Fuzzy Clustering Based Hierarchical Metamodeling For Space Reduction and Design Optimization

G. Gary Wang[*]
The University of Manitoba
Winnipeg, MB, Canada R3T 5V6

Timothy W. Simpson[†]
The Pennsylvania State University
University park, PA, USA 16802

Abstract

For computation-intensive design problems, metamodeling techniques are commonly used to reduce the computational expense during optimization; however, they often have difficulty or even fail to model an unknown system in a large design space, especially when the number of available samples is limited. This paper proposes an intuitive method to systematically reduce the design space to a relatively small region. This method entails three main elements: 1) constructing surrogate approximations using either response surface or kriging models to capture unknown systems in the original large space; 2) calculating many inexpensive points from the obtained surrogate model, clustering these points using the Fuzzy c-means clustering method, and choosing an attractive cluster and its corresponding reduced design space; 3) progressively generating sample points to construct kriging models and identify the design optimum within the reduced design space. The proposed method is illustrated using the well-known six-hump camel back problem. The method is then applied to a constrained, highly nonlinear optimization problem and a real design problem. After comparing with other methods, it is found that the proposed method can intuitively capture promising design regions and can efficiently identify the global or near-global design optimum with the presence of highly nonlinear constraints. The effect of using either response surface or kriging models in the original design space is also compared and discussed. Limitations of the proposed method are illustrated.

Keywords: Response Surface Method, Kriging, Metamodeling, Design Optimization, Fuzzy Clustering

[*] Assistant Professor, Department of Mechanical and Industrial Engineering, **Corresponding Author**, Tel: (204) 474 9463. Fax: (204) 275 7507. Email: gary_wang@umanitoba.ca

[†] Assistant Professor, Departments of Mechanical & Nuclear Engineering and Industrial & Manufacturing Engineering. Email: tws8@psu.edu

I. Introduction

Much of today's design involves the use of highly computation-intensive analyses and simulations. To incorporate such simulations in design optimization imposes daunting computational challenges since at least one function—the objective function or a constraint function—requires a computation-intensive process for function evaluation. For instance, one crash simulation of a full passenger car takes 36-160 hours to compute, according to engineers at Ford Motor Company.[1] To reduce the computational cost of computation-intensive design simulations and analyses, metamodeling techniques are currently being employed to develop inexpensive *surrogates* of these analyses and simulations. Many successful applications of metamodeling techniques in engineering design and other disciplines have been well documented, and recent reviews can be found in Refs. 2-4.

Optimization methods based on metamodels have many advantages over conventional optimization methods because the former aims to reduce the number of computation-intensive simulations as well as to find the optimum. Metamodeling techniques also facilitate parallel computation, but the efficacy of the metamodeling technique is constrained by the allowable number of computation-intensive simulations (i.e., function evaluations), since there is an inherent tradeoff between the accuracy and the number of samples used to build the metamodel. The unavailability of function derivatives, properties, etc. makes metamodel-based optimization difficult and also distinct from conventional optimization.

Among the various metamodeling techniques, the response surface method (RSM) and kriging attract the most attention. RSM originated from the formal design of experiment theory,[5,6] while kriging originated from geostatistics for field mapping.[7] Comparisons on the performance of these two types of methods and other metamodeling methods have been archived in Refs. 3,8-11. Generally speaking, RSM that employs low-order polynomial functions can efficiently model low-order problems, and the computation of a response surface model is quick and cheap. In addition, RSM facilitates engineering understanding by comparing parameter coefficients and also help eliminating unimportant design variables. Low-order polynomial response surfaces, however, are not good for highly nonlinear problems, especially those with "waving" behaviors.[8]

On the other hand, kriging models can accurately approximate unknown system even with high non-linearity, and the number of samples needed to fit a kriging model, theoretically, is lower than that for response surfaces.[12] However, the computational effort required to fit a kriging model is much greater, and the interpretation of kriging model parameters is not intuitive.[8] Kriging models can be used for screening, but the procedure is not as straightforward as it is with response surfaces.[13]

Regardless of which metamodeling technique is used for a specific problem, it is observed that the modeling efficiency and accuracy are directly related to the design space (i.e., the approximation space for metamodels). A branch of research under the metamodeling umbrella thus aims at developing methods that can gradually reduce the design space to improve the modeling accuracy. Two types of design space reduction schemes are seen in the literature. One is to reduce the dimensionality of the design space by reducing the number of design variables. The early work of Box and Draper[14] introduces a method to gradually refine the response surface to better capture the real function by "screening" out unimportant variables. The *variable-complexity response surface modeling* method uses analyses of varying fidelity to reduce the design space to the region of interest.[15,16] The reduction of dimensionality is very important as it aims to break the "curse of dimensionality"; however, dimensionality is difficult to reduce especially for multidisciplinary design problems.[12]

