

Limit state function identification using Support Vector Machines for discontinuous responses and disjoint failure domains

Anirban Basudhar*, Samy Missoum, Antonio Harrison Sanchez

Aerospace and Mechanical Engineering Department, The University of Arizona, Tucson, AZ 85721, USA

Received 25 March 2007; received in revised form 27 August 2007; accepted 30 August 2007

Available online 12 September 2007

Abstract

This article presents a method for the explicit construction of limit state functions using Support Vector Machines (SVM). Specifically, the approach aims at handling the difficulties associated with the reliability assessment of problems exhibiting discontinuous responses and disjoint failure domains. The SVM-based explicit construction of limit state functions allows for an easy calculation of a probability of failure and enables the association of a specific system behavior with a region of the design space. The explicit limit state function can then be used within a reliability-based design optimization (RBDO) problem. Two problems are presented to demonstrate the successful application of the developed method for explicit construction of limit state function and reliability-based optimum design.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Support Vector Machines; Limit state functions; Discontinuities; Disjoint failure domains; Reliability-based design optimization

1. Introduction

Nonlinear problems are often characterized by various and sudden behavioral changes which are associated with the presence of critical points. A typical example is a geometrically nonlinear structure which globally buckles for a load larger than the limit load. Because these abrupt changes can be triggered by infinitesimally small modifications of design parameters or loading conditions, the responses of the system are discontinuous in a mathematical sense. In the context of reliability, these slight variations often fall in the range of uncertainties.

In simulation-based design, discontinuities present a serious problem for optimization or probabilistic techniques because it is usually assumed that the system's responses are continuous. In optimization, this limits any traditional gradient-based method or response surface technique. When considering reliability, discontinuities might hamper the use of approximation methods such as First Order and Second Order Reliability Methods (FORM and SORM) [1], Advanced Mean

Value (AMV) [2], or Monte Carlo simulations [3] with response surfaces [4].

In addition to discontinuities, nonlinear problems are characterized by disjoint failure regions, thus further limiting the use of classical approaches to assess probabilities of failure. These failure regions are often associated with distinct system behaviors, a phenomenon that is found in structural impact problems (e.g. vehicle crash) [5].

In order to tackle discontinuities and disjoint spaces, we propose to decompose the design space by explicitly defining the boundaries of the failure regions. This is a major difference compared to traditional simulation-based approaches for which the limit state functions are defined implicitly (e.g. a threshold on a response given by a finite element code). By defining explicit boundaries, one can associate a region of the design space with a specific behavior. In addition, this provides a way to handle uncertainties; the calculation of failure probabilities is made efficient as the verification of the state of a sample (failed or safe) is straightforward.

With a discontinuous behavior, it is not always possible to define an a priori threshold defining the boundary between “failure” and “non-failure”. However, discontinuities can be detected by data mining techniques such as clustering [6] which automatically identifies groups of similar responses. It

* Corresponding author. Tel.: +1 520 309 5202.

E-mail address: anirban@email.arizona.edu (A. Basudhar).

is then possible to map these clusters to specific regions of the design space, thus providing the first step towards the explicit identification of failure regions.

Several attempts to explicitly decompose the design space have been proposed in the case of nonlinear transient dynamic problems. In [7], the optimization of a tube impacting a rigid wall was performed with respect to its length and thickness. The goal was to enforce crushing and avoid global buckling while taking uncertainties into account. The optimization was based on the decomposition of the design space using hyperplanes and ellipsoids. While one part of the design space corresponded to the crushing of the tube, the other one (failure region) was associated with global buckling. The approach was later extended to the use of a convex hull for the definition of the boundaries of a failure region [8].

However, the tools used to create a failure region were not satisfactory as they were limited to a single convex set, and therefore did not address the issue of non-convex disjoint failure domains. In this paper, the approach is generalized by constructing the boundaries of specific regions of the design space using Support Vector Machines (SVM) [9,10]. SVM is a powerful classification tool that allows the construction of linear or nonlinear optimal “decision functions” between classes in a multidimensional space. The decision functions (i.e. the limit state functions) can be non-convex and form several disjoint subsets.

The explicit design space decomposition is made possible by first studying the responses with a design of experiments (DOE) [11]. By using Improved Distributed Hypercube Sampling (IHS), the samples are uniformly distributed in the design space. The responses, obtained for each DOE sample, are classified into groups forming clusters. This allows one to assign a class to each response, thus enabling the use of SVM to define the boundaries of the failure regions.

The approach is applied to two problems: the first one demonstrates the construction of an explicit limit state function for an analytical problem with disjoint failure regions. This academic problem allows to accurately quantify the error that is made in reproducing the limit state function. A metric is introduced to quantify the error between the approximated limit state function and the actual one. The second problem deals with the reliability-based design optimization (RBDO) of a geometrically nonlinear arch exhibiting snap-through. For this problem, the discontinuities are detected using a clustering technique. The objective is to minimize the volume while avoiding global buckling with a given probability.

2. Support Vector Machines

SVM are a classification tool that belong to the class of machine learning techniques. They are becoming increasingly popular and have widespread applications in pattern recognition. The main features of SVM lie in their ability to define complex decision functions that optimally separate two classes of data samples.

The purpose of this section is to provide the reader with a first overview of the SVM algorithm. The basic SVM theory is

presented through a detailed explanation in the case of a linearly separable data set. It is then extended to the case where the data is not linearly separable.

2.1. Linear decision function

To introduce SVM, we define a set of N training points \mathbf{x}_i in a p dimensional space. Each point is associated with one of two classes characterized by a value $y_i = \pm 1$. The SVM algorithm finds the boundary (decision function) that optimally separates the training data into the two classes. In the case of linear decision functions, the basic idea is to maximize the “margin” between two parallel hyperplanes that separate the data. This pair of hyperplanes is required to pass at least through one of the training points of each class, and there cannot be any points inside the margin (Fig. 1). The points that these hyperplanes pass through are referred to as *support vectors*. The optimum decision function is half way between these two previously described hyperplanes. One of the outer hyperplanes consists of those points which satisfy:

$$\mathbf{w} \cdot \mathbf{x} + b = +1 \quad (1)$$

The other hyperplane contains the points that follow:

$$\mathbf{w} \cdot \mathbf{x} + b = -1, \quad (2)$$

where \mathbf{w} is the vector of hyperplane coefficients, \mathbf{x} is the vector of variables and b is the bias. All the points of the class $y = +1$ lead to a positive value of SVM and all the points in the class $y = -1$ are “negative”. Eqs. (1) and (2), and the constraint that no point can lie between the two aforementioned hyperplanes, can be combined in a single global constraint defined as follows:

$$y_i(\mathbf{w} \cdot \mathbf{x} + b) - 1 \geq 0 \quad (3)$$

The perpendicular distance between the two support hyperplanes is $\frac{2}{\|\mathbf{w}\|}$. Therefore, determining the support hyperplanes (i.e. solving for \mathbf{w} and b) reduces to the following optimization problem:

