A Convex Version of Multivariate Adaptive Regression Splines

Dachuan T. Shih

Conifer Health Solutions

2401 Internet Boulevard

Frisco, TX 75034

Diana L. Martinez and Victoria C. P. Chen (correspondence author)

Department of Industrial and Manufacturing Systems Engineering

The University of Texas at Arlington

Campus Box 19017

Arlington, TX 76019 USA E-mail: vchen@uta.edu

Seoung Bum Kim, School of Industrial Management Engineering

Korea University, Seoul, Republic of Korea

COSMOS Technical Report #12-01

Abstract

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 and simultaneously conduct variable selection. In optimization, MARS has been used successfully to estimate the unknown functions in stochastic dynamic programming (SDP), stochastic programming, and a Markov decision process, 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 paper a convex MARS modeling algorithm is described. 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 altered to eliminate the inherent nonconvexity. Finally, MARS convexity can be achieved by the fact that the sum of convex functions is convex. Convex-MARS is applied to inventory forecasting SDP problems with four and nine dimensions.

1 Introduction

Computer modeling is having a profound effect on scientific research. Many processes are so complex that physical experimentation is too time-consuming, too expensive or simply 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. The purpose of design and analysis of computer experiments (DACE, Sacks et al. 1989; Kleijnen 2008; Chen et al. 2006) 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. In DACE, the computer experiment replaces the physical experiment by organizing computer model runs and observing the model output of performance. A common DACE objective is to obtain a computationally-efficient response surface approximation (a.k.a., metamodel) of the output. This metamodel may then be used to study and potentially "optimize" the performance of the system. The effectiveness of an optimization method in using a metamodel to improve system performance depends on the convexity of the objective function (Luenberger 2004). A nonconvex metamodel requires a global optimization method, and in practice these cannot guarantee optimality. Consequently, if the true underlying performance objective function is known to be convex, it is highly desirable for the approximating metamodel to share this critical property.

Multivariate adaptive regression splines (MARS, Friedman 1991) modeling has been applied in DACE-based approaches for some large-scale optimization problems, including continuous-state stochastic dynamic programming (SDP, Chen 1999; Chen et al. 1999; Tsai et al. 2004; Tsai and Chen 2005; Cervellera et al. 2007; Yang et al. 2007, 2009), Markov decision processes (MDP, Chen et al. 2003; Siddappa et al. 2007, 2008), and two-stage stochastic programming (SP, Pilla et al. 2008, 2012; Shih et al. 2012). 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. 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, so 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 revenue objective function, so as to speed up the first-stage optimization. MARS has been successful in these applications not only because of the flexibility of its modeling, but also its parsimony. Parsimony is critical in achieving computational-tractability in large-scale complex problems. Shih et al. (2012) added a data mining variable selection phase that reduced the dimension of the airline fleet assignment model from about 1200 to 400 variables prior to executing DACE, so as to reduce the computational effort of DACE from 2.5 days to an estimated 0.5 days.

Under the assumption that an optimization function f is convex, it is desired that the response surface metamodel $\hat{f}$ 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 (Box and Draper 1987), spatial correlation models, a.k.a., kriging (Sacks et al. 1989), MARS, regression trees (Breiman et al. 1984; Friedman 2001), and artifical neural networks (Haykin 1999). None of these guarantee convexity. 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.

2 Multivariate Adaptive Regression Splines (MARS)

Friedman (1991) introduced MARS as a statistical method for high-dimensional modeling with interactions. The MARS model is essentially a linear statistical model with a forward stepwise algorithm to select model terms followed by a backward procedure to prune the model terms. A univariate version (appropriate for additive relationships) was presented by Friedman and Silverman (1989). The MARS approximation bends to model curvature at "knot" locations, and one of the objectives of the forward stepwise algorithm is to simultaneously select variables and appropriate knots. After selection of the basis functions is completed, smoothness to achieve a certain degree of continuity may be applied. MARS is both flexible and easily implemented with the computational effort primarily dependent on the number of basis functions added to the model. The MARS approximation is a linear model:

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

where $B_m(\boldsymbol{x})$ initially is a basis function of the form described below in equation (1) 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 represented in the form of truncated linear functions,

$$b^{+}(x;k) = [+(x-k)]_{+}, \quad b^{-}(x;k) = [-(x-k)]_{+}, \tag{1}$$

where $[q]_{+} = \max\{0, q\}$ and k is a univariate knot. The set of eligible knots are assigned separately for each input variable dimension and are chosen to coincide with input levels represented in the data. Interaction basis functions are created by multiplying an existing basis function with a truncated linear function involving a new variable. Both the existing "parent" basis function and the newly created interaction basis function are used in the MARS approximation. Thus, the form of the m-th basis function is

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

where $x_{v(l,m)}$ is the input 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. L_m is the number of truncated linear functions multiplied in the m-th basis function. The search for new basis functions can be restricted to interactions of a maximum order (e.g., $L_m \leq 2$ permits up through two-factor interactions). Using a generalized cross-validation lack-of-fit criterion, basis functions are added in pairs, corresponding to the two forms in equation (1). The algorithm stops when M_{max} basis functions have been selected, where M_{max} is user-specified. The original forward MARS algorithm is represented in Algorithm 1, and the key to Convex-MARS is reconstruction of this forward algorithm.