The other type of design space reduction seeks to reduce the size of the design space while assuming the dimensionality cannot be further reduced. Since the combined range of each design variable dictates the size of the design space, the larger the range for each design variable, the larger the design space; the larger the design space, the more difficult and costly to construct accurate metamodels. Engineers tend to give very conservative lower and upper bounds for design variables at the initial stage of setting up a design optimization problem. This is often due to the lack of sufficient knowledge of function behavior and interactions between objective and constraint functions at the early stage of problem definition. Chen and her co-authors[17] develop heuristics to lead the surface refinement to a smaller design space. Wujek and Renaud[18,19] compare a number of move-limit strategies that all focus on controlling the function approximation in a more "meaningful" design space. Many researchers advocate the use of a

sequential metamodeling approach using move limits[20] or trust regions[21-23]. For instance, the Concurrent SubSpace Optimization procedure uses data generated during concurrent subspace optimization to develop response surface approximations of the design space, which form the basis of the subspace coordination procedure.[24-26] More recently, Perez, *et al.*[27] present an adaptive experimental strategy for reducing the number of sample points needed to maintain accuracy of second-order response surface approximations during optimization. Osio and Amon[28] develop a multi-stage kriging strategy to sequentially update and improve the accuracy of surrogate approximations as additional sample points are obtained. Mathematically rigorous techniques and trust regions are also being combined to develop several other metamodel management frameworks to manage the use of approximation models in optimization.[29-32] Finally, Schonlau, *et al.*[33] describe a sequential algorithm to balance local and global searches using approximations during constrained optimization.

In recent years, the lead author and his colleagues have developed the Adaptive Response Surface Method (ARSM), which systematically reduces the size of the design space by discarding portions of it that correspond to objective function values larger than a given threshold value at each modeling-optimization iteration.[34-36] Thus, the design space is gradually reduced to the neighborhood of the global design optimum. The design space reduction procedure is characterized by its systematic consideration of the interrelationship between design variables. The improved ARSM applies inherited Latin Hypercube Designs (LHD) for sampling, which greatly reduced the number of samples required.[34] Tests on the ARSM prove the good accuracy of this method and efficiency in terms of the number of computation-intensive function evaluations required. The ARSM, on the other hand, bears some limitations. It assumes low-order polynomial constraints that can be obtained accurately by the RSM or are known to the designer. The space reduction is performed on the objective function and thus the space reduction converges to the unconstrained global optimum rather than constrained optimum. In addition, the ARSM employs $2n$ global optimization processes (where n is the number of design variables) to find the reduced design space. The space reduction process thus relies on the robustness of the underlying optimization algorithm.

In this paper, an intuitive strategy for design space reduction is proposed to extend our previous work in metamodeling. The proposed method will first gain knowledge of the function behavior in the original design space through low-accuracy metamodeling with a limited number of samples, based on which "regions of interest" in the design space will be identified through fuzzy clustering. Such a process can iterate until the design space is believed to be "irreducible" by design engineers. Then an accurate model for all of the necessary functions is progressively constructed using kriging. The global optimum can then be found based on more accurate metamodels in the reduced design space.

In the next section, background information on response surface methods, kriging models, and fuzzy c-means clustering are presented and the proposed methodology is introduced. In Section III, the proposed methodology is applied to two mathematical examples and an engineering application. Results from these examples, including comparisons with other methods, are discussed in Section IV. Closing remarks and future work are given in Section V.

II. Background and Proposed Methodology

A. Response Surface Method

The response surface method (RSM) is one of the *design of experiments* (DOE) methods used to approximate an unknown function for which only a few values are computed. The RSM stems from science disciplines in which physical experiments are performed to study the unknown relation between a set of variables and the system output, or response, for which only a few experiment values are acquired. These relations are then modeled using a mathematical model called a *response surface*.

The standard RSM first employs an experimental strategy to sample points in the design space, and then applies either a first-order model or a second-order model to approximate the unknown system. The first- and second-order models have forms as given in Eqs. 1 and 2.