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (4)$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \quad 1 \leq i \leq N$$

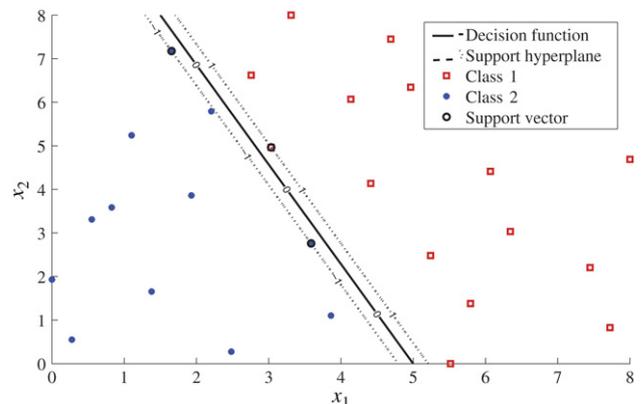


Fig. 1. Linear decision function separating class 1 from class 2.

This is a Quadratic Programming (QP) problem since the objective function is quadratic and the constraints are linear. Problem (4) is convex and can be solved efficiently with available optimization packages. As a result, the optimal \mathbf{w} , b , and the Lagrange multipliers λ_i at the optimum are obtained. From this, the classification of any test point \mathbf{x} is obtained by the sign of the following function:

$$s = b + \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i \cdot \mathbf{x} \quad (5)$$

Following the Kuhn and Tucker conditions, the Lagrange multipliers associated with the support vectors will be strictly positive while the others will be equal to zero. In general, the number of support vectors is a small fraction of the total number of training points. Eq. (5) can be rewritten with respect to the number of support vectors NSV :

$$s = b + \sum_{i=1}^{NSV} \lambda_i y_i \mathbf{x}_i \cdot \mathbf{x} \quad (6)$$

In the case where the data is not linearly separable, the optimization problem (4) will be infeasible. The inequality constraints are then relaxed by the introduction of nonnegative slack variables ξ_i which are minimized through a penalized objective function. The relaxed optimization problem is

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \quad (7)$$

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq -\xi_i \quad 1 \leq i \leq N$$

The coefficient C is referred to as the misclassification cost. In the dual formulation of Problem (7), C becomes the upper bound for all the Lagrange multipliers.

2.2. Nonlinear decision function

SVM can be extended to the case of nonlinear decision functions by mapping the original set of variables to a higher dimensional space referred to as the feature space. In this n dimensional feature space, the new components of a point \mathbf{x} are given by $(\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_n(\mathbf{x}))$ where ϕ_i are the features. The nonlinear decision function is obtained by formulating the linear classification problem in the feature space. The classification is then obtained by the sign of

$$s = b + \sum_{i=1}^{NSV} \lambda_i y_i \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle \quad (8)$$

where $\Phi = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_n(\mathbf{x}))$ and \langle, \rangle is the inner product.

The inner product in Eq. (8) forms a kernel K , so that the decision function is written:

$$s = b + \sum_{i=1}^{NSV} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) \quad (9)$$

2.3. Types of kernels

The two most commonly used kernels functions, the polynomial and the Gaussian kernels are presented. Some other kernels that may be used are multilayer perceptrons, Fourier series and splines.

2.3.1. Polynomial kernels

A polynomial kernel is defined as

$$K(\mathbf{x}_i, \mathbf{x}) = (\langle \mathbf{x}_i, \mathbf{x} \rangle + 1)^d \quad (10)$$

where d is the degree of the polynomial. In Fig. 2, a second degree polynomial kernel is used to classify a data set into two classes. The classes are represented by blue asterisk and red square data points. The zero value isocontour is the optimum decision function, which splits the space into a negative and positive region. The -1 and $+1$ isocontours pass through the support vectors of each class (circles on the figure).

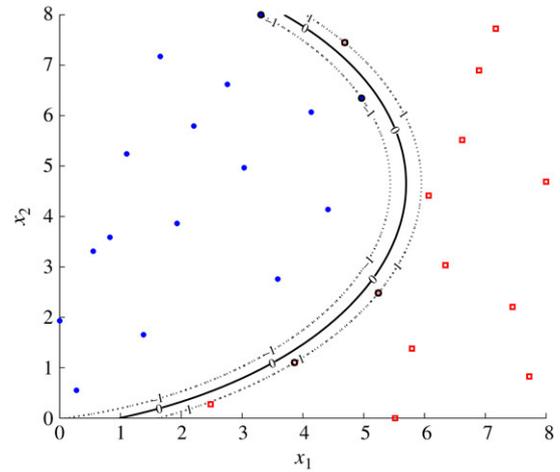


Fig. 2. Two-dimensional second degree polynomial separating two classes. The zero value isocontour is the optimum decision function and the support vectors are depicted with circles.

2.3.2. Gaussian kernels

A Gaussian kernel is defined as:

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}\right) \quad (11)$$

where σ is the width parameter. An example of a Gaussian kernel is provided in Fig. 3.

2.4. General features of SVM

SVM has several qualities which make it a very powerful tool for pattern recognition and classification. These qualities make it a very useful tool in probabilistic design and optimization. Some features of SVM are:

1. SVM is multidimensional: It is capable of classifying data in the multidimensional space. This is a key factor in structural design optimization where the number of design variables in some cases is very large. In Fig. 4 a Gaussian kernel in

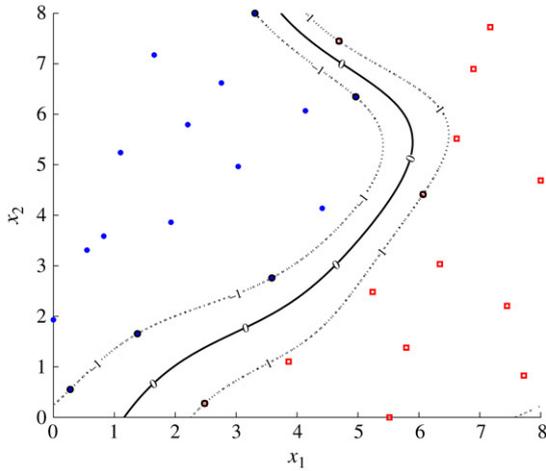


Fig. 3. Two-dimensional Gaussian kernel separating two classes. The two classes are shown by blue asterisk and red squares. The zero value isocontour represents the optimum decision function and the support vectors are depicted with circles.

