Submitted to *IIE Transactions*
manuscript (Please, provide the mansucript number!)

Global Optimization of Non-convex Piecewise Linear Regression Splines

Nadia Martinez[1], Hadis Anahideh[2], Jay M. Rosenberger[2], Diana Martinez[3], Victoria C. P. Chen[2], Bo Ping Wang[4]

[1]TransSolutions, LLC., 14600 Trinity Blvd, Suite 200, P.O. Box 155486, Fort Worth, TX 76155, nmartinez@transsolutions.com

[2]Department of Industrial and Manufacturing Systems Engineering, University of Texas at Arlington, Arlington, TX 76019, hadis.anahideh@mavs.uta.edu, jrosenbe@uta.edu, vchen@uta.edu

[3]TMAC, 202 E. Border St., Ste. 323, Arlington, TX 76010, diana.martinez@tmac.org

[4]Department of Mechanical and Aerospace Engineering, University of Texas at Arlington, Arlington, TX 76019, bpwang@uta.edu

Multivariate Adaptive Regression Spline (MARS) is a statistical modeling method used to represent a complex system. More recently, a version of MARS was modified to be piecewise linear. This paper presents a mixed integer linear program, called MARSOPT, that optimizes a non-convex piecewise linear MARS model subject to constraints that include both linear regression models and piecewise linear MARS models. MARSOPT is customized for an automotive crash safety system design problem for a major US automaker and solved using branch and bound. The solutions from MARSOPT are compared with those from customized genetic algorithms.

Key words: Global optimization; branch and bound; surrogate methods; multivariate adaptive regression splies; crashworthiness; genetic algorithms.

1. Introduction

Global optimization has been applied to a wide variety of science and engineering design problems. Although numerous global optimization techniques have been developed and studied for decades, when used for complex systems such as the design of an aircraft or an automobile, the results are impractical or not completely satisfactory. One of the major challenges is the computational time required to solve these problems. The majority of design problems require a significant number of experiments or simulations in order to find a globally best solution. However a single simulation can take between seconds and days to finish. For example, to quote Wang and Shan (2007): "it is reported that it takes Ford Motor Company about 36-160 hrs to run one crash simulation (Gu 2001). For a two-variable optimization problem, assuming on average 50 iterations are needed by optimization and assuming each iteration needs one crash simulation, the total computation time would be 75 days to 11 months, which is unacceptable in practice." This example shows the challenges that many real world problems are still facing, especially when dealing with large scale problems.

Although the definition of global optimization is well-defined, there exists an extensive variety of global optimization techniques (Romeijn and Pardalos 1995, Horst et al. 2000) primarily due to different assumptions about the optimization problem. For example, some global optimization researchers develop deterministic algorithms for solving non-convex problems that are capable of guaranteeing convergence. These non-convex optimization problems generally assume some knowledge of the structure of the functions being optimized and typically use branch and bound to find an optimal solution (Keha et al. 2006, Vielma et al. 2008, Sherali and Tuncbilek 1992, Sherali and Wang 2001).

Other researchers assume that the functions being optimized are unknown but can be evaluated with a black box or simulation. Consequently, they develop black-box methods using heuristics, such as evolutionary algorithms. However, the main drawback of methods like evolutionary algorithms and population-based strategies, such as extensively applied genetic algorithms, is that even though they are able to provide some feasible solutions, there is no certainty that the solution found is a globally optimal solution (Mohamed et al. 2012, Peremezhney et al. 2014).

Surrogate optimization approaches, such as response surface methodology and design and analysis of computer experiments, iteratively optimize a surrogate statistical meta-model where the assumption is that the computer experiments or the evaluation of the functions being optimized is extremely expensive. Like other heuristic methods, these techniques only attempt to find good solutions, but they are not provably globally optimal. In addition, the studies presented on surrogate optimization methods, (Emmerich et al. 2002, Willmes et al. 2003, Regis and Shoemaker 2007), among others, do not even globally optimize the surrogate model.

This research addresses two gaps in the surrogate optimization literature (e.g., Emmerich et al. 2002, Willmes et al. 2003, Regis and Shoemaker 2007). One using branch and bound, this paper describes a surrogate optimization method that globally optimizes its surrogate model. Two, unlike other global optimization methods, the developed method assumes a static or fixed set of data; that is, there is no certainty that additional data can be gathered. This inability to collect additional data can occur in real-world problems such as crash simulations or medical treatments, in which experimentation is unavailable. More specifically, this paper describes a non-dominating method to evaluate the quality and the robustness of solutions.

1.1. Literature Review

1.1.1. Non-Convex Piecewise Linear Optimization. Problems involving non-convex piecewise linear functions frequently use branch and bound, which is a widely used deterministic algorithm for solving numerous optimization problems. Keha et al. (2006) proposed a branch-and-cut algorithm without auxiliary binary variables for solving non-convex separable piecewise linear optimization problems that uses cuts and applies SOS2 branching. Vielma et al. (2008) studied an extension of the branch-and-cut algorithm for solving linear problems with continuous separable piecewise linear cost functions developed by Keha et al. (2004) in the case where the cost function is only lower semi-continuous. Sherali and Tuncbilek (1992) proposed a generic branch-and-bound algorithm for globally optimizing continuous polynomial programming problems, which employs constructed linear bounding problems using a reformulation linearization technique (RLT) in concert with a suitable partitioning strategy that guarantees the convergence of the overall algorithm. Sherali and Wang (2001) presented a global optimization approach for solving non-convex factorable programming problems. The approach involves a branch-and-bound procedure with a suitable partitioning scheme and two levels of relaxations, ensuring convergence to a global optimum.

1.1.2. Heuristics. Other optimization techniques incorporate intelligent exploration and exploitation search procedures, such as the well-known group of evolutionary algorithms. Evolutionary techniques are meta-heuristic models that base their algorithms on biological processes.

Examples from the heuristics literature are hill climbing (Rich and Knight 1991), simulated annealing (Kirkpatrick et al. 1983), tabu search (Glover 1977), and genetic algorithms (Holland 1975, Goldberg 1989). More recently, Mohamed et al. (2012) proposed an alternative differential evolution (ADE) algorithm for solving unconstrained global optimization problems, which improves the local search ability and increases the convergence rate. The ADE was tested on a set of well-known high-dimensional unconstrained continuous functions and compared with other differential evolution algorithms, performing better with respect to the search process efficiency, convergence rate, and final solution quality. Peremezhney et al. (2014) proposed a sequential procedure based on a combined application of Gaussian processes, mutual information, and a genetic algorithm to find an approximation to the optimal solution of multi-target optimization of expensive to evaluate functions. The optimal solutions in the Pareto set is selected using the conducted surrogate model's predictions and is evaluated comparing the real system. The proposed approach was compared with a surrogate-based online evolutionary algorithm to show the performance of multi-objective active leaner algorithm using the hypervolume indicator.

1.1.3. Surrogate Optimization A significant number of approaches optimize surrogate approximation models. A surrogate model mimics the original model with a reduced number of simulations and has statistical properties that help develop patterns. Some of the meta-models used in surrogate optimization are: polynomial regression, radial basis functions, Kriging, and multivariate adaptive regression splines, which is the surrogate model used in this study. The following are examples from surrogate optimization literature that involve the use of meta-models. Jones et al. (1998) developed a method called Effcient Global Optimization (EGO) using Kriging as the approximation model, which is especially good at modeling non-linear multimodal functions. In this method the next evaluation point is chosen to be the one that maximizes the expected improvement in the objective function value. Gutmann (2001) introduced a global optimization method based on a general response surface technique. This method uses radial basis functions as interpolants, and a measure of bumpiness is also available. The method was tested in a few numerical examples, showing favorable results in comparison to other global optimization methods. Emmerich et al. (2002) presented the use of metamodels based on Kriging techniques in the context of evolution strategies-based optimization algorithms. Willmes et al. (2003) showed the optimization performance of three well known test functions using evolution strategies assisted by meta-models such as Kriging. Regis and Shoemaker (2007) introduced a stochastic response surface (SRS) method for the global optimization of expensive black-box functions that utilizes radial basis functions. A special case of SRS, called Metric SRS (MSRS), uses a distance criterion when selecting the function evaluation points. A global optimization and a multistart local optimization version of MSRS were developed. Crino and Brown (2007) proposed a global optimization procedure by combining multivariate adaptive regression splines with a response surface methodology. This approach was applied to seven test cases, all of them are low-dimensional examples. Sherali and Ganesan (2003) presented two pseudo-global optimization approaches for solving formidable constrained optimization problems such as the containership design model.