The MARS backward algorithm was intended to eliminate overfitting, but due to the extremely low error variability in most DACE applications, this can often be omitted to save computational effort. To compensate for the omission of the backward algorithm, Tsai and Chen (2005) modified the MARS forward algorithm to incorporate an automatic stopping rule (ASR) and seek more robust models with fewer high-order interaction terms. Instead of using the original MARS stopping rule that depends on a user-specified M_{max} in the MARS forward stepwise algorithm, ASR stops automatically based on the improvement in the coefficient of determination or adjusted coefficient of determination. This enables an automated implementation of MARS model fitting within optimization routines (Tsai and Chen 2005). The robust component aims to obtain a MARS approximation that is less sensitive to extreme points by selecting the lower-order terms over high-order ones given the contributions or fits are comparable. The actual implementation of Convex-MARS is based on Robust ASR-MARS; however, for clarity, the presented Convex-MARS forward algorithm will follow the structure of Algorithm 1.

Friedman's MARS provides a continuous first derivative everywhere by replacing the truncated linear basis functions with cubic functions after completing the forward and backward algorithms. To give MARS a continuous second derivative everywhere, quintic functions derived by Chen (1993) are used in place of Friedman's cubic functions. First, define two side knots k_+ and k_- (in addition to the original center knot k). Then define $\Delta = k_+ - k_-$, $\Delta_1 = k_+ - k$, and $\Delta_2 = k - k_-$. The quintic functions can be written as:

$$Q(x|s=+1,k_-,k,k_+) = \begin{cases} 0, & x \leq k_- \\ \alpha_+(x-k_-)^3 + \beta_+(x-k_-)^4 + \gamma_+(x-k_-)^5, & k_- < x < k_+ \\ x-k, & x \geq k_+, \end{cases} \tag{2}$$

where,

$$\alpha_+ = \frac{[6\Delta_1 - 4\Delta_2]}{\Delta^3},$$
$$\beta_+ = \frac{[-8\Delta_1 + 7\Delta_2]}{\Delta^4},$$
$$\gamma_+ = \frac{[3\Delta_1 - 3\Delta_2]}{\Delta^5},$$

and

$$Q(x|s=-1,k_-,k,k_+) = \begin{cases} k-x, & x \leq k_- \\ \alpha_-(x-k_+)^3 + \beta_-(x-k_+)^4 + \gamma_-(x-k_+)^5, & k_- < x < k_+ \\ 0, & x \geq k_+, \end{cases} \tag{3}$$

where,

$$\alpha_- = \frac{[4\Delta_1 - 6\Delta_2]}{\Delta^3},$$
$$\beta_- = \frac{[7\Delta_1 - 8\Delta_2]}{\Delta^4},$$
$$\gamma_- = \frac{[3\Delta_1 - 3\Delta_2]}{\Delta^5}.$$

3 Achieving Convexity in MARS

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 terms is altered to eliminate the inherent nonconvexity. A preliminary version of Convex-MARS (Shih et al. 2006) essentially incorporated these modifications to guarantee convexity. However the flexibility of this version was limited, so the current paper presents an improved version. Convex-MARS requires the following algorithms: (i) Convex Interaction Transformation Algorithm (CIT), to create the convex forms of the interaction basis functions, (ii) Forward Coefficient Restriction Algorithm (FCR), to incorporate convexity restrictions on the model coefficients while selecting basis functions, and (iii) Backward Pruning and Refitting Algorithm (BPR), to check for nonconvexities and eliminate them. The BPR algorithm is needed because MARS basis functions are overlapping; hence, the addition of new basis functions can alter the existing coefficients. However, a well constructed Convex-MARS approximation should make minimal use of BPR.

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 (1). An unpaired univariate basis function takes on only one of the forms in equation (1). 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 1 display two forms in equation (1) with $k = 0$. The lower left plot in Figure 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 1 here. $>$

3.2 Convex Interaction Terms

A nonconvex MARS approximation is inherently possible because interaction terms are products of univariate terms. In this case, not only must the coefficient for the interaction basis function 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 interaction basis functions are constructed so that the smoothing in Section 2 can be applied. To achieve this, the variables involved in the interaction basis function are transformed via a rotation of their axes, and univariate truncated linear basis functions are formed along the rotated axes. Figure 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, coefficients must be constrained in the same manner as univariate basis functions in the previous section. Details on the algorithms for Convex-MARS are given in Section 4.