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i \tag{1}$$

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \beta_{ii} x_i^{\,2} + \sum_{i<j} \sum_{j=1}^{n} \beta_{ij} x_i x_j \qquad (2)$$

where β_i, β_{ii}, and β_{ij} represent regression coefficients, $x_i, (i = 1 \cdots n)$ are design variables, and y is the response. Usually the design variables are normalized to [-1 1]. There are in total $(n+1)(n+2)/2$ number of coefficients in Eq. 2, thus a minimum of $(n+1)(n+2)/2$ number of sample points are needed to obtain these coefficients. Based on normalized variables, one can tell the relative importance of each term by comparing the magnitude of its coefficient. One can reduce the number of terms of a complete quadratic model beforehand if *a priori* knowledge of the model is available or some systematic methods such as the forward stepwise regression is employed. In addition, by deliberately fixing unimportant variables, one can reduce the dimensionality of the design space and reduce the number of samples required to fit the revised model (due to the reduced number of terms in Eq. 2). Systematic methods for "screening" are discussed in greater detail in Ref. 6. In this work, for the description convenience, we base our discussion on the model expressed by Eq. 2.

B. Kriging Models

Given observed $y_i = f(x_i)$ for $i = 1,\ldots, m$, we want to construct a global model $\hat{f}(x)$ of $f(x)$. If assuming that there is no uncertainty in the $y_i = f(x_i)$, it is then reasonable to require the surrogate function to interpolate these values, i.e., to require that $\hat{f}(x_i) = f(x_i)$. This assumption is valid for deterministic computer experiments for which random error is non-existent.

Consider a simple example of kriging by assuming that f is a realization of a stochastic process F with known mean $\mu(x) = 0$ and known covariance function $c(\cdot,\cdot)$, and that each symmetric matrix $C = [\, c(\cdot,\cdot)\,]$ is strictly positive definite.[37] Let

$$y = [f(x_1),\ f(x_2),\ ...,\ f(x_m)]' \qquad (3)$$

contain the sample values of the response. For each $x \in \Re$ define $b(x) \in \Re$ to minimize $E[y'b - F(x)]^2$. Then the *best linear unbiased predictor* (BLUP) of $f(x)$ is:

$$\hat{f}(x) = y'b(x) = y'C^{-1}c(x) \qquad (4)$$

where C is the symmetric positive definite n x n matrix $[c(x_i, x_j)]$, and

$$c(x) = [c(x_1, x),\ c(x_2,\ x),\ ...,\ c(x_m,\ x)]' \tag{5}$$

Notice for any $x = x_i$, $i = 1, \ldots, m$, $C^{-1}c = e_i$, the unit vectors [1, 0, …, 0], [0, 1, …, 0], etc., so that the kriging model interpolates the sample points as follows:

$$\hat{f}(x_i) = y'C^{-1}c(x_i) = y'e_i = y_i = f(x_i) \tag{6}$$

More generally, we assume the stochastic process F is a Gaussian process with $\mu(x) = a(x)'\beta$ and covariance $c(x_i, x_j) = \sigma^2 R(x_i,\ x_j)$. We assume that $a(x) : \Re \to \Re^q$ is a known function, that $\beta \in \Re^q$ is an unknown vector, and that $R(x_i,\ x_j)$ is an unknown element correlation function between two sample points x_i and x_j. Let A denote the m x q matrix $[a_j(x_i)]$; let **R** denote the symmetric m x m matrix $[R(x_i, x_j)]$, and let

$$r(x) = [R(x_1,\ x),\ R(x_2,\ x),\ ...,\ R(x_m,\ x)]' \tag{7}$$

Then, for fixed β and correlation function, the BLUP of $f(x)$ is

$$\hat{f}(x) = a(x)'\beta + (y - A\beta)'\mathbf{R}^{-1}r(x) \tag{8}$$

A kriging model represented by Eq. 8 is often considered as a combination of a "global" linear model with a "lack of fit" component that is a realization of a stochastic process. If we take a close look at the Eq. 8, the first term is the generalized least squares prediction while the second component "pulls" the generalized least squares prediction through the observed data points. The elasticity of the "pull" is solely determined by the correlation function $R(x_i, x_j)$. The predictions at the sampling points are exactly the corresponding observations, and the mean square error equals to zero. As a new point x_0 moves away from all the sample points, the second term of Eq. 8 approaches zero, yielding the generalized least squares prediction. The mean square error at that point increases. This observation is true if F is another stochastic process rather than Gaussian.[38]