1.2. Contributions

This paper presents a deterministic mixed integer linear program, named MARSOPT, for globally optimizing a modified version of Multivariate Adaptive Regression Splines (MARS) subject to constraints that include both linear regression models and piecewise linear MARS models.

As discussed earlier, this research addresses two gaps in the surrogate optimization and heuristics literature. One, using branch and bound, MARSOPT globally optimizes its surrogate MARS model. Two, this paper presents a non-dominating Pareto evaluation procedure, which validates the quality and robustness of solutions obtained from MARSOPT or other methods, even though no additional experimental data is collected.

Solutions from MARSOPT are compared with those from customized genetic algorithms, which are extensively used in the surrogate optimization literature. The customization of the genetic algorithms involves a rounding approach to incorporate categorical variables. These solutions are compared using the aforementioned Pareto evaluation procedure on a static set of real vehicle crashworthiness data, from a major U.S. automaker, to optimize safety system design. The results show that solving MARSOPT with branch and bound yields substantially better solutions than those from the genetic algorithms, and the CPU time is negligible. Although these experiments use vehicle crashworthiness data, the methods developed in this paper are general and have potential for optimizing numerous complex systems.

The remainder of this paper is organized as follows. Section 2 explains original MARS and the modified piecewise linear version of MARS as the background of this research. Section 3 formulates the new mixed integer linear program MARSOPT. Section 4 shows results comparing solutions from MARSOPT and genetic algorithms on the automotive crash safety system design problem using the Pareto evaluation procedure. Finally, section 5 presents conclusions and future research.

2. Background on Multivariate Adaptive Regression Splines

This section summarizes background research on multivariate adaptive regression splines (MARS), which was introduced by Friedman (1991), and a piecewise linear version of MARS developed by Martinez (2013), Martinez et al. (2015), and Shih (2006). Original MARS, by Friedman (1991), is particularly useful for representing high-dimensional systems involving interactions and curvature. Fitting a MARS model involves a forward-backward stepwise subset selection procedure that builds a model using a set of spline basis functions that best fits the data.

Optimizing MARS has been used for large-scale optimization problems (Siddappa et al. 2007, Pilla et al. 2008). However the MARS models in these cases were assumed to be convex. MARS has also been used as a surrogate model in different optimization approaches, but literature reports its applications only on well-known unconstrained optimization test functions and low-dimensional examples.

The MARS model terms are based on truncated linear functions, where the univariate terms are piecewise linear, and the interaction terms, which are generated by taking products of univariate indicator factors, include nonlinearities.

The MARS approximation has the form:

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

where x is an n-dimensional vector of explanatory variables, β_0 is the intercept coefficient, which is the mean of the response values, M is the maximum number of linearly independent basis functions, β_m is the unknown coefficient for the mth basis function, and $B_m(x)$ is a basis function that utilizes truncated linear functions. The *univariate basis functions* are truncated linear functions of the form $b^+(x;k) = [x-k]_+$ or $b^-(x;k) = [k-x]_+$, where $[q]_+ = \max\{0,q\}$, x is a single explanatory variable, and k is the corresponding univariate knot, where the approximation bends. The *interaction basis functions* are formed as a product of two or more truncated univariate basis functions and is of the following form:

$$B_m(x) = \prod_{l=1}^{L_m} [s_{ml}(x_{v(m,l)} - k_{ml})]_+, \tag{2}$$

where L_m is the number of interaction terms in the mth basis function, $x_{v(m,l)}$ is the explanatory variable corresponding to the lth truncated linear function in the mth basis function, and k_{ml} is the knot value corresponding to $x_{v(m,l)}$. The value s_{ml} is the direction that the truncated linear basis function can take, either +1 or -1.

As mentioned, the univariate terms are piecewise linear, but the interaction terms are not. Therefore to enable use of mixed integer linear programming methods, the interaction terms of a MARS model are transformed to piecewise linear forms, enabling a much easier and faster search of a global optima (Martinez 2013, Martinez et al. 2015, Shih 2006).

As developed in Martinez (2013), Martinez et al. (2015), and Shih (2006), the nonlinearities generated by interaction terms can be modified to a new-one dimensional variable by using the following transformation term z_m:

$$z_m = a_{m0} + \sum_{l=1}^{L_m} a_{ml} x_{v(m,l)}, \tag{3}$$

where

$$a_{m0} = \sum_{l=1}^{L_m} \frac{s_{ml}k_{ml}}{s_{ml}k_{ml} - 1} \qquad a_{ml} = \frac{s_{ml}}{1 - s_{ml}k_{ml}} \tag{4}$$

Using the transformation in equations (3) and (4), the general form of a piecewise linear interaction basis function is then defined as, $B_m(x) = [\phi_m z_m]_+$, where ϕ_m is either +1 or -1 representing the direction of the linear combination of variables.

3. Formulation of MARSOPT

In this section, we develop the MARSOPT mixed integer linear programming formulation for optimizing a non-convex piecewise linear MARS function, subject to a system of piecewise linear MARS function constraints and linear regression constraints.

Consider the following sets, parameters, and variables for MARSOPT. Let C be a set of continuous explanatory variables. For each $j \in C$, let decision variable x_j be the value of the continuous explanatory variable j. For each $j \in C$, let parameters l_j and u_j be lower and upper bounds of variable x_j. Let I be a set of categorical explanatory variables. For each categorical explanatory variables $p \in I$, let K_p be the set of levels of categorical variable p minus a single reference level. For each $p \in I$, $\ell \in K_p$, let decision variable $x_{p\ell}$ a binary such that

$$x_{p\ell} = \begin{cases} 1 & \text{if categorical variable } p \text{ is set to level } \ell, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Let $J = C \bigcup_{p \in I} K_p$ be the set of all continuous and binary explanatory variables.

Let P be a set of piecewise linear MARS models, and let piecewise linear MARS model $o \in P$ be the objective function. Let Q be a set of linear regression models. For each $i \in Q \cup P \setminus \{o\}$, let b_i be a parameter for constraining the maximum value of model i. For each $i \in P$, let M_i be the set of basis functions in piecewise linear MARS model i. For each $i \in P$, $m \in M_i$, let decision variable B_{im} be the value of basis function m in piecewise linear MARS model i as shown in (1). For each $i \in P$, $m \in M_i$, let parameter β_{im} be the coefficient of basis function m in piecewise linear MARS model i as shown in (1). Similarly, let β_{i0} be the intercept coefficient. For each $i \in P$, $m \in M_i$, let decision variable z_{im} be the linear transformation term in (3) for basis function m in piecewise linear MARS model i. For each $i \in P$, $m \in M_i$, let parameter u_{im} be a known upper bound on the absolute value of the linear transformation term z_{im}. For each $i \in P$, $m \in M_i$, let parameter ϕ_{im} be either +1 or -1 representing the direction of the linear transformation term z_{im} in basis function

m in piecewise linear MARS model i. For univariate terms, parameter $\phi_m = +1$. For each $i \in P$, $m \in M_i$, let decision variable y^+_{im} and y^-_{im} be binaries such that

$$y^+_{im} = \begin{cases} 1, & \text{if } z_{im} \geq 0 \\ 0, & \text{otherwise,} \end{cases} \qquad y^-_{im} = \begin{cases} 1, & \text{if } z_{im} \leq 0 \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

In addition, we use the notation $y^{\phi_{im}}_{im}$ to refer to y^+_{im} whenever $\phi_{im} = +1$; otherwise $y^{\phi_{im}}_{im}$ refers to y^-_{im}.