< Figure 2 here. >

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

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

where the notation is the same as defined in Section 2. The convex form of interaction basis functions transforms the multiple variables in the interaction to a one-dimensional variable via a linear combination. Given the set of variables $x_{v(l,m)}$ for an interaction term and corresponding knots $k_{l,m}$ and signs $s_{l,m}$, define:

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

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

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

$$z^+(\boldsymbol{x}) = \omega_0(\boldsymbol{x}) + \omega_1(\boldsymbol{x}; s_{L,m} = +1) \ ; \ z^-(\boldsymbol{x}) = \omega_0(\boldsymbol{x}) + \omega_1(\boldsymbol{x}; s_{L,m} = -1) \ \ . \quad (4)$$

To show that z^+ or z^- are linear combinations of the input variables, re-write:

$$\omega_0(\boldsymbol{x}) = a_0 + \sum_{l=1}^{L_m-1} a_l\, x_{v(l,m)} \ \ , \quad (5)$$

where,

$$a_0 = \sum_{l=1}^{L_m-1} k_{l,m}s_{l,m}/(s_{l,m}k_{l,m} - 1) \ \ , \ \ a_l = s_{l,m}/(1 - s_{l,m}k_{l,m}) \ \ , \quad (6)$$

and

$$\omega_1(\boldsymbol{x}; s_{L,m}) = s_{L,m} k_{L_m,m}/(s_{L,m}\, k_{L_m,m} - 1) + s_{L,m}/(1 - s_{L,m} k_{L_m,m}) \cdot x_{v(L_m,m)} \quad . \tag{7}$$

Given z^+ or z^-, we can now define pairs of univariate truncated linear functions, as in equation (1), with sign ϕ either $+1$ or -1 for each pair:

$$b^+(z^+;\tau) = [+(z^+ - \tau)]_+ \quad , \quad b^-(z^+;\tau) = [-(z^+ - \tau)]_+ \quad \text{or} \tag{8}$$

$$b^+(z^-;\tau) = [+(z^- - \tau)]_+ \quad , \quad b^-(z^-;\tau) = [-(z^- - \tau)]_+ \quad . \tag{9}$$

Because the transformation defined in equations (4)–(7) also transforms the multivariate knot $\boldsymbol{k}$ to $\tau = 0$ in equations (8)–(9), the two candidate pairs of interaction basis functions for Convex-MARS are as follows:

$$B_m(\boldsymbol{x}; \phi_m = +1) = [z^+]_+ \quad , \quad B_{m+1}(\boldsymbol{x}; \phi_m = -1) = [-z^+]_+ \quad \text{or} \tag{10}$$

$$B_m(\boldsymbol{x}; \phi_m = +1) = [z^-]_+ \quad , \quad B_{m+1}(\boldsymbol{x}; \phi_m = -1) = [-z^-]_+ \quad . \tag{11}$$

To better understand the role of ϕ_m, consider the two-way interaction example shown in Figure 3. Knots for the two input variables x_1 and x_2 are 0.25 and -0.5, respectively, and $s_{1,m} = -1$. The upper two contour plots set $s_{2,m} = 1$ and demonstrate the pair of two-way interaction basis functions with $\phi_m = +1$, and $\phi_m = -1$. Similarly, the lower two contour plots set $s_{2,m} = -1$ and show the pair of two-way interaction basis functions with $\phi_m = +1$ and $\phi_m = -1$.

< Figure 3 here. >

3.3 Convexity Proof

The convexity constraints on the coefficients were identified based on the truncated linear basis functions, but the final MARS approximation employs the smoothed quintic functions in equations (2) and (3). In this section, a convexity proof is provided for the pairs of MARS univariate terms in quintic form. The proof of convexity of a pair of Convex-MARS interaction terms in quintic form is identical for both univariate basis functions and interactions using the transformed variables z^+ or z^-. The goal is to prove that the combined pair of univariate terms:

$$Q = \beta_1 Q(x|s = +1, k_-, k, k_+) + \beta_2 Q(x|s = -1, k_-, k, k_+) \tag{12}$$

is a convex function on $[k_-, k_+]$, where β_1 and β_2 are coefficients of basis functions. As illustrated in Figure 1, convexity is assured for the truncated linear functions if $\beta_1 + \beta_2 \geq 0$. Without loss of generality, we express this case as $\beta_2 = -\beta_1 + d$, where d is a nonegative constant, $d \geq 0$, and rewrite equation (12) as:

$$Q = \beta_1 Q(x|s = +1, k_-, k, k_+) - \beta_1 Q(x|s = -1, k_-, k, k_+) + dQ(x|s = -1, k_-, k, k_+). \tag{13}$$

A twice differentiable function is convex if and only if its second derivative is ≥ 0. To prove that equation (13) 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 (please refer to the notation in Section 2:

$$\begin{aligned} Q'' = \beta_1 & \left(\frac{6[6\Delta_1 - 4\Delta_2][(x - k_-)]}{\Delta^3} + \frac{12[-8\Delta_1 + 7\Delta_2][(x - k_-)^2]}{\Delta^4} + \frac{20[3\Delta_1 - 3\Delta_2][(x - k_-)^3]}{\Delta^5} \right) - \\ & \beta_1 \left(\frac{6[4\Delta_1 - 6\Delta_2][(x - k_+)]}{\Delta^3} + \frac{12[7\Delta_1 - 8\Delta_2][(x - k_+)^2]}{\Delta^4} + \frac{20[3\Delta_1 - 3\Delta_2][(x - k_+)^3]}{\Delta^5} \right) + \\ & d \left(\frac{6[4\Delta_1 - 6\Delta_2][(x - k_+)]}{\Delta^3} + \frac{12[7\Delta_1 - 8\Delta_2][(x - k_+)^2]}{\Delta^4} + \frac{20[3\Delta_1 - 3\Delta_2][(x - k_+)^3]}{\Delta^5} \right). \end{aligned} \tag{14}$$