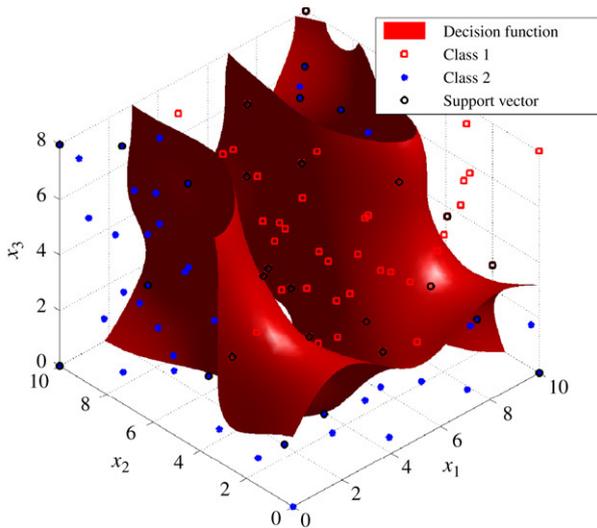


Fig. 4. Three-dimensional Gaussian kernel separating the two classes shown by blue asterisk and red squares.

three dimensions is used to optimally define the boundary between class 1 (blue asterisk) and class 2 (red square).

2. Optimal decomposition: There can be several ways to separate two classes of data. However, SVM decomposes the design space by an optimal separating function which maximizes the margin between the classes.
3. Separation of disjoint regions: SVM is capable of identifying disjoint regions. Hence, it can be applied to problems for which the limit state function forms the boundaries of disjoint failure regions.
4. SVM used as regression tool: In addition to the classification of data, SVM can also be used for regression [12,13].

3. Methodology for explicit identification of failure regions

The methodology for the explicit failure region identification with SVM is presented in this section. The first step consists of

performing a design of experiments (DOE) [14] in which the random variables are sampled. The responses of the system for the DOE samples are then evaluated and classified into distinct classes that correspond to failure or safe system behaviors. These classified design configurations are then used as training points for the SVM algorithm. The steps are summarized in Fig. 5.

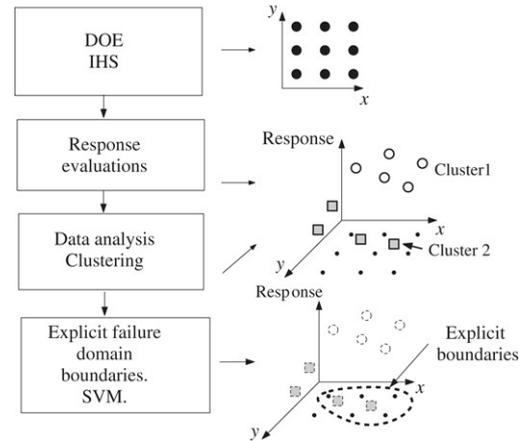


Fig. 5. Methodology for explicit failure region identification.

3.1. Step 1: Design of experiments — IHS

There exist several DOE techniques such as Random Sampling, Latin Hypercube Sampling (LHS) [15], D-optimal sampling [11], Optimal Latin Hypercube Sampling (OLHS) [16]. In our approach, the training samples are generated using an Improved Distributed Hypercube Sampling (IHS) [17]. IHS constitutes an improvement compared to the traditional LHS as it uniformly distributes samples over the design space and therefore avoids clustering of information. Fig. 6 provides an example of LHS and IHS samplings.

3.2. Step 2: Estimation and classification of responses

The response of a system for each training sample is obtained using a finite element software such as ANSYS. The responses are then classified using a threshold value or a clustering method such as K-means [18] or hierarchical clustering [6,19]. The classification of responses into two distinct classes (e.g. safe or failed) provides the information needed by the SVM algorithm to generate the optimal separating function (i.e. limit state function). Fig. 7 provides an example of clustering detection using K-means. The widely used K-means clustering technique, which is used for the present work, is explained in the Appendix A.

3.3. Step 3: Definition of an explicit limit state function

Following the classification of the response into two classes, SVM provides an explicit limit state function in terms of the random variables. In the following section, several factors influencing the accuracy of SVM are considered.

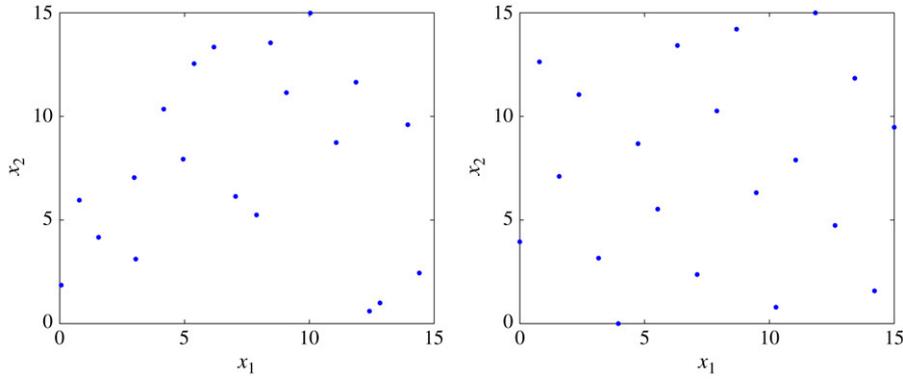


Fig. 6. Two-dimensional sample distribution using LHS and IHS respectively.