By varying β and the correlation function, we define a family of interpolating functions from which we can select a specific $\hat{f}$ to model f. For a given correlation function, the *maximum likelihood estimates* (MLEs) of β and σ^2 have explicit formulas:

$$\hat{\beta} = (A'\mathbf{R}^{-1}A)^{-1}A'\mathbf{R}^{-1}y \tag{9}$$

and

$$\hat{\sigma}^2 = \frac{1}{n}(y - A\hat{\beta})'\mathbf{R}^{-1}(y - A\hat{\beta}) \tag{10}$$

In this work, we employ a Gaussian correlation function of the form:

$$R(x_i, x_j) = \exp[-\sum_{k=1}^{n} \theta_k \mid x_i^k - x_j^k \mid^2] \tag{11}$$

where n is the number of design variables, θ_k are the unknown correlation parameters used to fit the model, and x_i^k and x_j^k are the k^{th} components of sample points x_i and x_j.

To compute $\hat{\theta}$, the MLE of θ, the following equation must be maximized:

$$n\log(\sigma^2) + \log\det\mathbf{R} \tag{12}$$

as a function of θ. Once $\hat{\theta}$ is obtained, one can substitute $\hat{\theta}$ into Eqs. 8-11 to build a kriging model. While any values for θ create an interpolative model, the "best" kriging model is found by solving the n-dimensional unconstrained non-linear optimization problem given by Eq. 12. It has been found that a constant term β can yield a satisfactory "global" model;[38, 39] consequently, we use a constant β in this work. It is also found that in some cases using a single correlation parameter gives sufficiently good results;[40-42] however, we employ n θ values here.

C. Fuzzy c-means Clustering

Clustering of numerical data forms the basis of many classification and system modeling algorithms. The purpose of clustering is to identify natural groupings of data from a large data set to produce a concise representation of a system's behavior.[43] *Fuzzy c-means* (FCM) is a data clustering technique originally introduced by Bezdek[44] as an improvement on earlier clustering methods. It provides a method for grouping data points that populate some multidimensional space into a specific number of different clusters. The algorithm is based on the minimization of the overall dissimilarity of all cluster data. When a cluster number c is selected, the algorithm can automatically identify cluster centers and distribute all given data sets into the appropriate data cluster.

For a specified number of sub-sets, c, the overall dissimilarity between each data point and each fuzzy prototype can be represented as:

$$J_{cd}(U,\vec{v}) = \sum_{k=1}^{m}\sum_{i=1}^{c}(u_{ik})^{d} \parallel x_k - v_i \parallel^2 \tag{13}$$

where U is a fuzzy c-partition of the m data points x_k (k = *1, …, m;* $x \in \Re^n$), $\vec{v} = (v_1, v_2, \cdots, v_c)$, with v_i is the i^{th} cluster center or prototype, $1 \le i \le c$; d is a constant greater than 1 (typically $d = 2$); and u_{ik} is the *degree of membership* of the kth data in the ith cluster,

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{\parallel x_k - v_i \parallel}{\parallel x_k - v_j \parallel}\right)^{2/(d-1)}} \tag{14}$$

The optimal solution U^*, $\vec{v}^*$ is obtained by:

$$\min_{U,\vec{v}} J_{cd}(U, \vec{v}) \tag{15}$$

The fuzzy c-means clustering algorithm always converges to a strict local minimum of J_{cd}. Given a group of data points and the number of clusters, the cluster centers $\vec{v}^*$ and the *degree of membership* of each point to each cluster u_{ik} (saved in U^*) is computed. It is easy to find that for a given x_k, $0 \le u_{ik} \le 1$ and $\sum_{i=1}^{c} u_{ik} = 1$.

In this work, the fuzzy logic toolbox provided by Matlab™ is used. The fuzzy-clustering method is used in this work due to its simplicity, robustness, and convenience. The proposed methodology does not dictate the exclusive use of fuzzy clustering, and other clustering methods may be equally acceptable. For a review of other clustering methods, please refer to Ref. 45.

D. Proposed Methodology

The proposed methodology is based on the observation that design engineers tend to give very conservative lower and upper bounds for design variables at the initial stage of setting up a design optimization problem. This is often due to the lack of sufficient knowledge of function behavior and interactions between objective and constraint functions at the early stage of

problem definition. This work thus proposes a hierarchical metamodeling method, by which more insights on the design problem can be gained in the initial design space. Then based on the obtained model, a more attractive design space is obtained using the fuzzy clustering method. In the reduced design space, Latin Hypercube Design (LHD) samples are gradually added to construct a kriging model, and optimization is performed. This process will converge, with a high probability, to the global optimum. The proposed methodology entails three major steps as shown in Figure 1 and described in the following sections.