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_+ (Chen 1993). Specifically, to avoid such nonconvexities, we must constrain

$$\frac{\Delta_1}{\Delta} \geq \frac{2}{5} \text{ and } \frac{\Delta_2}{\Delta} \geq \frac{2}{5}. \tag{15}$$

To simplify equation (14), we use the fact that $\Delta_2 = \Delta - \Delta_1$:

$$Q'' = 12d \left(\frac{(5\Delta_1 - 3\Delta)(x - k_+)}{\Delta^3} + \frac{[15\Delta_1 - 8\Delta][(x - k_+)^2]}{\Delta^4} + \frac{[10\Delta_1 - 5\Delta][(x - k_+)^3]}{\Delta^5} \right), \tag{16}$$

and from equation (15) we constrain

$$\frac{2}{5} \leq \frac{\Delta_1}{\Delta} \leq \frac{3}{5}. \tag{17}$$

Without loss of generality, $\Delta = 1$ is specified, and equation (16) is reduced to:

$$Q'' = 60d(1 - f)\{f[(\Delta_1 - \frac{2}{5}) + f(1 - 2\Delta_1)]\},$$

where $f = (x - k_-)$ and $(1 - f) = (k_+ - x)$. Since $d \geq 0$ and $0 \leq f \leq 1$, proving $Q'' \geq 0$ requires showing:

$$(\Delta_1 - \frac{2}{5}) + f(1 - 2\Delta_1) \geq 0. \tag{18}$$

Under the constraint in (17), the left-hand side of (18) is minimized at $f = 1$ and $\Delta_1 = \frac{3}{5}$, at which it is equal to zero. Hence, equation (18) holds, and Q is convex on $[k_-, k_+]$.

4 Algorithms for Convex-MARS

4.1 Convex-MARS Forward Coefficient Restriction Algorithm

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. This modified algorithm constrains the coefficients for the basis functions throughout the search process. Whenever there are basis functions being added to the current set of basis functions, either a pair or an unpaired basis function (univariate or interaction) is possible. In the first case, the sum of the two coefficients are constrained to be nonnegative. In the latter case, the coefficient is restricted to be nonnegative. In the preliminary version of Convex-MARS (Shih et al. 2006), the interaction basis functions were not added in pairs, limiting the flexibility of the Convex-MARS approximation.

The forward coefficient restriction algorithm (FCR) is shown in Algorithm 3. FCR incorporates the convex interaction transformation (CIT) in equations (4)–(7) that is shown in Algorithm 2. To improve the fit of Convex-MARS interaction basis functions, FCR conducts a more flexible search for candidate pairs of basis functions. This is achieved by considering both pairs in equations (10)–(11) into the search loop for interaction terms.

4.2 Convex-MARS Backward Pruning and Refitting Algorithm

Since MARS basis functions are overlapping, the coefficients may change each time new basis functions are added in the model. The backward pruning and refitting algorithm (BPR) re-checks the coefficients after running FCR, searches for convexity violations and removes them. Intuitively, if the true underlying function is convex, then basis functions that introduce potential nonconvexity should not be needed. For a pair of basis functions that violate convexity, the basis function with the smaller coefficient will be dropped; however, if both coefficients are negative, this pair will be dropped completely. If any basis functions are dropped, then Convex-MARS must be refit with 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.

4.3 Convex-MARS Forward Coefficient Restriction Threshold Algorithm

The use of BPR is a necessary, but not ideal means of guaranteeing convexity. One practical approach to minimizing the need for BPR is to require stricter convexity in FCR. In computational studies of Convex-MARS, we have observed that pairs whose coefficients barely satisfy convexity are more likely to turn nonconvex in later iterations. Hence, FCR was modified from Algorithm 3 to Algorithm 5 to employ a stricter *threshold* on convexity. In FCR with a stricter threshold (FCR-T), the candidate basis functions are the same as FCR. However, instead of only requiring nonnegativity on the sum of a pair of coefficients (or single coefficient for an unpaired case), a strictly positive threshold is specified.

The challenge now lies in selecting a *threshold* value. One guideline is to run original MARS (with standardized variables) on the data set, then set the threshold to be about $1 \sim 10\%$ of the absolute value of the maximum estimated coefficient. If the *threshold* is set too high, then this can reduce flexibility since fewer candidate basis functions will satisfy the stricter convexity constraint. If the selected *threshold* is too low, then there will be little difference between FCR and FCR-T. In practice, if many basis functions are removed via BPR, then the threshold is too low.

4.4 Convex-MARS Smoothing Procedure

The smoothing routine in Convex-MARS uses the quintic functions presented in Section 2. For the interaction terms, CIT transforms the multiple variables in the interaction term into a one-dimensional variable z^+ or z^-. Since the transformed center knot is always zero for the interaction terms of Convex-MARS, the corresponding side knots can be set symmetrically at κ and $-\kappa$, where κ can be considered as a smoothing factor. The larger the smoothing factor is, the smoother the quintic function will be. Assuming standardized units, one recommendation for κ is 0.5.