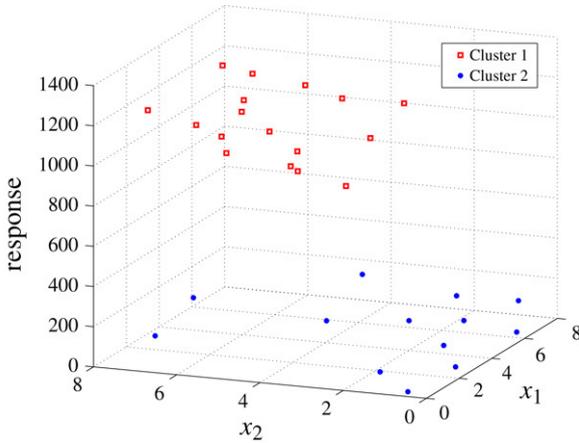


Fig. 7. Classification of data into two clusters using K-means.

4. Influence of training sample size and kernel parameters

For constructing an accurate limit state function, it is important to choose an appropriate number of training samples. However, function evaluation at a single training point can be quite expensive computationally, and hence, an optimal number of points needs to be selected. In addition, the limit state function may vary depending on the type of kernel and values of associated parameters.

4.1. Study of the influence of training sample size. Convergence measure

In order to find the appropriate number of training samples for a particular problem, the evolution of SVM with respect to the number of training points is studied. For this purpose, another set of n_{conv} points are generated using IHS. These points are referred to as “convergence” points. For a particular number of training samples, the value of SVM at each of these convergence points is calculated. A vector is then constructed that contains the relative changes in SVM values for each of the convergence points between two successive sets of training samples. The successive change in limit state function with the increase in number of training samples is quantified as the norm

of this vector:

$$\Delta_k = \sqrt{\sum_{i=1}^{n_{\text{conv}}} \left(\frac{s_{k-1}^i - s_k^i}{s_k^i} \right)^2} \quad (12)$$

where s_k^i refers to the SVM value for the i th convergence point at iteration k . Δ_k is the norm of the relative change of SVM, measured on the convergence points, between iteration $k - 1$ and k . An example of the effect of training sample size is given in Fig. 9. Initially, the increase of the number of training points has a large impact on the limit state function generated by SVM. However, this variation between successive limit state functions decreases gradually as the number of training samples increases. This behavior is as expected since, as more training sample points are added, they add information. However, after certain limit the information provided by the new points is redundant and hence, does not contribute much in changing the function. The training sample size for which the limit state function becomes steady is selected for the classification.

4.2. Study of the influence of kernel parameters. Introduction of an error measure

Because of its generality, the Gaussian kernel is chosen for this study. However, the width parameter σ in the Gaussian kernel, as described in Eq. (11), has to be chosen carefully.

In order to perform this study, known analytical limit state functions are used so that the error between the predicted and actual functions can be evaluated accurately. The number of training samples is kept constant, and different values of σ are used to generate the separating function using SVM. The corresponding limit state functions are then compared to the actual analytical ones. For measuring the error, N_{test} test samples are generated by IHS. By generating a large number of test points, the error can be assessed by calculating the fraction of misclassified test points for each candidate limit state for a given value of σ . A test sample point for which the sign of SVM does not match the sign of the actual function is considered misclassified. That is, the error ϵ is

$$\epsilon = \frac{\text{num} \left(\left(b + \sum_{i=1}^{NSV} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}_{\text{test}}) \right) y_{\text{test}} \leq 0 \right)}{N_{\text{test}}} \quad (13)$$

where \mathbf{x}_{test} and y_{test} represent a test sample and the corresponding class value (± 1) for the actual (known) limit state function.

This error measure is used to assess the optimal value for the width parameter of the Gaussian kernel. In the case where the actual function is not known, an approach consists of choosing the width parameter so as to minimize the number of support vectors.

5. Reliability-based design optimization (RBDO) methodology

RBDO [20–22] problems can be formulated as:

$$\begin{aligned} \min_{\bar{\mathbf{x}}} \quad & F(\bar{\mathbf{x}}) \\ \text{s.t.} \quad & P(g(\mathbf{x}) > 0) \leq P_{\text{target}} \\ & \mathbf{h}(\bar{\mathbf{x}}) \leq \mathbf{0} \end{aligned} \quad (14)$$

where F is the objective function, g is a limit state function, and \mathbf{h} is a set of deterministic constraints. $\bar{\mathbf{x}}$ is the vector of mean values which are the variables of the optimization problem. Note that without loss of generality, \mathbf{x} can contain random and deterministic variables. The limit state function $g(\mathbf{x}) = 0$ divides the design space into a failure region $g(\mathbf{x}) > 0$ and a safe region. P_{target} is referred to as the target probability of failure. Note that there can be several limit state functions for the same problem.

5.1. Probability of failure estimate

The calculation of the probability of failure P_f can be achieved through approximation schemes such as Monte Carlo Simulations (MCS), FORM (First-Order Reliability Method) or SORM (Second-Order Reliability Method), Advanced-Mean Value (AMV), etc. In the proposed SVM-based approach it is natural to use Monte Carlo simulations because the evaluation of the state of a sample is very efficient.

However, the inclusion of a brute Monte Carlo process within an optimization loop is not recommended for three main reasons:

1. It is time consuming.
2. The probability calculated by MCS is noisy due to the randomness of the sampling.
3. The probabilities are typically low and can vary by orders of magnitude during the optimization process.

In order to regularize the probabilistic constraint, the reliability index β is used:

$$\beta = -\Phi^{-1}(P_f) \quad (15)$$

with Φ being the standard normal cumulative distribution function.

5.2. Response surface approximation of the reliability index

The issues related to the use of MCS are also solved using response surfaces. In order to avoid repetitive and costly Monte Carlo simulations and create smoother response variations, the

reliability index is approximated by response surfaces [4,23]. The approximation function $\hat{\beta}$ is built using the reliability index values calculated at each of the N training points. In this paper, the approximation is also performed by SVM but used for regression [12,13].

The accuracy of the response approximation is measured by the R square quantity that measures the global error, and the RMAE (relative maximum absolute error) that measures the maximum local error.

$$R^2 = 1 - \frac{\sum_{i=1}^N (\beta_i - \hat{\beta}_i)^2}{\sum_{i=1}^N (\beta_i - \bar{\beta})^2} \quad (16)$$

$$\text{RMAE} = \frac{\max(|\beta_i - \hat{\beta}_i|)}{\sigma_\beta} \quad (17)$$

where β_i is the actual reliability index value, $\hat{\beta}_i$ is the approximated value, and $\bar{\beta}$ and σ_β are the mean and standard deviation of the N samples.