For each $i \in P$, $m \in M_i$, $j \in J$, let parameter a_{imj} be the coefficient of explanatory variable j in the mth basis function in piecewise linear MARS model i as given by (4). Similarly, let a_{im0} be the intercept. To simplify notation, let $a_{imj} = 0$ for variables not in the linear transformation. For univariate basis functions, $a_{imj} = s_{im}$ for the explanatory variable in the basis function, and $a_{im0} = -s_{im}k_{im}$; otherwise $a_{imj} = 0$. For each $i \in Q$, $j \in J$, let parameter β_{ij} be the coefficient of explanatory variable j, and let β_{i0} be the intercept coefficient in linear regression model i. MARSOPT is given by the following:

$$\max \beta_{o0} + \sum_{m \in M_o} \beta_{om} B_{om} \tag{7}$$

$$\text{s. t. } \beta_{i0} + \sum_{m \in M_i} \beta_{im} B_{im} \leq b_i \qquad \forall i \in P \setminus \{o\}, \tag{8}$$

$$\beta_{i0} + \sum_{j \in J} \beta_{ij} x_j, \leq b_i \qquad \forall i \in Q \tag{9}$$

$$a_{im0} + \sum_{j \in J} a_{imj} x_{imj} = z_{im} \qquad \forall i \in P, m \in M_i, \tag{10}$$

$$-u_{im} y^-_{im} \leq z_{im} \leq u_{im} y^+_{im} \qquad \forall i \in P, m \in M_i, \tag{11}$$

$$0 \leq B_{im} \leq u_{im} y^{\phi_{im}}_{im} \qquad \forall i \in P, m \in M_i, \tag{12}$$

$$\phi_{im} z_{im} \leq B_{im} \leq \phi_{im} z_{im} + u_{im} y^{-\phi_{im}}_{im} \qquad \forall i \in P, m \in M_i, \tag{13}$$

$$y^+_{im} + y^-_{im} = 1 \qquad \forall i \in P, m \in M_i, \tag{14}$$

$$y^+_{im}, y^-_{im} \in \{0, 1\} \qquad \forall i \in P, m \in M_i, \tag{15}$$

$$\sum_{\ell \in K_p} x_{p\ell} \leq 1 \qquad \forall p \in I, \tag{16}$$

$$x_{p\ell} \in \{0, 1\} \qquad \forall p \in I, \ell \in K_p, \tag{17}$$

$$l_j \leq x_j \leq u_j, \qquad \forall j \in C. \tag{18}$$

The objective (7) is to maximize a piecewise linear MARS model as developed in Martinez (2013), Martinez et al. (2015), and Shih (2006). Constraint set (8) restricts a set of piecewise linear MARS models, while (9) represents a set of linear regression constraints. Constraint set (10) ensures equation (3) is true for each piecewise linear MARS model. Constraints in (11) , (14), and (15) link the binary variables with the linear transformation variables and guarantee they are defined as in (6). Similarly, constraints (12) and (13) guarantee $B_m(x) = [\phi_m z_m]_+$ as discussed in Section 2. Constraints (16) and (17) ensure that each categorical variable is assigned to at most one level; unassigned categorical variables are assumed to be assigned to the reference level. Finally, constraints in set (18) represent the bounds on the continuous variables.

4. Automotive Crash Safety problem

When an automobile is developed, the safety system design becomes one of the major attributes. Crashworthiness is the ability of a structure to protect its occupants during an impact in such a way that the structure of the vehicle can attenuate the crash force when impact occurs. Multiple crash scenarios need to be analyzed during an automotive crashworthiness study. These scenarios include full front impact, 50% front offset impact, roof crush impact, and side impact.

4.1. Surrogate Optimization Methods for Crashworthiness

Optimizing design is considered computationally intractable due to the significant number of simulations required. Therefore different approximation or surrogate models have been examined for vehicle crashworthiness for occupant safety design. Gu et al. (2001) presented a non-linear response surface-based safety optimization process applied to the vehicle crash safety design of side impact. Yang et al. (2005) studied five response surface methods using a real-world frontal impact design problem as an example. Hamza and Saitou (2005) constructed a new method that utilizes an ensemble of surrogate models constructed from a different sets of samples of finite element analyses to estimate crash performance. A multi-scenario co-evolutionary genetic algorithm was applied to minimize the different aggregates of the outputs of the surrogate models. Liao et al. (2008) proposed a multi-objective optimization procedure for the design of vehicle crashworthiness using simple stepwise regression models. A non-dominated sorting genetic algorithm procedure was employed for searching Pareto efficient solutions. Wang et al. (2010) established a support vector regression in a time-based meta-modeling manner proposed for frontal crash design problems, where the constructed meta-models were optimized using particle swarm optimization (PSO). Song et al. (2012) compared the performance of response surface, Kriging, support vector regression, and radial basis functions for a foam-filled tapered thin-walled structure case. Sequential quadratic programming and PSO were used to search for optimal solutions. More recently, Yin et al. (2014) presented and compared a crashworthiness optimization technique together with a multi-objective particle swarm optimization (MOPSO) algorithm by employing a dynamic ensemble meta-modeling method together with polynomial response surface, radial basis functions, Kriging, and support vector regression.

4.2. Overview of Problem and Formulation

Stepwise regression methods have been commonly used as meta-models to approximate computationally expensive complex systems such as safety related functions in automotive crash analysis, multi-objective optimization for crash safety design of vehicles, frontal impact design problems, and crash safety design of vehicles (Yang et al. 2000, Gu et al. 2001, Yang et al. 2005, Liao et al. 2008). In the following case study, MARSOPT is applied to an automotive crash safety system design example in which the objective is to optimize the crash performance of a vehicle safety system design function subject to constraints and bounds on design variables. Stepwise linear regression (SLR) and piecewise linear MARS (PL-MARS) models are used to approximate the system.

The automotive crash safety system design case study consists of 33 input variables, where 23 of them are continuous, 7 are two-level categorical variables, and 3 are 4-level categorical variables. It includes 51 output variables, one that represents the objective function, which is to be minimized, and 50 that are limited in constraints. Tables 1 and 2 provide lower and upper bounds of the explanatory variables, and the right-hand side (RHS) values of the inequality constraints, respectively.

Table 1 Information on explanatory variables

No.	Description	Lower Bound	Upper Bound	Variable Type
1	PAB Shape	1	4	4-level categorical
2	PAB Size	-0.2	1.0	continuous
3	Buckle pretensioner flag	0	1	2-level categorical
4	Retractor pretensioner flag	0	1	2-level categorical
5	Adaptive belt load limiter flag	0	1	2-level categorical
6	Crash locking tongue flag	0	1	2-level categorical
7	Knee airbag flag	0	1	2-level categorical
8	Passenger airbag adapt vent flag	0	1	2-level categorical
9	Heel stopper flag	0	0	2-level categorical
10	Buckle pretensioner pull in (m)	0.06	0.1	continuous
11	Buckle pretensioner time to fire (s)	0.008	0.013	continuous
12	Retractor pretensioner pull in (m)	0.06	0.1	continuous
13	Retractor pretensioner time to fire (s)	0.008	0.013	continuous
14	Retractor torsion bar force level-1	2000	3000	continuous
15	Retractor torsion bar force level-2	2000	3200	continuous
16	Retractor torsion bar dispalcement interval-05	0.05	0.3	continuous
17	Retractor torsion bar dispalcement interval-50	0.05	0.3	continuous
18	Knee airbag time to fire (s)	0.013	0.2	continuous
19	Knee airbag inflator power	0.75	1.5	continuous
20	Knee airbag vent size (mm)	0	15	continuous
21	Passenger airbag lower tether length (mm)	0.4	0.52	continuous
22	Passenger airbag lower tether location	1	4	4-level categorical
23	Passenger airbag time to fire (s)-u05	0.01	0.013	continuous
24	Passenger airbag Z-Scale	0.8	1.2	continuous
25	Passenger airbag adaptive vent size (mm)	40	120	continuous
26	Passenger airbag time to fire (s)-b05	0.01	0.1	continuous
27	Passenger airbag time to fire (s)-b50	0.01	0.1	continuous
28	Passenger airbag time to fire (s)-u05	0.02	0.1	continuous
29	Passenger airbag time to fire (s)-u50	0.02	0.1	continuous
30	Passenger airbag fixed vent size (mm)	40	80	continuous
31	Passenger airbag inflator power	0.8	1.2	continuous
32	Passenger airbag upper tether length (mm)	0.4	0.52	continuous
33	Passenger airbag upper tether location	1	4	4-level categorical

