal Control Problems.* PhD thesis, University of Texas at Arlington, 2005.

[58] Yang, Z. *A Decision-Making Framework for Ozone Pollution Control.* PhD thesis, University of Texas at Arlington, 2004.

BIOGRAPHICAL STATEMENT

Dachuan Thomas Shih was born in Chang-hwa, Taiwan, in 1971. He received his B.S. and M.S. degrees from National Chengchi University, Taipei, in 1994, and 1996, all in Statistics. In 1998, he joined Chuanseng Paper Tube Ltd., as quality control engineer for the design and optimization of paper tube production systems. From 2000 to 2002, he worked with the Mentor Data Systems, as a quality assurance engineer, then a quality assurance supervisor in Taiwan. His current research interest is in the area of data mining, statistical process control, applid statistics and operations research. He is a member of Tau Beta Pi honor society for engineers since 2004. He pursued his Ph.D. in Industrial Engineering at University of Texas at Arlington since 2003.