5 Inventory Forecasting Application

In this section Convex-MARS is tested on four-dimensional and nine-dimensional inventory forecasting SDP problems studied by Chen (1999). The goal of the inventory forecasting problem is to minimize inventory holding and backorder costs. The state of the system is represented by the inventory levels for the products and their demand forecasts. The optimal value function, known to be theoretically convex, specifies the minimum expected cost to operate the system, and is a function of the system state. The versions of MARS are fit to data from the last time period of the three-period inventory forecasting SDP.

5.1 Four-dimensional Inventory Forecasting Problem

In this inventory forecasting SDP problem, there are two products, each with one demand forecast. MARS was fit to a data set of 125 points from the last period, using the same orthogonal array experimental design studied in Chen (1999). Table 1 summarizes the parameter settings for original MARS, the preliminary version of Convex-MARS, Convex-MARS and Convex-MARS-T (Convex version of MARS with coefficient threshold constraint). A set of 100 randomly generated validation data points is used to compare the different models, and boxplots of the absolute errors, computed using the formula $|y - \hat{f}|$, are shown in Figure 4, where y is the actual cost of the system, and $\hat{f}$ is the MARS or Convex-MARS prediction.

$<$ Table 1 here. $>$

$<$ Figure 4 here. $>$

5.2 Nine-dimensional Inventory Forecasting Problem

In this inventory forecasting SDP problem, 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, using the same orthogonal array experimental design studied in Chen (1999). Table 2 summarizes the parameter setting for original MARS, Convex-MARS and Convex-MARS-T. A set of 1000 randomly generated 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 5, where y is the actual cost of the system, and $\hat{f}$ is the MARS/ Convex-MARS approximations for the actual cost.

$<$ Table 2 here. $>$

$<$ Figure 5 here. $>$

Table 3 shows different values for the threshold that were tested for both, the four-dimensional and the nine-dimensional inventory forecasting problems. They were defined based on various percentages of the maximum absolute coefficient of the MARS approximation. For the four-dimensional case, the minimum value for the median absolute error was obtained from 5.78% to 8.00%. For the nine-dimensional case, a lower value for the median absolute error was obtained from 1.12% to 1.61%. Additionally, Convex-MARS code was not able to select any basis functions when the threshold was defined using above 5%. As it was mentioned previously, determining the threshold value is still being studied.

< Table 3 here. >

5.3 Computational Results

The performance of the Convex-MARS versions is comparable to original MARS in both cases. In the four-dimensional case, the preliminary version of Convex-MARS was also tested with the purpose of showing the improvement of the newer version; the median absolute error is clearly higher than that of original MARS, while the median absolute error of Convex-MARS is now just slightly higher, however both versions guarantee convexity. Most importantly, the Convex-MARS-T version (with threshold) shows a comparable, but slightly superior fit than original MARS. For reference, the mean true cost of the 100 validation points was 137.47. In the nine-dimensional case, the preliminary version of Convex-MARS was not considered in the absolute error plot since it did not show satisfactory results. Convex-MARS and Convex-MARS-T versions demonstrate a similar median absolute error to the original MARS. For reference, the mean true cost of the 1000 validation points was 376.33. Convexity must be assured to obtain the global optimum for these inventory forecasting SDP problems, and future work will incorporate Convex-MARS within an SDP numerical solution method. In terms of CPU time, all the tested MARS runs required less than 5 seconds on a Quad 3.00-GHz 8GB RAM Dell Precision Workstation.

6 Conclusions

The major contribution of this research is a version of MARS that guarantees convexity without degrading the quality of fit. Given the existing success of MARS in some complex, large-scale optimization problems, the convexity guarantee provides stronger motivation to employ Convex-MARS in problems with known convexity. Testing on inventory forecasting SDP problems demonstrates a comparable fit to original MARS. While a significant structural modification for interaction basis functions was required to guarantee convexity, Convex-MARS maintains most of the structure of MARS, including a forward stepwise procedure that adds basis functions in pairs, the use of truncated linear functions, and a smoothing routine to enable continuous derivatives. Some challenges that will be investigated in future work include more study on setting the positive threshold for strict convexity discussed in Section 4.3 and alternate methods for defining univariate directions z^+ or z^- in Section 3.2.

Acknowledgment

This research was partially supported by the Dallas-Fort Worth International Airport and National Science Foundation grant ECCS-0801802.

References

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

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

Cervellera, C., Chen, V. C. P. and Wen, A. (2007). Neural network and regression spline value function approximations for stochastic dynamic programming, *Computers and Operations Research* **34**: 70–90.

Chen, V. C. P. (1993). *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.

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

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

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

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

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

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

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

Haykin, S. S. (1999). *Neural Networks: A Comprehensive Foundation*, second edition edn, Prentice Hall, Upper Saddle River, NJ.

Kleijnen, J. P. C. (2008). *Design and Analysis of Simulation Experiments*, Springer-Verlag, Heidelberg.

Luenberger, D. (2004). *Linear and Nonlinear Programming (second edition)*, Kluwer Academic Publishers, Boston, MA.