The case study also includes two sets of data. Data Set 1 contains 200 points and is used to build the SLR and the PL-MARS system models, while Data Set 2 has 1249 points. To develop the SLR and the PL-MARS models, the variables were scaled values based on the mid-range and the half-range of the set of data variable values. For the 10 categorical explanatory variables, a reference level was selected, and for each of the remaining levels, a binary variable was created. Consequently, the seven variables with two levels were treated as binary variables, while three binary variables were used for categorical variable 1 that has four levels. Although categorical variables 22 and 33 also have four levels, Data Set 1 includes no observations for one of the four levels, so only two binary

Table 2 Information on the objective function and constraints

No.	**Name**	**Objective /RHS**	**No.**	**Name**	**RHS**
	Obj-pb05-RRS	Minimize			
1	constr-far50-ChestD	≤ 1	26	constr-pb50-HIC	≤ 1
2	constr-far50-ChestG	≤ 1	27	constr-pb50-Head-IP-min	≥ 1
3	constr-far50-Chest-IP-min	≥ 1	28	constr-pb50-NeckFzMax	≤ 1
4	constr-far50-FemurL	≤ 1	29	constr-pb50-NeckFzMin	≤ 1
5	constr-far50-FemurR	≤ 1	30	constr-pb50-Nij	≤ 1
6	constr-far50-HIC	≤ 1	31	constr-pu05-ChestD	≤ 1
7	constr-far50-Head-IP-min	≥ 1	32	constr-pu05-ChestG	≤ 1
8	constr-far50-NeckFzMax	≤ 1	33	constr-pu05-Chest-IP-min	≥ 1
9	constr-far50-NeckFzMin	≤ 1	34	constr-pu05-FemurL	≤ 1
10	constr-far50-Nij	≤ 1	35	constr-pu05-FemurR	≤ 1
11	constr-pb05-ChestD	≤ 1	36	constr-pu05-HIC	≤ 1
12	constr-pb05-ChestG	≤ 1	37	constr-pu05-Head-IP-min	≥ 1
13	constr-pb05-Chest-IP-min	≥ 1	38	constr-pu05-NeckFzMax	≤ 1
14	constr-pb05-FemurL	≤ 1	39	constr-pu05-NeckFzMin	≤ 1
15	constr-pb05-FemurR	≤ 1	40	constr-pu05-Nij	≤ 1
16	constr-pb05-HIC	≤ 1	41	constr-pu50-ChestD	≤ 1
17	constr-pb05-Head-IP-min	≥ 1	42	constr-pu50-ChestG	≤ 1
18	constr-pb05-NeckFzMax	≤ 1	43	constr-pu50-Chest-IP-min	≥ 1
19	constr-pb05-NeckFzMin	≤ 1	44	constr-pu50-FemurL	≤ 1
20	constr-pb05-Nij	≤ 1	45	constr-pu50-FemurR	≤ 1
21	constr-pb50-ChestD	≤ 1	46	constr-pu50-HIC	≤ 1
22	constr-pb50-ChestG	≤ 1	47	constr-pu50-Head-IP-min	≥ 1
23	constr-pb50-Chest-IP-min	≥ 1	48	constr-pu50-NeckFzMax	≤ 1
24	constr-pb50-FemurL	≤ 1	49	constr-pu50-NeckFzMin	≤ 1
25	constr-pb50-FemurR	≤ 1	50	constr-pu50-Nij	≤ 1

variables were used for them. Incorporating these binary variables, the number of explanatory variables is 37 in which 14 of them are binary variables. To maintain consistency with the scaling of the other variables, the binary variables use levels $\{-1, 1\}$, instead of $\{0, 1\}$.

The objective function response variable was fit using an SLR model and a PL-MARS model, resulting in coefficients of determination R^2 equal to 0.77 and 0.90, respectively. The higher R^2 for the PL-MARS model indicates that the objective fits the PL-MARS model better than the SLR model. SLR models were also constructed for the output variables of the 50 constraints. Of these 50 SLR models, 10 either show curvature in residual plots or have R^2 less than 0.70, indicating that the SLR models do not fit the data well for these response variables. Consequently, PL-MARS approximations were fit for them. The PL-MARS functions were restricted to up to two-way interaction terms. The number of basis functions for each of the 10 PL-MARS models varies from 6 to 10. Additional details on the SLR and PL-MARS models are in Martinez (2013).

Since the underlying function is unknown, two MARSOPT models were formulated, one with the SLR objective function and one with the PL-MARS objective function. To account for the $\{-1,1\}$ scalarization of the binary variables, MARSOPT used a continuous variable $x'_{p\ell}$, $\forall \ell \in K_p$, $p \in I$ and the following set of linking constraints

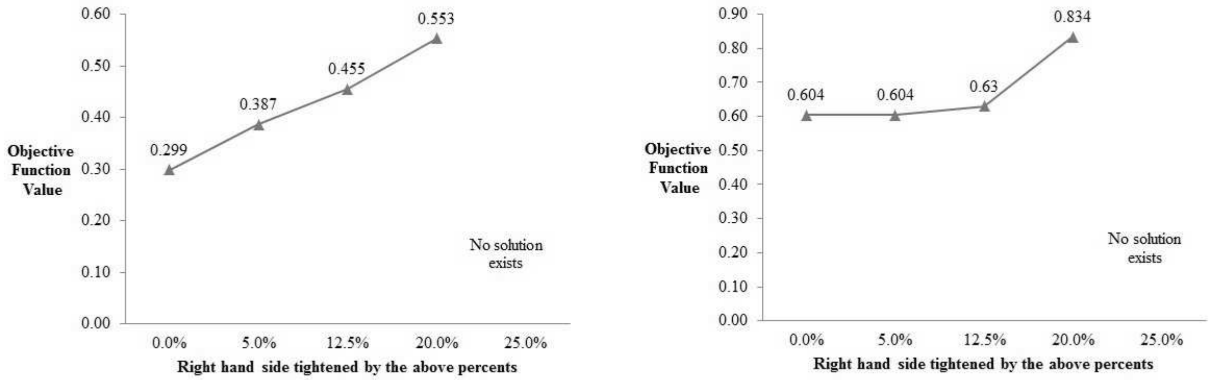
$$x'_{p\ell} = 2x_{p\ell} - 1 \qquad \forall \ell \in K_p, p \in I. \tag{19}$$

The variable $x'_{p\ell}$ was then used in the system models in constraints (9) and (10), while $x_{p\ell}$ was used in constraint set (16). Furthemore, since the SLR and PL-MARS system models were developed on scaled data, MARSOPT yields scaled solutions. Consequently, these solutions were unscaled back to the units of the original data.

4.3. Results and Sensitivity Analysis

The two MARSOPT models described in Section 4.2 were generated by a C-programming code and solved by branch and bound using IBM CPLEX on a Dual 2.6 GHz Athlon workstation. The solutions of these models are presented in Appendix A. The CPU times taken to solve the SLR MILP objective model and the PL-MARS MILP objective model were 0.02 seconds and 0.07 seconds, respectively, indicating that CPU time required to solve MARSOPT was not a concern for this case study.

In the solutions of these MARSOPT models, some of the output variables in the constraints are binding; that is, they are equal to their RHS b values. However, the SLR and PL-MARS models in the constraints are imperfect and include error, so it is possible that the solutions from MARSOPT violate constraints. Consequently, alternative MARSOPT models in which the RHS b values are tightened were solved to yield more robust solutions. *Tightening constraints* by γ indicates that b_i is decreased to $(1-\gamma)b_i$ for all less-than-or-equal-to constraints and increased to $b_i/(1-\gamma)$ for all greater-than-or-equal-to constraints. Figures 1a and 1b show the objective values, using both MARSOPT models, when tightening the constraints by $\gamma = 5\%$, 12.5%, and 20%.



(a) Sensitivity analysis using SLR model (b) Sensitivity analysis using PL-MARS model

Figure 1 Sensitivity analysis for the objective function

For MARSOPT with the PL-MARS objective, the objective function value depreciates slowly as the constraints are tightened. No solution to the problem exists when the constraints are tightened by 25%.