The optimization problem becomes

$$\begin{aligned} \min_{\bar{\mathbf{x}}} \quad & F(\bar{\mathbf{x}}) \\ \text{s.t.} \quad & \hat{\beta}(\bar{\mathbf{x}}) \geq \beta_{\text{target}} \\ & \mathbf{h}(\bar{\mathbf{x}}) \leq \mathbf{0} \end{aligned} \quad (18)$$

6. Examples

Two test examples demonstrating the efficiency of the explicit design space decomposition using SVM are presented. The first problem uses an analytical function to emulate the case of disjoint failure regions. The second problem represents an actual structural problem with a discontinuous behavior due to nonlinearities. Because of the discontinuities, a clustering technique needs to be used in order to classify the responses. A reliability-based design optimization is carried out for this problem.

6.1. Analytical problem with disjoint regions. Design space decomposition

For this problem, the limit state function is defined by an analytical function of two variables x_1 and x_2 .

$$f(x_1, x_2) = x_2 - |\tan(x_1)| - 1 \quad (19)$$

This function represents disjoint regions. The region where $f(x_1, x_2) > 0$ is labeled +1 and the complementary region is labelled -1. The x_1 and x_2 variables are considered uniformly distributed variables with ranges [0, 7] and [0, 6] respectively.

6.1.1. Construction of an explicit limit state function

90 training samples generated using IHS are used for the construction of the limit state function using SVM. A Gaussian kernel with parameter σ equal to 2.2 is used. The misclassification coefficient C is set to infinity. Fig. 8 shows

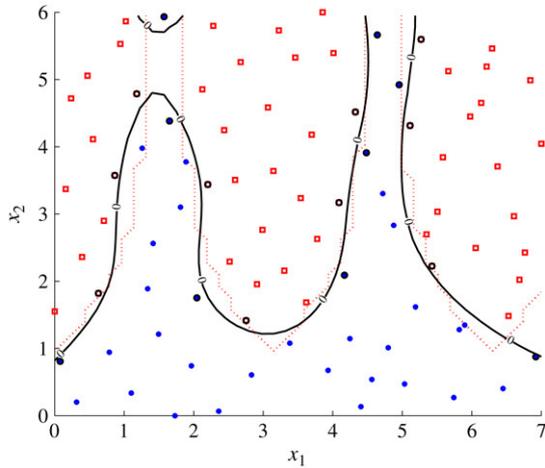


Fig. 8. Two-dimensional problem: design space decomposition using SVM. The dotted curve is actual (known) decision function and the solid curve is the SVM limit state function.

the design space decomposition obtained by SVM. The SVM limit state function found is represented by the solid curve while the actual expected function is shown by the dotted curve. The support vectors x_i , Lagrange multipliers and class values are shown in Table 1. Note that the bias $b = 16.8196$.

Table 1
Support vector data

Support Vector	x_1	x_2	Lagrange multiplier	Class y
1	2.2022	3.4382	87	+1
2	4.0899	3.1685	3,317	+1
3	4.6404	5.6629	317	-1
4	6.9213	0.8764	65	-1
5	5.2697	5.5955	1,696	+1
6	5.1124	4.3146	6,520	+1
7	5.4270	2.2247	105	+1
8	4.1685	2.0899	427	-1
9	4.4831	3.9101	10,742	-1
10	4.9551	4.9213	7,976	-1
11	4.3258	4.5169	8,505	+1
12	1.5730	5.9326	287	-1
13	1.1798	4.7865	1,240	+1
14	1.6517	4.3820	1,655	-1
15	0.8652	3.5730	60	+1
16	2.7528	1.4157	10	+1
17	2.0449	1.7528	172	-1
18	0.0787	0.8090	92	-1
19	0.6292	1.8202	193	+1

Coordinates of support vectors, Lagrange multipliers, and class (-1 or 1).

6.1.2. Study of SVM with respect to number of training points

A study of the evolution of SVM is done with respect to the number of training samples in the design of experiments as described in Section 4.1. For a two-dimensional problem, the number of convergence points n_{conv} is chosen as 800. Note that only the SVM is evaluated at these points and not the actual response. The difference between successive limit state functions decreases gradually and the separating function

calculated by SVM becomes steady around 90 training points as can be seen in Fig. 9.

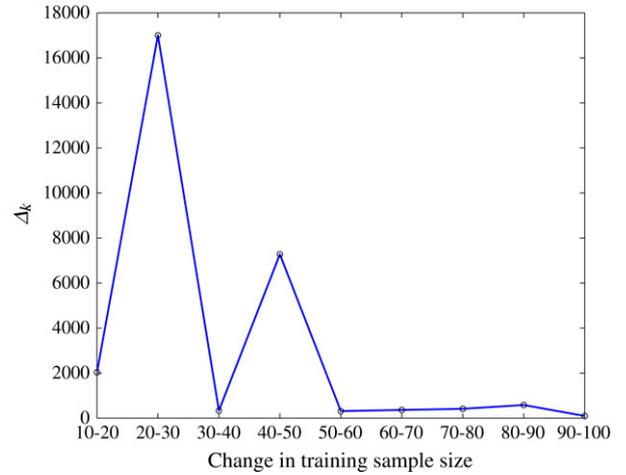


Fig. 9. Two-dimensional problem: variation of SVM with respect to the variation of number of training samples.

6.1.3. Variation of the limit state function with respect to kernel parameters

As mentioned in Section 4.2, the limit state function is also contingent upon the value of the width parameter σ of the Gaussian kernel. For a range of width parameter value from 0.5 to 5.0, the classification error was studied for 80 and 90 training samples (Fig. 10). 800 test points were used to quantify the error. Since the function is analytical, the number of samples can be quite high. From the numerical experiments it is observed that the classification error with σ between 2.0 and 3.0 is quite low for the problem at hand. Note that this result would be different had the variables been scaled.

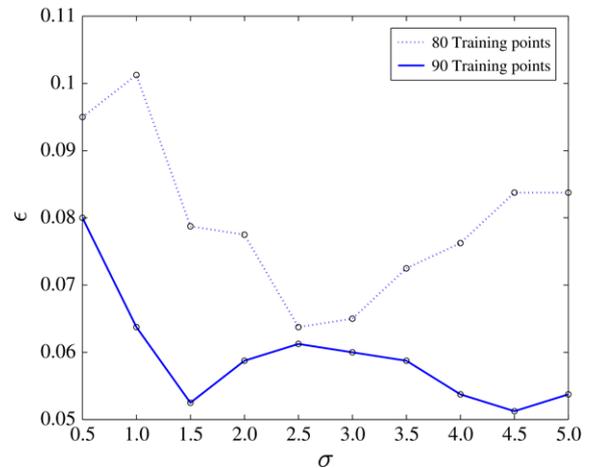


Fig. 10. Error in estimation of limit state function with respect to the width parameter σ of the Gaussian kernel.