Pilla, V. L., Rosenberger, J. M., Chen, V. C. P., Engsuwan, N. and Siddappa, S. (2012). A multivariate adaptive regression splines cutting plane approach for solving a two-stage stochastic programming fleet assignment model, *European Journal on Operational Research* **216**: 162–171.

Pilla, V. L., Rosenberger, J. M., Chen, V. C. P. and Smith, B. C. (2008). A statistical computer experiments approach to airline fleet assignment, *IIE Transactions* **40**: 524–537.

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

Shih, D. T., Chen, V. C. P. and Kim, S. B. (2006). Convex version of multivariate adaptive regression splines, *Proceedings of the 2006 Industrial Engineering Research Conference*, Orlando, FL, USA.

Shih, D. T., Kim, S. B., Chen, V. C. P., Rosenberger, J. M. and Pilla, V. L. (2012). Efficient computer experiment-based optimization through variable selection, *Annals of Operations Reseach*, *special volume on Data Mining and Informatics* **to appear**.

Siddappa, S., Günther, D., Rosenberger, J. M. and Chen, V. C. P. (2007). Refined experimental design and regression splines method for network revenue management, *Journal of Pricing and Revenue Management, INFORMS Special Issue* **6**: 188–199.

Siddappa, S., Rosenberger, J. M. and Chen, V. C. P. (2008). Optimising airline overbooking using a hybrid gradient approach and statistical modeling, *Journal of Pricing and Revenue Management* **7**: 207–218.

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

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

Yang, Z., Chen, V. C. P., Chang, M. E., Murphy, T. E. and Tsai, J. C. C. (2007). Mining and modeling for a metropolitan atlanta ozone pollution decision-making framework, *IIE Transactions, special issue on Data Mining* **39**: 607–615.

Yang, Z., Chen, V. C. P., Chang, M. E., Sattler, M. L. and Wen, A. (2009). A decision-making framework for ozone pollution control, *Operations Research* **57**: 484–498.

Algorithm 1 Original MARS Forward Algorithm

Initialize $M = 1$; maxIA $=$ maximum # input variables in an interation.

while ($m < M_{\text{max}}$) **do**

  LOF = ∞.

  for all $m = 0, \ldots, M-1$ **do**

    if basis function m involves fewer than maxIA input variables **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 2 Convex-MARS Interaction Transformation Algorithm (CIT)

z = (ϕ_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.

Algorithm 3 Convex-MARS Forward Coefficient Restriction Algorithm (FCR)

Initialize $M = 1$; maxIA $=$ maximum # input variables in an interation.

while ($m < M_{\text{max}}$) **do**

  LOF* = ∞.

  for all $m = 0, \ldots, M - 1$ **do**

    if basis function m involves fewer than maxIA input variables **then**

      for all v = 1 to n **do**

        if $v \notin$ basis function m **then**

          for all k = 1 to K **do**

            for all candidate nonzero basis functions **do**

              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 candidate basis functions

          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

Algorithm 4 Convex-MARS Backward Pruning and Refitting Algorithm (BPR)

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 $\cap$ 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

Algorithm 5 Convex-MARS-T Forward Coefficient Restriction Threshold Algorithm (FCR-T)

Initialize $M = 1$; maxIA $=$ maximum # input variables in an interation.

while ($m < M_{\text{max}}$) **do**

LOF* = ∞.

for all m =0,...,$M - 1$ **do**

if basis function m involves fewer than maxIA input variables **then**

for all $v = 1$ to n **do**

if $v \notin$ basis function m **then**

for all $k = 1$ to K **do**

for all candidate nonzero basis functions **do**

if (coefficient $>$ threshold from an unpaired basis function) $\cup$ (sum of coefficients $>$ threshold from a pair of basis functions) **then**

Calculate lack-of-fit LOF.

if LOF $<$ LOF* **then**

LOF* = LOF; save $m*$, $v*$, $k*$, ϕ_m*.

end if

end if threshold

end for candidate basis functions

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

Table 1: Parameter settings for the versions of MARS on the four-dimensional inventory forecasting problem.

	MARS	Preliminary C-MARS	C-MARS	C-MARS-T
Robust/ tolerance:	Robust/ 0.3	N/A	N/A	N/A
Threshold:	N/A	N/A	N/A	23
Common Setting:	knots: 3	points: 125	$M_{\max}$: 100	
	interactions: 3	ASR stopping tolerance: 0.02		

Table 2: Parameter settings for the versions of MARS on the nine-dimensional inventory forecasting problem.

	MARS	C-MARS	C-MARS-T
Robust/ tolerance:	Robust/ 0.3	N/A	N/A
Threshold:	N/A	N/A	31.5
Common Setting:	knots: 9	points: 1331	$M_{\max}$: 300
	interactions: 3	ASR stopping tolerance: 0.02	

Table 3: Comparison of various threshold values based on different percentages of the maximum absolute coefficient from original MARS. Median absolute error is reported.