4.4. Genetic Algorithms Comparison

This section develops customized genetic algorithms, which are prevalent optimization techniques for surrogate optimization of vehicle crashworthiness design (Hamza and Saitou 2004, 2005, D. Aspenberg and Nilsson 2013). A genetic algorithm (GA) is a heuristic search method based on the principles of life and natural selection. A GA encodes the decision variables in an initial set of candidate solutions within a population, which are also called genotypes, individuals, members, or chromosomes. A *chromosome* is made of genes that hold information and control the inheritance of certain traits affecting future offspring. In a GA, these chromosomes are represented by a string of variables that have feature values. Once an initial population is randomly created and evaluated by a single performance measure called *fitness* or an evaluation function, the population starts evolving by iteratively selecting solutions and creating new generations until the process reaches a defined maximum number of generations (MAXGENS). This selection is based on the fitness measure of individuals and two genetic operators, *crossover* and *mutation*. In addition, GA's use other control parameters, such as population and generation size and the encoding of chromosomes.

There are many GA variations with many possible combinations to set parameters. While there is no universal best method to set such parameters for any problem, this study is limited to a particular simple GA, which was built by Denis Cormier (North Carolina State University) and modified by Sita S. Raghavan (University of North Carolina at Charlotte). This simple GA assumes there is no distinction between the fitness of an individual and the objective value. It uses proportional selection, one-point crossover, uniform mutations, and includes a routine called "elitist," which ensures that the best chromosomes are retained between generations. This GA code is available from Michalewicz (1996), and the corresponding pseudo code is presented in Algorithm 1.

Algorithm 1 Genetic Algorithm pseudo code by Cormier and Raghavan Michalewicz (1996)

INITIALIZE: initializes the genes within the variables bounds

EVALUATE: implements the user-defined evaluation function for fitness

KEEP THE BEST: keeps track of the best member of the population

generation = 1

while generation < MAXGENS **do**

  SELECTOR performs standard proportional selection

  CROSSOVER: selects two parents for a single point crossover

  MUTATE: performs a random uniform mutation

  EVALUATE: implements the user-defined evaluation function for fitness

  ELITIST: stores the best member of the previous generation

  generation = generation + 1

end while

The GA searches for chromosomes within a continuous hypercube, so for the case study, the chromosomes represent strings of the explanatory variables within a 37-dimensional $[-1,1]$ continuous hypercube. Within the EVALUATE function of Algorithm 1, the fitness of each chromosome is calculated in two steps. The first step maps the chromosome to a *rounded solution* that satisfies the binary restrictions on the 14 binary variables as well as the three constraints in the set (16).

The second step is to calculate the fitness as the objective value (7) minus a penalty for violations of constraints in sets (8) and (9).

Specifically, consider a candidate solution in the population $x_j \in [-1,1]$, for each variable $j \in J$. In the rounded solution $\overline{x}$, $\overline{x}_j = x_j$, $\forall j \in C$, and for each $\ell \in K_p, p \in I$, $\overline{x}_{p\ell}$ is given by (20).

$$\overline{x}_{p\ell} = \begin{cases} 1 & \text{if } \ell \in \arg\max_{\tilde{\ell} \in K_p} \left\{ x_{p\tilde{\ell}} \right\} \text{ and } x_{p\ell} \geq \frac{2}{|K_p| \sqrt{|K_p|+1}} - 1 \\ -1 & \text{otherwise} \end{cases} \tag{20}$$

In the rounded solution $\overline{x}$, the constraints in the set (16) are implicit. In addition, it can be shown that if each $x_{p\ell}$ is randomly sampled from a continuous uniform distribution over the interval $[-1,1]$, then the probability that the rounded value of $\overline{x}_{p\ell} = 1$ is $1/(|K_p|+1)$. Furthermore, the probability that $\overline{x}_{p\ell} = -1$, $\forall \ell \in K_p$, is similarly $1/(|K_p|+1)$, which implies that selecting the reference level is equally likely.

Using the rounded solution $\overline{x}$, the fitness of the chromosome is calculated as the objective value minus a user-defined penalty on the violation of constraints. Specifically, for each $i \in Q \cup P$, let $\hat{g}_i(x)$ be the system model, and let δ be the user-defined penalty. The fitness is calculated as by (21):

$$\hat{g}_o(\overline{x}) - \delta \sum_{i \in Q \cup P \setminus \{o\}} \left[\hat{g}_i(\overline{x}) - b_i \right]_+ . \tag{21}$$

For the remainder of this paper, the customized GA is now referred to as: PL-MARS-GA.

Using the GA presented in Algorithm 1, the PL-MARS-GA runs were performed using the two GA parameter settings presented in Table 3.

Table 3 GA parameter settings

Parameters	**G (Grefenstette 1986)**	**C&R (Michalewicz 1996)**
Population size	30	50
Maximum number of generations	300	1000
Probability of crossover	0.9	0.8
Probability of mutation	0.01	0.15

Grefenstette (1986) conducted experiments for searching and determining optimal control parameters for a class of global optimization procedures, suggesting the values shown in second column.

The third column shows the set of parameter settings used for the simple GA proposed by Cormier and Raghaven (C&R), (Michalewicz 1996).

Ten trials for each of the two objective function system models were performed using the parameters shown in Table 3 (5 with G-settings and 5 with C&R-settings). Different penalties (δ) were applied to each run. These trials were then run tightening the constraints by 0%, 5%, 12.5%, and 20%, respectively. A C-programming code executed on a Dual 2.6 GHz Athlon workstation was used to generate 130 PL-MARS-GA solutions, and the CPU time for all 130 executions were less than 2 seconds. However, within the set of PL-MARS-GA solutions, none of them optimized the PL-MARS model. In general, solutions from PL-MARS-GA in which the penalty $\delta < 5$ had many violated constraints. The C&R-settings show better results than the G-settings. The PL-MARS-GA algorithm found only one feasible solution when the constraints were tightened by 20%.

A clear disadvantage of an evolutionary algorithm is that there is no certainty that the solution found is an optimal solution, which is the case for PL-MARS-GA. The PL-MARS-GA algorithm using the PL-MARS objective model with $\delta \geq 5$ found solutions with an average objective value of 0.65964 (3% worse than that of MARSOPT) and a minimum of 0.61651 (2% worse than that of MARSOPT). Similarly, PL-MARS-GA using the SLR objective model with $\delta \geq 5$ found solutions with an average objective value of 0.46433 (55% worse than that of MARSOPT) and a minimum of 0.38750 (26% worse than that of MARSOPT).

4.5. Evaluation Procedure

In this research, we had no access to a crash simulator to collect addition data. Consequently, we developed an alternative method to evaluate solutions from different sources. As described earlier, we generated 8 MARSOPT solutions (Section 4.3), 130 PL-MARS-GA solutions (Section 4.5), and we had a total of 1449 solutions from the original data sets (200 from Data Set 1 and 1249 from Data Set 2), for a total of 1587 solutions. The evaluation procedure uses nine methods to evaluate a solution and construct a Pareto optimal frontier. Six of the evaluators are related to the objective function values, and the other three consider the feasibility of a solution.

To construct the evaluators, we first develop 99% confidence bands $(\hat{g}_i^l(x), \hat{g}_i^u(x))$ on each system model $i \in Q \cup P$, including both the SLR and PL-MARS objective models, based upon the Working-Hotelling method (Kutner 1974). The six objective evaluators include the expected objective value $\hat{g}_o(x)$ and the lower and upper bands on the objective value $(\hat{g}_o^l(x), \hat{g}_o^u(x))$ for both the SLR and PL-MARS objective models. The intuition is that a solution with a truly good objective value should perform well on all six of these evaluators. For each constraint $i \in Q \cup P \setminus \{o\}$, we say that i has *no violation* if $b_i \geq \hat{g}_i^u(x)$, i has a *possible violation* if $b_i < \hat{g}_i^u(x)$, i has an *expected violation* if $b_i < \hat{g}_i(x)$, and i has a *confident violation* if $b_i < \hat{g}_i^l(x)$. The three feasibility objective evaluators are the number of possible violations, expected violations, and confident violations in the constraints.

After calculating the evaluators for each of the 1587 solutions, we constructed a Pareto efficient frontier minimizing them. The solutions on the Pareto efficient frontier are nondominated; that is, for each solution on the Pareto efficient frontier, there is no solution within the 1587 solutions that is superior in all nine evaluators.

Table 4 includes the 15 nondominated solutions on the Pareto efficient frontier. Using these 15 solutions, a decision maker could select a single solution based upon personal preferences of the evaluators. One reasonable decision would be to create a *reduced Pareto efficient frontier* by eliminating solutions with at least one confident violation. This reduced Pareto efficient frontier includes only eight solutions (Solutions 2-6, 11, 14, and 15 on Table 4). Of the seven eliminated solutions, six of them (Solutions 1, 7, 9, 10, 12, and 13) were found using a genetic algorithm, and one of them (Solution 8) is the only point on the original Pareto efficient frontier from Data Set 2.