6.2. Arch structure with discontinuous response. Construction of the limit state function

The design space decomposition using SVM is applied to an arch structure (see Fig. 11) subjected to a point load at

the center. The arch is a typical example of a geometrically nonlinear structure exhibiting a snap-through behavior once the limit load is reached. The design space decomposition will enable the explicit separation of the regions of the design space corresponding to buckling (i.e., failure) and non-buckling.

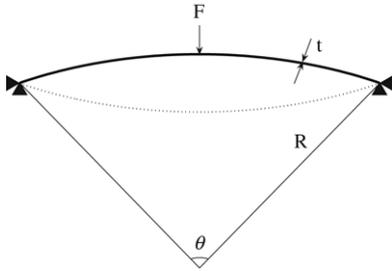


Fig. 11. Arch geometry and loading. Snap-through behavior.

The design variables are the thickness t and the width w of the arch, and are considered as random variables. In addition, the load is also a random parameter. The arch has a radius of curvature R of 8 m and it subtends an angle θ of 14 degrees at its center of curvature. It is modeled using shell elements and the arch is hinged at the supports. Due to the symmetries of the problem, only one fourth of the arch needed to be modeled. The range of values allowed for the design parameters are tabulated in Table 2.

Table 2
Range of design parameters for arch problem

	Thickness (t) (mm)	Width (w) (mm)	Force (F) (N)
Min value	3	150	2000
Max value	10	500	8000

A design of experiments is generated by IHS using 150 training samples with respect to the thickness, the width and the force. The variables are normalized by dividing the values by their respective maximum values. Note that the sampling does not yet take into account the specific probabilistic distributions, and the variables are assumed uniformly distributed. This is done in order to have as much information as possible over the whole design space to build the explicit limit state function and the response surfaces for optimization.

The studied response is the displacement of the central node which is solved for at each training sample using the finite element software ANSYS. This response is clearly discontinuous when a limit load is reached as the arch exhibits snap-through. The discontinuous variation of the displacement with respect to the thickness and width is depicted in Fig. 12 for a fixed value of the applied load.

Discontinuities are used efficiently as they allow one to segregate the responses into clusters. This is done by a clustering technique such as K-means. By specifying the search for two clusters, it is then possible to sort the responses between failure (buckling) and non-failure, and classify the samples in the design space into “+1” and “-1” classes. This information is input to the SVM algorithm, which then separates the regions

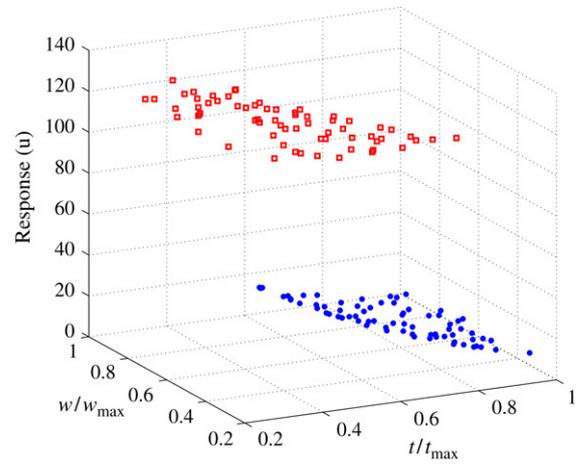


Fig. 12. Discontinuous response of arch problem. The response has been obtained for a constant load $F = 6400$ N.

of distinct behavior by an explicit limit state function. In this problem, a Gaussian kernel with parameter σ equal to 2.2 is used to determine the separating SVM hypersurface (Fig. 13). The misclassification cost C is set to infinity. The support vectors and the corresponding Lagrange multipliers are given in Table 3.

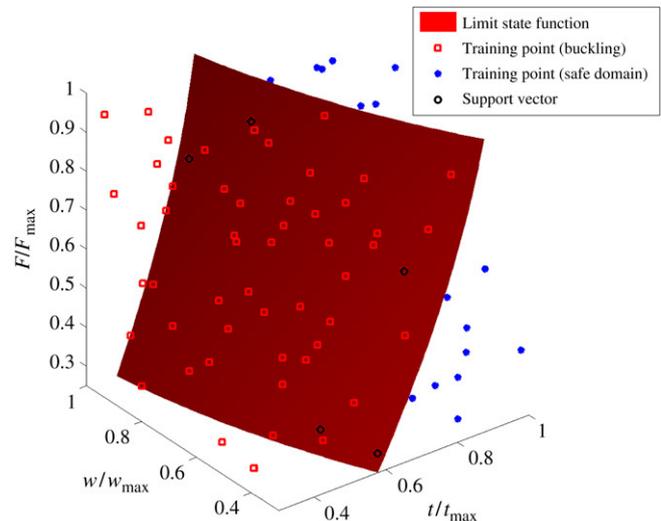


Fig. 13. Arch problem: Identification of the failure domain (design space decomposition) using SVM.

The sensitivity of SVM with respect to the number of training samples is shown in Fig. 14. The change between two successive limit state functions constructed with different number of training samples is quantified as described in Section 4.2 with 1600 convergence points. The limit state function stabilizes at around 150 training samples.

6.3. Arch structure — Reliability-based design optimization

Once the explicit SVM limit state function is obtained, design parameters of the arch are optimized while considering the uncertainties in the geometrical parameters and the force.

Table 3
Support vector data

Support Vector	Thickness	Width	Force	Lagrange multipliers	Class y
1	0.7228	0.3940	0.6728	87,846	+1
2	0.5537	0.9577	0.7634	2,114	+1
3	0.5866	0.3141	0.2903	7,671	+1
4	0.7698	0.4785	0.9245	32,532	-1
5	0.6523	0.4034	0.4765	88,174	-1
6	0.6195	0.8168	0.8993	3,705	+1
7	0.5349	0.4550	0.3054	19,370	+1

Normalized coordinates of support vectors, Lagrange multipliers, and class (-1 or 1). $b = 8.0473$.