Four-dimensional			Nine- dimensional		
Percentage	Threshold	Median	Percentage	Threshold	Median
2.00	7.15	5.68	1.12	15.00	7.73
5.00	17.88	5.68	1.49	20.00	7.73
5.59	20.00	6.04	1.57	21.00	7.92
5.87	21.00	4.76	1.61	21.5	7.92
6.00	21.45	4.76	1.72	23.00	8.02
6.43	23.00	4.76	1.87	25.00	17.27
7.00	25.03	4.76	2.00	26.78	17.27
8.00	28.60	4.76	2.24	30.00	9.96
8.39	30.00	5.62	2.61	35.00	36.00
9.00	32.18	5.62	5.00	66.96	-
9.79	35.00	5.62	8.00	107.14	-
10.00	35.76	5.62	10.00	133.93	-

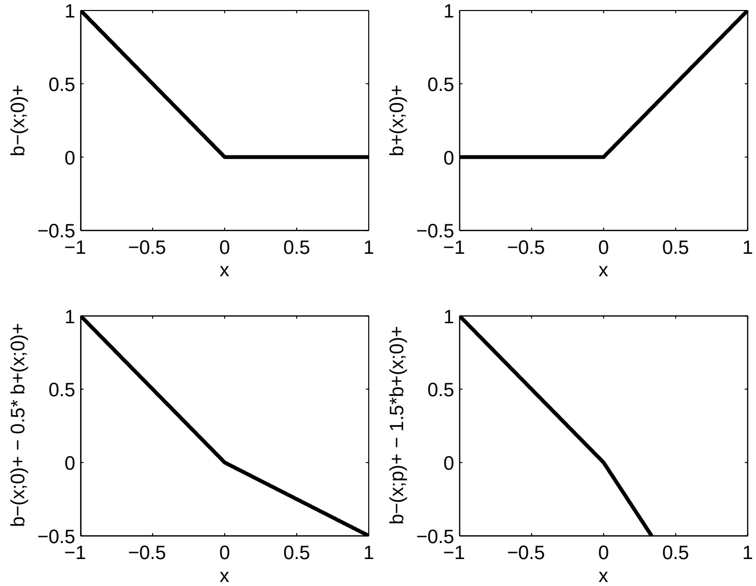

Figure 1: Pair of basis functions.

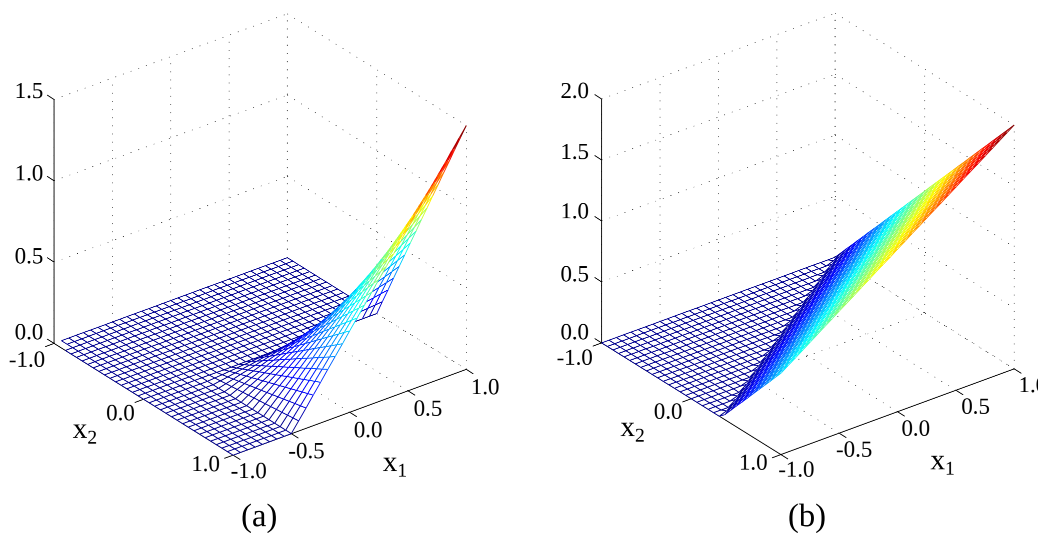

Figure 2: Comparison of MARS interaction basis functions: (a) Original MARS, (b) Convex-MARS.