A decision maker who is highly sensitive to potential constraint violations would likely choose a solution from one of the four solutions (Solutions 3, 4, 11, and 14) on the Pareto efficient frontier that have no violations. Of these four solutions, two of them (Solutions 3 and 14) were found using the SLR objective model and 5% tightening on the constraints. Solution 3 though was found using a genetic algorithm, while Solution 14 was found using MARSOPT. Similarly, Solution 11 was also found using a genetic algorithm but instead of tightening the constraints by 5%, its algorithm had $\delta = 10$, which is 100 times larger than $\delta = 0.1$ in the algorithm that found Solution 11. The remaining solution with no violations (Solution 4) is from Data Set 1. Of these solutions with no violations, Solution 4 from Data Set 1 and Solution 14 found using MARSOPT have optimal and near optimal PL-MARS objective values, respectively.

Two solutions (Solutions 2 and 11) are only superior to other solutions (Solutions 15 and 14, respectively) in the upper band on the SLR objective model. A decision maker would not likely consider Solution 2 over Solution 15, due to Solution 2 having an expected violation. Similarly, the PL-MARS objective of Solution 11 makes it less attractive than Solution 14.

The reduced Pareto efficient frontier includes three of the eight MARSOPT solutions (37.5%), three of the 130 PL-MARS-GA solutions (less than 3%), two of the 200 solutions (1%) from Data Set 1, and zero from Data Set 2. Consequently, we conclude that MARSOPT was very efficient at finding high-quality feasible solutions in the case study. In addition, Table 5 shows how the other five solutions found using MARSOPT are dominated. Of these five, four of them (Solutions 16-19) are dominated by Solution 14, and one of them (Solution 20) is dominated by Solution 15, both of which were found using MARSOPT. In fact, Solution 14 is particularly attractive because it has no violations, a near optimal PL-MARS objective value, and a very low SLR objective value.

Table 4 Efficient Pareto frontier points

Solutions					Evaluators								
					Objective Values						Constraint Violations		
Solution	**Objective Model**	**Algorithm**	**Tightening of RHS**	**GA (Settings,δ)**	**SLR-** $\hat{g}_o^l(x)$	**SLR**	**SLR-** $\hat{g}_o^u(x)$	**PL-MARS-** $\hat{g}_o^l(x)$	**PL-MARS**	**PL-MARS-** $\hat{g}_o^u(x)$	**Possible**	**Expected**	**Confident**
1	SLR	PL-MARS-GA	0%	(G,0.01)	0.250	0.283	0.316	0.965	0.977	0.989	5	4	3
2		Data Set 1	-	(-,-)	0.544	0.569	0.594	0.601	0.604	0.607	1	1	0
3	SLR	PL-MARS-GA	5%	(C&R,0.1)	0.323	0.353	0.382	0.735	0.74	0.745	0	0	0
4		Data Set 1	-	(-,-)	0.547	0.571	0.596	0.601	0.604	0.607	0	0	0
5	SLR	MARSOPT	0%	(-,-)	0.266	0.299	0.332	0.770	0.777	0.783	5	0	0
6	SLR	PL-MARS-GA	0%	(C&R,0.2)	0.331	0.359	0.387	0.731	0.736	0.741	3	1	0
7	SLR	PL-MARS-GA	0%	(C&R,0.05)	0.250	0.281	0.313	0.962	0.974	0.986	6	4	2
8		Data Set 2	-	(-,-)	0.516	0.534	0.553	0.601	0.604	0.607	1	1	1
9	SLR	PL-MARS-GA	5%	(C&R,0.05)	0.225	0.257	0.288	0.953	0.965	0.976	5	5	3
10	SLR	PL-MARS-GA	0%	(C&R,0.01)	0.225	0.257	0.290	0.977	0.99	1.002	6	4	3
11	SLR	PL-MARS-GA	0%	(C&R,10)	0.362	0.388	0.413	0.680	0.684	0.688	0	0	0
12	PL-MARS	PL-MARS-GA	5%	(G,0.05)	0.313	0.344	0.375	0.919	0.929	0.939	4	4	2
13	SLR	PL-MARS-GA	5%	(C&R,0.01)	0.208	0.239	0.270	0.953	0.966	0.978	5	5	4
14	SLR	MARSOPT	5%	(-,-)	0.355	0.387	0.419	0.608	0.612	0.615	0	0	0
15	PL-MARS	MARSOPT	5%	(-,-)	0.525	0.565	0.605	0.601	0.604	0.607	1	0	0

Table 5 Dominated MARSOPT Solutions

Solutions				Evaluators								
				Objective Values						Constraint Violations		
Solution	**Objective Model**	**Algorithm**	**Tightening of RHS**	**SLR-** $\hat{g}_o^l(x)$	**SLR**	**SLR-** $\hat{g}_o^u(x)$	**PL-MARS-** $\hat{g}_o^l(x)$	**PL-MARS**	**PL-MARS-** $\hat{g}_o^u(x)$	**Possible**	**Expected**	**Confident**
Solution:												
16	SLR	MARSOPT	12.5%	0.422	0.455	0.488	0.766	0.773	0.779	0	0	0
17	SLR	MARSOPT	20%	0.521	0.553	0.584	0.983	0.994	1.005	0	0	0
18	PL-MARS	MARSOPT	12.5%	0.709	0.745	0.782	0.627	0.63	0.633	0	0	0
19	PL-MARS	MARSOPT	20%	0.787	0.824	0.86	0.829	0.834	0.839	0	0	0
Dominated by:												
14	SLR	MARSOPT	5%	0.355	0.387	0.419	0.608	0.612	0.615	0	0	0
Solution:												
20	PL-MARS	MARSOPT	0%	0.658	0.7	0.741	0.601	0.604	0.607	3	0	0
Dominated by:												
15	PL-MARS	MARSOPT	5%	0.525	0.565	0.605	0.601	0.604	0.607	1	0	0

Like Solution 14 from MARSOPT, Solution 4 from Data Set 1 is also attractive. In addition to having no violations and an optimal PL-MARS objective value, it had the smallest true objective value (0.57309) in Data Set 1. Observe its PL-MARS objective value (0.60433) and SLR objective value (0.57114) are quite close to the true objective value. Unscaled solutions for Solutions 5, 14, and 15 found using MARSOPT and Solution 4 are given in Table B1 in Appendix B.

Finally, Table 6 shows the safety star ratings based upon the objective Relative Risk Score (RRS). Of the eight solutions found using MARSOPT, four of them (Solutions 14, 15, 18, and 20)

Table 6 Relative Risk Score Star Rating

RRS $\leq$ 0.67	5 Stars
0.67$\leq$ RRS $<$ 1.33	4 Stars
1.33$\leq$ RRS $<$ 2.00	3 Stars
2.00$\leq$ RRS $<$ 2.67	2 Stars
RRS $>$ 2.67	1 Star

have 5-star ratings based on the PL-MARS objective model, five of them (Solutions 5, 14, 15, 16, and 17) have 5-star ratings according to the SLR objective model, and two of them (Solutions 14 and 15), which were both Pareto efficient, have star ratings according to both objective models. Only one Solution 19 does not have a 5-star rating based upon either objective model. No solution from MARSOPT had less than a 4-star rating for any objective model.

5. Conclusions

This research presented a new mixed integer linear programming to optimize piecewise linear functions generated by a modified version of multivariate adaptive regression splines (MARS), subject to both linear and piecewise linear MARS constraints. The method is computationally fast and is also capable of handling non-convexity, non-linearity, and allows for continuous and categorical decision variables.

MARSOPT was applied to a case study problem to optimize the crash performance of a vehicle safety system design that consisted of 33 design variables and 51 output variables. The method was able to globally optimize the surrogate models representing the search space of the problem, where SLR and PL-MARS models were approximated. These meta-models were built from a relatively small set of design variables. By tightening the constraints, MARSOPT effectively provided more robust designs with very small objective deterioration. MARSOPT was compared to customized genetic algorithms, which used penalties to minimize violations of constraints on the output variables. Although this evolutionary algorithm was able to provide feasible solutions, it was unable to optimize the surrogate model.