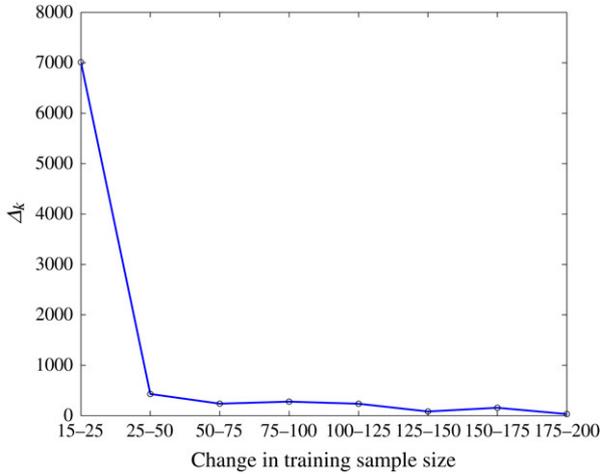


Fig. 14. Arch problem: variation of SVM with respect to the variation of number of training samples.

The objective is to minimize the volume of the arch structure while preventing global buckling with a probability less than 10^{-3} . The thickness t and the width w of the arch are the two random design variables, while the load is an external random parameter having a mean value of 6400 N. All three random variables have truncated normal distributions. The means of the design variables \bar{t} and \bar{w} are the optimization variables. Table 4 provides the distribution parameters for the random variables.

The optimization problem is:

$$\begin{aligned}
 &\min_{\bar{t}, \bar{w}} \text{Volume}(\bar{t}, \bar{w}) \\
 &\text{s.t. } P(\text{buckling}) \leq 10^{-3} \\
 &\quad 3.0 \leq \bar{t} \leq 10.0 \\
 &\quad 150.0 \leq \bar{w} \leq 500.0
 \end{aligned} \tag{20}$$

As described in Section 5.2, the probabilistic constraint is handled by using the SVM explicit limit state function and by fitting the reliability index with a response surface.

To calculate the probability of failure, a Monte Carlo simulation with 10^6 samples is run for every point of the initial design of experiments. The probability of failure calculated as the fraction of Monte Carlo samples lying in the failure domain, is then transformed using Eq. (15). We recall here that the use of brute Monte Carlo simulation is made possible as the evaluation of each sample consist of checking the sign of an analytical function.

Table 4
Arch problem: Distribution of random variables

	Mean	Std. dev.	Lower limit	Upper limit
Thickness	\bar{t} (optimized)	0.2 mm	$\bar{t} - 1$ mm	$\bar{t} + 1$ mm
Width	\bar{w} (optimized)	5 mm	$\bar{w} - 25$ mm	$\bar{w} + 25$ mm
Force	6400 N	640 N	4800 N	8000 N

Values of the reliability index calculated for the 150 training points are then fitted with SVM used for regression. The use of SVM for regression has the advantage of being able to fit highly multimodal functions and avoid over fitting. The kernel used is again Gaussian, with parameter $\sigma = 2.2$. Regression fitting of the reliability index β is shown in Fig. 15. However, before fitting the β values, points for which $P_f = 0$ or $P_f = 1$ are removed as the corresponding indices go to infinity. This does not affect the accuracy of the probabilistic constraint. There is enough information in the range around the target probability that is critical for estimating the probabilistic constraint.

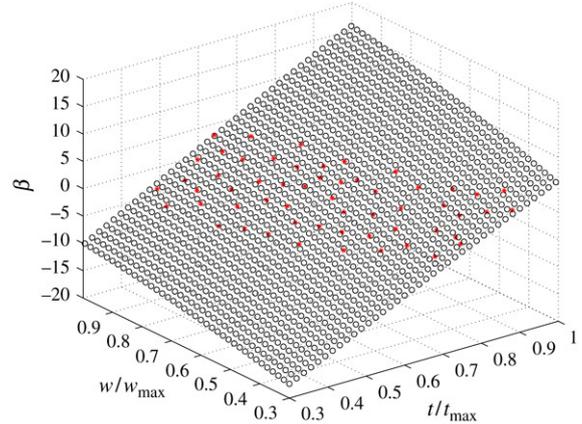


Fig. 15. SVM regression fitting of β values.

The maximum global goodness of fit and local error values for the approximation are:

$$R^2 = 0.9986$$

$$\text{RMAE} = 0.1075$$

The optimization problem consists of a simple analytical function and a response surface. It can therefore be solved efficiently with a gradient-based method such as the sequential quadratic programming method (SQP). Fig. 16 shows the objective function, constraints and the optimum design point.

The optimum results are gathered in Table 5. For comparison, the actual probability of failure at the found optimum is also given.

Table 5
Summary of the RBDO results for the arch

Normalized optimum thickness	0.9136
Normalized optimum width	0.3
Optimum thickness	9.136 mm
Optimum width	150 mm
Calculated optimum probability of buckling	0.001
Actual optimum probability of buckling	8.64×10^{-4}
Objective function value	$2.6788 \times 10^6 \text{mm}^3$

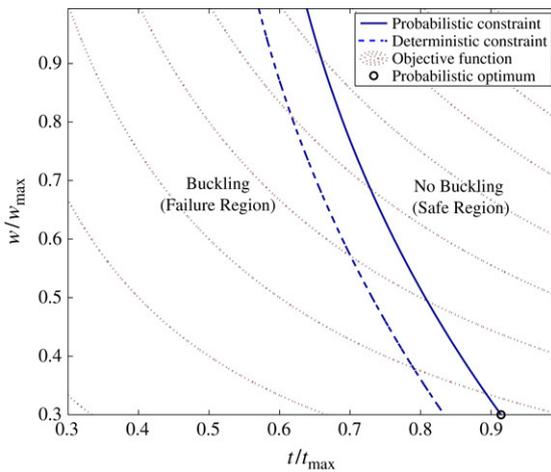


Fig. 16. Arch problem: Objective function, constraint functions, and optimum.

7. Concluding remarks

7.1. Summary

The proposed approach enables the construction of explicit limit state functions using SVM. The technique is particularly useful for problems exhibiting discontinuous responses and disjoint failure domains. Two problems with disjoint failure regions and discontinuous responses are presented to demonstrate the efficiency of the method for the explicit construction of limit state functions and for RBDO.