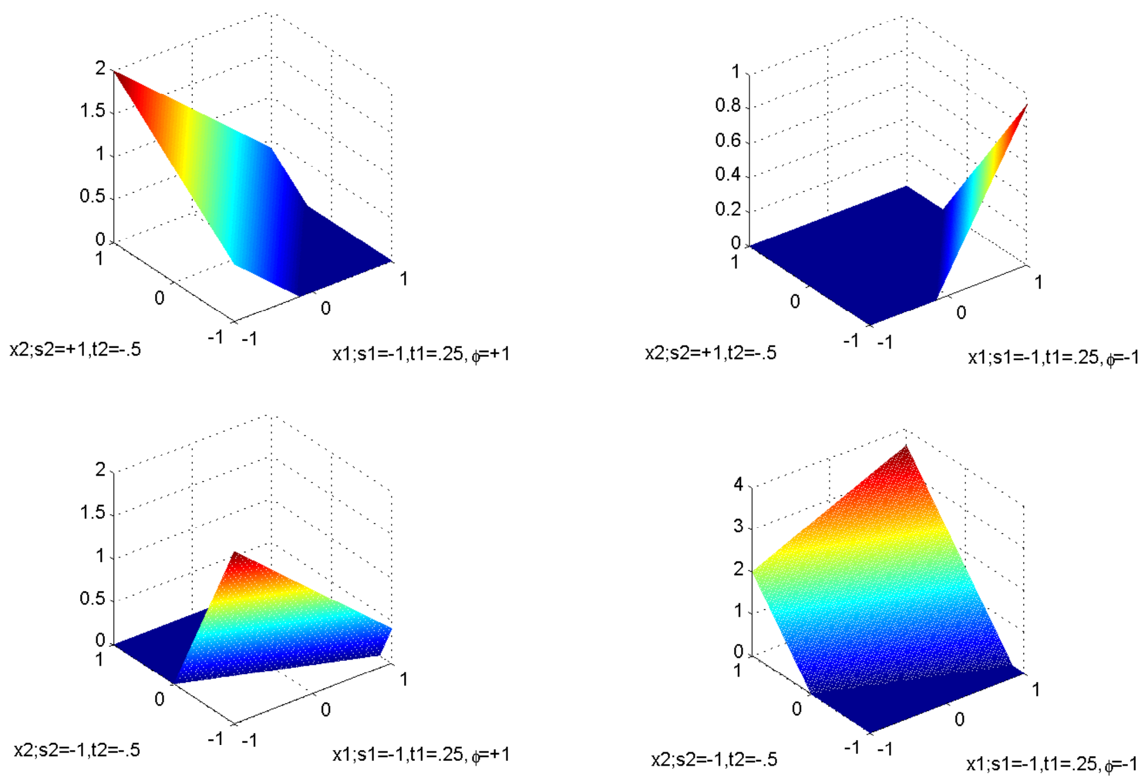

Figure 3: Comparison of candidate pairs of Convex-MARS interaction basis functions.

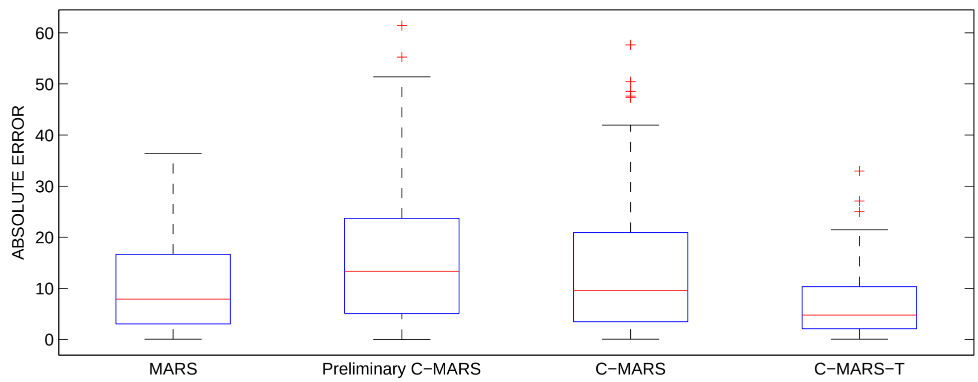

Figure 4: Comparison of boxplots (four-dimensional inventory forecasting problem) based on a validation set of 100 points: (1) MARS, (2) Preliminary Convex-MARS, (3) Convex-MARS, (4) Convex-MARS-T.

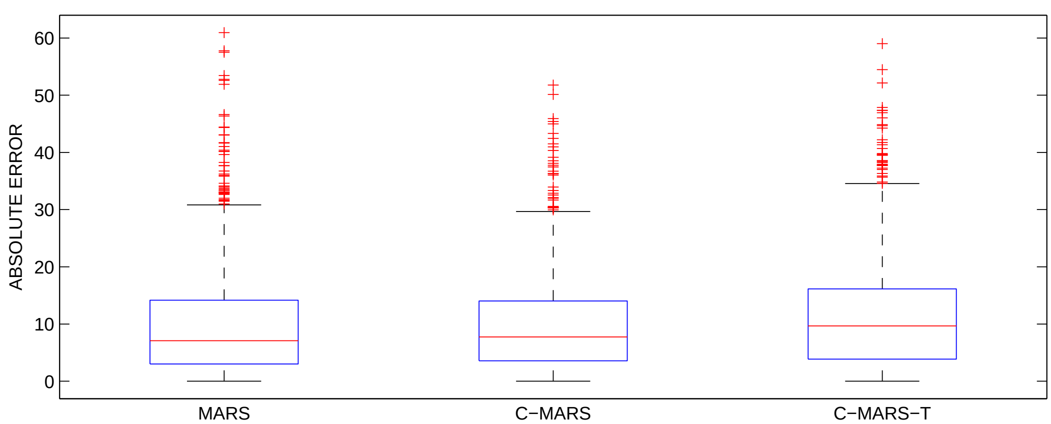

Figure 5: Comparison of boxplots (nine-dimensional inventory forecasting problem) based on a validation set of 1000 points: (1) MARS, (2) Convex-MARS, (3) Convex-MARS-T.