A Pareto evaluation procedure based on nine evaluators compared solutions found using MARSOPT, the customized genetic algorithms, and solutions from the original data. These nine evaluators included confidence bands over the approximated objective functions and intervals for possible, expected, and confident violations in the constraints. By eliminating solutions with confident violations, only eight solutions resulted on the efficient frontier, including three of the eight solutions found using MARSOPT. A relative risk score, with a 5-star rating as the highest, showed that all eight solutions from MARSOPT had a 4-star rating or better, and seven of the eight solutions achieved a 5-star rating based upon at least one model.

Acknowledgments

This research was partially supported by National Science Foundation Award CMMI–1434401.

References

Crino, Scott, Donald E. Brown. 2007. Global optimization with multivariate adaptive regression splines. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* **37**(2).

D. Aspenberg, J. Jergues, L. Nilsson. 2013. Robust optimization of front members in a full frontal car impact. *Engineering Optimization* **45** 245–264.

Emmerich, M., A. Giotis, M. Ozdemir, T. Back, K. Giannakoglou. 2002. Metamodel assisted evolution strategies. *Parallel Problem Solving from Nature V (PPSN VII), Springer-Verlag* 362–370.

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

Glover, F. 1977. Heuristics for integer programming using surrogate constraints. *Decision Sciences* **8** 156–166.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley Publication Co., Reading, M.A.

Grefenstette, John. J. 1986. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics* **16** 122–128.

Gu, L. 2001. A comparison of polynomial based regression models in vehicle safety analysis. *ASME Design Engineering Technical Conferences - Design Automation* .

Gu, L., R.J. Yang, C.H. Tho, M. Makowskit, O. Faruquet, Y.Li. 2001. Optimization and robustness for crashworthiness of side impact. *International Journal of Vehicle Design* **26**(4) 348–360.

Gutmann, H.-M. 2001. A radial basis function method for global optimization. *Journal of Global Optimization* **19**(3) 201–227.

Hamza, K., K. Saitou. 2004. Crashworthiness design using meta-models for approximating the response of structural members. *University Conference on Mechanical Design and Production* .

Hamza, K., K. Saitou. 2005. Vehicle crashworthiness design via a surrogate model ensemble and a co-evolutionary genetic algorithm. *International Design Engineering Technical conferences and Computers and Information in Engineering Conference* 1–9.

Holland, J. H. 1975. Adaptation in natural and artiffcial systems .

Horst, R., P. M. Pardalos, N.V. Thoai. 2000. *Introduction to Global Optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Jones, Donald R., Matthias Schonlau, William J. Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* **13** 455–492.

Keha, A. B., I. R. de Farias, G. L. Nemhauser. 2004. Models for representing piecewise linear cost functions. *Operations Research Letters* **32** 44–48.

Keha, A. B., I. R. de Farias, G. L. Nemhauser. 2006. A branch-and-cut algorithm without binary variables for nonconvex piecewise linear optimization. *Operations Research* **54** 847–858.

Kirkpatrick, S., C.D. Gelatt, M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* **220** 671–680.

Kutner, Michael H. 1974. Applied linear statistical model. *McGraw-Hill/Irwin* .

Liao, Xingtao, Li Qing, Yang Xujing, Zhang Weigang, Li Wei. 2008. Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Structural and Multidisciplinary Optimization* **35**(6) 561–569.

Martinez, Diana. 2013. Variants of multivariate adaptive regression splines (mars): Convex vs. non-convex, piecewise-linear vs. smooth and sequential algorithms. Ph.D. thesis, The University of Texas at Arlington.

Martinez, Diana, Thomas Dachuan Shih, V. C. P. Chen, S. B. Kim. 2015. A convex version of multivariate adaptive regression splines. *Computational Statistics and Data Analysis* **81** 89–106.

Michalewicz, Zbigniew. 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Third, revised and extended edition.

Mohamed, Ali W., Hegazy Z. Sabry, Motaz Khorshid. 2012. An alternative differential evolution algorithm for global optimization. *Journal of Advanced Research* **3** 149–165.

Peremezhney, N., E. Hines, A. Lapkin, C. Connaughton. 2014. Combining gaussian processes, mutual information and a genetic algorithm for multi-target optimization of expensive-to-evaluate functions. *Engineering Optimization* **46(11)** 1593–1607.

Pilla, V. L., J. M. Rosenberger, V. C. P. Chen, B. C. Smith. 2008. A statistical computer experiments approach to airline fleet assignment. *IIE Transactions* .

Regis, Rommel G., Christine A. Shoemaker. 2007. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing* **19**(4) 497–509.

Rich, E., K. Knight. 1991. *Artificial Intelligence*. McGraw-Hill.

Romeijn, H. E., P. M. Pardalos. 1995. *Handbook of Global Optimization*, vol. 2. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Sherali, H. D., V. Ganesan. 2003. A pseudo-global optimization approach with application to the design of containerships. *Journal of Global Optimization* **26** 335–360.

Sherali, H. D., C. H. Tuncbilek. 1992. A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. *Journal of Global Optimization* **2** 101–112.

Sherali, H. D., H. Wang. 2001. Global optimization of non-convex factorable programming problems. *Math. Program., Ser. A* **89** 459–478.

Shih, Thomas Dachuan. 2006. Convex versions of multivariate adaptive regression splines and implementation for complex optimization problems. *PhD thesis, Department of Industrial and System Engineering, UT Arlington* .

Siddappa, S., D. Gnther, J.M. Rosenberger, V. C. P. Chen. 2007. Refined experimental design and regression splines method for network revenue management. *Journal of Pricing and Revenue Management* **6**(3) 188–199.

Song, X., G. Li, W. Gao, Q. Li. 2012. Crashworthiness optimization of foam-filled tapered thin-walled structure using multiple surrogate models. *Structural and Multidisciplinary Optimization* **47(2)** 221–231.

Vielma, J.P., A.B. Keha, G.L. Nemhauser. 2008. Nonconvex, lower semi-continuous piecewise linear optimization. *Discrete Optimization* **5** 467–488.

Wang, GG., S. Shan. 2007. Review of metamodeling techniques in support of engineering design optimization. *ASME Transactions, Journal of Mechanical Design* **129**(4) 370–380.

Wang, H., G. Li, E. Li. 2010. Time-based metamodeling technique for vehicle crashworthiness optimization. *Computer Methods in Applied Mechanics and Engineering* **199** 37–40.

Willmes, L., T. Back, Y. Jin, B. Sendhoff. 2003. Comparing neural networks and kriging for fitness approximation in evolutionary optimization. *IEEE Congress on Evolutionary Computation* 663–670.

Yang, R. J., L. Gu, L. Liaw, C. Gearhart, C. H. Tho, X. Liu, B. P. Wang. 2000. Approximations for safety optimization of large systems. *ASME 2000 Design Engineering Technical Conferences - Design Automation Conference* .

Yang, R. J., N. Wang, C. H. Tho, J. P. Bobineau, B. P. Wang. 2005. Metamodeling development for vehicle frontal impact simulation. *Journal of Mechanical Design* **127**(5) 1014–1020.

Yin, H., G. Wen, H. Fang, Q. Qing, X. Kong, J. Xiao, Z. Liu. 2014. Multiobjective crashworthiness optimization design of functionally graded foam-filled tapered tube based on dynamic ensemble metamodel. *Materials and Design* **55** 747–757.

Appendix

Appendix A

Table A1 displays the scaled and unscaled solutions found with MARSOPT using both the SLR model and the PL-MARS model for the objective function, while Table A2 reports the objective values and the output variables (left-hand sides of the constraints) for both solutions.