7.2. Limitations and future improvements

The proposed SVM-based methodology successfully meets the challenges of handling discontinuous responses and disjoint failure regions in reliability assessment or optimization. However, there are still certain areas where improvements can be made.

7.2.1. Reduction of the number of training points

The number of DOE points required to train the SVM increases with the problem dimensionality. The developed method needs to be improved further to reduce the number of training sample points required for constructing an accurate explicit limit state function. A method for adaptive update of the SVM limit state function is being developed for this purpose.

This will reduce the number of training points required, and also limit the increase in the number of points with dimensionality. Thus, extension of the method to problems with larger dimensions will be made efficient. Further, the scheme for adaptive update of DOE will automatically detect when a sufficient number of training points has been selected.

7.2.2. Accuracy of the probability of failure estimates

The sensitivity of the probability of failure estimates with respect to the number of training points has not been studied here. However, it should be noted that there might be some uncertainty with the probabilistic distribution used, which might have even more influence on the RBDO result than the accuracy of the limit state function. The final accuracy of the SVM limit state function will be increased by using the adaptive update scheme being developed. Even for small target failure probabilities, the probabilities of failure calculated with the final limit state function will be very accurate, and almost insensitive to further increase in the training set size. However, the accuracy of the probability estimates also depends on the number of Monte Carlo samples. In the future, it is envisioned to use importance sampling in order to decrease the number of Monte Carlo samples and reach smaller probabilities of failure. A method of direct calculation of the probability of failure, using the explicit expression of the SVM limit state function, is also being studied.

Appendix A. K-means clustering

The K-means method [18] is a non-hierarchical approach to identify clusters in data sets. The algorithm partitions N data points into K clusters S_j containing n_j points, so as to minimize the variance of the data within each cluster. The function to minimize is

$$J = \sum_{j=1}^K \sum_{n \in S_j} \|\mathbf{x}_n - \mu_j\|^2 \quad (21)$$

where \mathbf{x}_n is a vector representing the n th data point and μ_j is the geometrical mean of the data in S_j .

The algorithm starts by randomly partitioning the data into K initial sets. It then calculates the centroid of each set and rearranges the data points by assigning each one to its nearest centroid. This process is continued until convergence.

One of the limitations of the K-means algorithm, used in this paper, is that the number of clusters is given a priori. Techniques such as hierarchical clustering might help to avoid this difficulty as it investigates a set of data simultaneously over a variety of scales of distance and let the investigator decide which is the most appropriate to choose.

References

- [1] Haldar A, Mahadevan S. Probability, reliability, and statistical methods in engineering design. New York: Wiley and Sons; 2000.
- [2] Youn BD, Choi KK, Du L. Adaptive probability analysis using an enhanced hybrid mean value method. Structural and Multidisciplinary Optimization 2005;29(2):134–48.
- [3] Melchers R. Structural reliability analysis and prediction. John Wiley & Sons; 1999.

- [4] Myers RH, Montgomery DC. Response surface methodology. 2nd ed. Wiley; 2002.
- [5] Missoum S. Controlling structural impact failure modes in the presence of uncertainty. *Structural and Multidisciplinary Optimization* 2007; doi:10.1007/s00158-007-0100-z.
- [6] Martinez WL, Martinez AR. Exploratory data analysis with matlab. Computer science and data analysis series, Chapman & Hall/CRC; 2005.
- [7] Missoum S, Benchaabane S, Sudret B. Handling bifurcations for the optimal design of transient dynamics problems. In: Proceedings of the 45th conference AIAA/ASME/ASCE/AHS/ASC on structures, dynamics and materials. California (USA): Palm Springs; April 2004.
- [8] Missoum S, Ramu P, Haftka RT. A convex hull approach for the reliability-based design of nonlinear transient dynamic problems. *Computer Methods in Applied Mechanics and Engineering* 2007; doi:10.1016/j.cma.2006.12.008.
- [9] Shawe-Taylor J, Cristianini N. Kernel methods for pattern analysis. Cambridge University Press; 2004.
- [10] Tou JT, Gonzalez RC. Pattern recognition principles. Reading, MA: Addison-Wesley; 1974.
- [11] Montgomery DC. Design and analysis of experiments. Wiley and Sons; 2005.
- [12] Farag A, Mohamed RM. Regression using support vector machines: Basic report. Technical report. Louisville (KY, United States): Computer Vision and Image Processing Laboratory, Electrical and Computer Engineering Department, University of Louisville; 2004.
- [13] Clarke SM, Griebisch JH, Simpson TW. Analysis of support vector regression for approximation of complex engineering analyses. *Journal of Mechanical Design. Transactions of the ASME* 2005;127:1077–87.
- [14] Giunta AA, Wojtkiewicz SF, Eldred MS. Overview of modern design of experiments methods for computational simulations. In: 41st AIAA Aerospace Sciences Meeting and Exhibit. 2003.
- [15] Butler AN. Optimal and orthogonal latin hypercube designs for computer experiments. *Biometrika* 2001;88(3):847–57.
- [16] Liefvendahl M, Stocki R. A study on algorithms for optimization of latin hypercubes. *Journal of Statistical Planning and Inference* 2006;136:3231–47.
- [17] Beachkofski BK, Grandhi R. Improved distributed hypercube sampling. In: Proceedings of the 43rd conference AIAA/ASME/ASCE/AHS/ASC on Structures, Dynamics and Materials. Paper AIAA-2002-1274. April 2002.
- [18] Hartigan JA, Wong MA. A *K*-means clustering algorithm. *Applied Statistics* 1979;28:100–8.
- [19] Hamerly G, Elkan C. Alternatives to the k-means algorithm that find better clusterings. In: 11th International conference on information and knowledge management, November 2002. p. 600–7.
- [20] Youn BD, Choi KK. Selecting probabilistic approaches for reliability-based design optimization. *AIAA Journal* 2004;42(1):124–31.
- [21] Kharmanda G, Mohamed A, Lemaire M. Efficient reliability-based design optimization using a hybrid space with application to finite element analysis. *Structural and Multidisciplinary Optimization* 2002;24:233–45.
- [22] Youn BD, Choi KK, Yang RJ, Gu L. Reliability-based design optimization for crashworthiness of vehicle side impact. *Structural and Multidisciplinary Optimization* 2004;26:272–83.
- [23] Wang G, Shan S. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design* 2007; 129(4):370–80.