Table A1 Scaled and unscaled solutions obtained from MARSOPT using SLR model and PL-MARS model

SLR model solution				PL-MARS model solution			
ID	**Scaled**	**ID**	**Unscaled**	**ID**	**Scaled**	**ID**	**Unscaled**
1a	-1.000000	1	3.00000	1a	-1.000000	1	3.00000
1b	-1.000000			1b	-1.000000		
1c	1.000000			1c	1.000000		
2	-1.000000	2	-0.10000	2	-0.200000	2	0.30000
3	-1.000000	3	0.00000	3	-1.000000	3	0.00000
4	1.000000	4	1.00000	4	-1.000000	4	0.00000
5	1.000000	5	1.00000	5	-1.000000	5	0.00000
6	-1.000000	6	0.00000	6	1.000000	6	1.00000
7	1.000000	7	1.00000	7	-1.000000	7	0.00000
8	-1.000000	8	0.00000	8	-1.000000	8	0.00000
9	-1.000000	9	0.00000	9	-1.000000	9	0.00000
10	1.000000	10	0.10000	10	-1.000000	10	0.06000
11	-1.000000	11	0.00800	11	-1.000000	11	0.00800
12	1.000000	12	0.08000	12	-1.000000	12	0.06000
13	1.000000	13	0.01300	13	-0.209814	13	0.00998
14	-1.000000	14	2000.00000	14	-1.00000	14	2000.00000
15	-1.000000	15	2000.00000	15	-1.000000	15	2000.00000
16	1.000000	16	0.20000	16	0.396026	16	0.154702
17	-1.000000	17	0.05000	17	0.600000	17	0.25000
18	-1.000000	18	0.01300	18	1.000000	18	0.20000
19	-1.000000	19	0.75000	19	1.000000	19	1.50000
20	-1.000000	20	0.00000	20	1.000000	20	15.00000
21	1.000000	21	0.52000	21	1.000000	21	0.52000
22a	-1.000000	22	3.00000	22a	-1.000000	22	4.00000
22c	1.000000			22c	-1.000000		
23	-1.000000	23	0.01000	23	1.00000	23	0.01300
24	0.677532	24	1.09357	24	0.428571	24	1.05000
25	1.000000	25	120.00000	25	1.000000	25	120.00000
26	-0.226507	26	0.04320	26	1.000000	26	0.08000
27	1.000000	27	0.08000	27	-1.00000	27	0.02000
28	-1.000000	28	0.04000	28	-1.000000	28	0.04000
29	-1.000000	29	0.02000	29	-1.000000	29	0.02000
30	0.410575	30	61.15863	30	-0.333332	30	50.0000
31	1.000000	31	1.20000	31	-1.000000	31	0.80000
32	1.000000	32	0.52000	32	1.000000	32	0.52000
33b	1.000000	33	2.00000	33b	-1.000000	33	4.00000
33c	-1.000000			33c	-1.000000		

Appendix B

The unscaled solutions for Solutions 5, 14, and 15 found using MARSOPT and Solution 4 from Data Set 1 are displayed in Table B1.

Table B1 Unscaled solutions for selected points

	MARSOPT			Data Set 1
Variable	**Solution 5**	**Solution 14**	**Solution 15**	**Solution 4**
1	3	3	3	4
2	-0.1	0.332471	0.3	0.1
3	0	0	0	0
4	1	1	0	1
5	1	1	0	1
6	0	0	1	0
7	1	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0.1	0.1	0.1	0.1
11	0.008	0.008	0.013	0.013
12	0.08	0.08	0.08	0.08
13	0.013	0.013	0.008	0.013
14	2000	2000	2000	2000
15	2000	2000	2000	2800
16	0.2	0.2	0.175839	0.15
17	0.05	0.05	0.25	0.25
18	0.013	0.013	0.013	0.013
19	0.75	0.75	0.75	1
20	0	0	0.940278	0
21	0.52	0.52	0.52	0.52
22	3	3	1	3
23	0.01	0.01	0.01	0.01
24	1.0935681	1.05	1.05	1.05
25	120	120	120	60
26	0.04320479	0.08	0.08	0.06
27	0.07999997	0.02	0.08	0.06
28	0.04	0.04	0.04	0.08
29	0.02	0.040026	0.02	0.02
30	61.158625	51.759294	49.999995	50
31	1.2	0.8	0.8	1.2
32	0.52	0.52	0.52	0.52
33	2	2	4	4

Table A2 Objective value and output values for the constraints

No.	Name	Model used	Objective /RHS	SLR	PL-MARS
	Obj-pb05-RRS		Minimize	0.29872	0.60433
1	constr-far50-ChestD	SLR	≤ 1	0.35019	0.15127
2	constr-far50-ChestG	SLR	≤ 1	0.55169	0.53030
3	constr-far50-Chest-IP-min	PL-MARS	≥ 1	2.52737	2.76537
4	constr-far50-FemurL	PL-MARS	≤ 1	0.25288	0.41356
5	constr-far50-FemurR	PL-MARS	≤ 1	0.55534	0.44486
6	constr-far50-HIC	SLR	≤ 1	0.27121	0.63180
7	constr-far50-Head-IP-min	SLR	≥ 1	6.75682	9.12077
8	constr-far50-NeckFzMax	PL-MARS	≤ 1	0.44529	0.22826
9	constr-far50-NeckFzMin	SLR	≤ 1	0.47635	0.98620
10	constr-far50-Nij	PL-MARS	≤ 1	1.00000	0.87613
11	constr-pb05-ChestD	SLR	≤ 1	0.34780	0.22531
12	constr-pb05-ChestG	SLR	≤ 1	0.64840	0.86639
13	constr-pb05-Chest-IP-min	SLR	≥ 1	4.52105	4.11027
14	constr-pb05-FemurL	PL-MARS	≤ 1	0.20939	0.03644
15	constr-pb05-FemurR	SLR	≤ 1	0.12128	0.05257
16	constr-pb05-HIC	PL-MARS	≤ 1	0.66561	0.62946
17	constr-pb05-Head-IP-min	SLR	≥ 1	3.72812	3.97494
18	constr-pb05-NeckFzMax	SLR	≤ 1	0.34700	0.84054
19	constr-pb05-NeckFzMin	SLR	≤ 1	0.03380	0.10829
20	constr-pb05-Nij	SLR	≤ 1	0.26884	0.68289
21	constr-pb50-ChestD	PL-MARS	≤ 1	0.35458	0.38053
22	constr-pb50-ChestG	SLR	≤ 1	0.68464	1.00000
23	constr-pb50-Chest-IP-min	SLR	≥ 1	5.11609	4.28510
24	constr-pb50-FemurL	SLR	≤ 1	0.10689	0.39837
25	constr-pb50-FemurR	SLR	≤ 1	0.09968	0.26965
26	constr-pb50-HIC	SLR	≤ 1	0.99885	1.00000
27	constr-pb50-Head-IP-min	SLR	≥ 1	4.21463	5.31041
28	constr-pb50-NeckFzMax	SLR	≤ 1	0.28652	0.31939
29	constr-pb50-NeckFzMin	SLR	≤ 1	0.05337	0.05627
30	constr-pb50-Nij	SLR	≤ 1	0.42838	0.31123
31	constr-pu05-ChestD	SLR	≤ 1	0.32661	0.42138
32	constr-pu05-ChestG	SLR	≤ 1	0.54122	0.67659
33	constr-pu05-Chest-IP-min	SLR	≥ 1	1.00000	1.14924
34	constr-pu05-FemurL	SLR	≤ 1	0.95823	0.67842
35	constr-pu05-FemurR	SLR	≤ 1	0.99617	0.83785
36	constr-pu05-HIC	SLR	≤ 1	0.28085	0.38509
37	constr-pu05-Head-IP-min	SLR	≥ 1	5.98856	6.76919
38	constr-pu05-NeckFzMax	SLR	≤ 1	0.10464	0.19225
39	constr-pu05-NeckFzMin	SLR	≤ 1	0.31794	0.45812
40	constr-pu05-Nij	SLR	≤ 1	0.86393	0.62704
41	constr-pu50-ChestD	PL-MARS	≤ 1	0.34813	0.39099
42	constr-pu50-ChestG	SLR	≤ 1	0.74906	0.80456
43	constr-pu50-Chest-IP-min	PL-MARS	≥ 1	1.10747	1.62352
44	constr-pu50-FemurL	SLR	≤ 1	0.67282	0.91119
45	constr-pu50-FemurR	SLR	≤ 1	0.65942	0.92952
46	constr-pu50-HIC	SLR	≤ 1	1.00000	0.92596
47	constr-pu50-Head-IP-min	SLR	≥ 1	7.52112	8.95609
48	constr-pu50-NeckFzMax	SLR	≤ 1	0.12831	0.32739
49	constr-pu50-NeckFzMin	SLR	≤ 1	0.40559	0.54765
50	constr-pu50-Nij	SLR	≤ 1	0.69390	0.66288