Insert Figure 1 about here.

Step 1. Construct quadratic polynomial response surface or kriging models in the original design space of the computation-intensive function(s)

When dealing with a computation-intensive design optimization problem, quadratic polynomial response surface models are very useful for screening design variables by studying the coefficients of the response surface. The response surface is also cheap to compute. However, RSM can only roughly approximate sample points; it cannot interpolate samples to obtain an accurate model. The kriging model is very good to model complex functions, and it interpolates all of the sample points, which is ideal for computer experiments. It requires fewer samples to fit a low-order polynomial function;[12] however, interpreting a kriging model is not intuitive,[46] and they are expensive to compute and sensitive to ill-conditioned matrices.[10,47] At Step 1, the objective is to capture the global behavior of the unknown function, in preparation for the space reduction. The accuracy of the model is less of concern, unlike in Step 3. Therefore, either response surfaces or kriging models can be used for this step, and a comparison of these two methods is provided in Section IV.

This step will help to build a global view of the function in a relatively large design space. Samples can be generated in parallel. If a quadratic model is used, one can apply the conventional design of experiments methods (e.g., analysis of variance) to study the importance of each design variable and the residuals. If kriging is used, one can perform the screening using the method discussed in Ref. 13. Based on these results, unimportant design variables can be "screened out," and the goodness of the fit can be obtained. Then RSM can be applied again based on the reduced dimensionality until the design space cannot be reduced any further. If the

unknown function happens to be first- or second-order, then the response surface model would be very accurate, and the metamodeling process can terminate here. To identify this behavior, generally twice the minimum number of required samples, $(n+1)(n+2)/2$, is used, where n is the number of variables. By using portions of the sample points to fit the model and the rest for validation, one can estimate the accuracy of the metamodel. In this work, Latin Hypercube Designs (LHDs) are used to sample the design space since they are uniformly distributed along each variable direction and are inheritable.[34,48]

Step 2: Generate inexpensive points from the surrogate, cluster the points, and identify the attractive cluster

After Step 1, if the unknown function proves to be complex and cannot be accurately fit, Step 2 and 3 are applied. Using the metamodel(s) from Step 1, we first generate many points using the surrogate to form a point set. These model points could be simple grid points, LHD points, orthogonal array points, Hammersley samples, or any other uniform distributed samples.[49] Since a surrogate function is cheap to compute, the cost for generating these points is negligible. Then a threshold function value is chosen to eliminate points at which the function value is larger than the threshold (in the case of minimization) to reduce the number of points used for clustering.

Clustering and Identification of the Attractive Cluster

Based on the remaining point set, the fuzzy c-means clustering method is applied to identify an attractive region in the design space. For this work in particular, we concern ourselves only with the standard optimization problem with only one objective function, which we want to minimize. The use of metamodels for other types of optimization problems such as multiobjective optimization, robust design optimization, and so on are discussed elsewhere in the literature.[3,47]

One difficulty of the fuzzy c-means method is that the number of clusters is required as an input, and the number of clusters will be unknown for an unknown function. Clustering methods such as subtractive clustering can estimate how many clusters in a point set as long as the radius of influence of each cluster is given, but the number of clusters is very sensitive to the radius of influence, which is also unknown.[50,51] Other methods to identify the number of clusters are described in Ref. 52, but many of these methods demand additional computational effort and

specific stopping rules. In this work a constant number of clusters, $c = 2$, is chosen. It is based on following reasons:

1) The surrogate at Step 1 is obtained with a limited number of samples, because the goal of Step 1 is to capture the rough behavior of the real function rather than achieving high modeling accuracy. Then efforts required to accurately identify the number of clusters on a non-accurate surrogate model are not justifiable. Only with high-accuracy models, the identification of all the clusters would be meaningful.
2) What we need is to identify a region with low function values for further high-accuracy approximation and discard regions with high function values. Thus, $c = 2$, is chosen for simplicity, which is also sufficient for the purpose of design space reduction.

For any design function with $x \in \Re^n$, we have $\hat{f}(x)$ based on the metamodel, which we treat as an additional component of x during clustering. Thus, for a $n = 1$ problem, we have a 2-D clustering problem. By doing so, clustering can identify a concentrated region in which all points have relatively low function values. For example in a $n = 1$ case, the attractive cluster entails a cloud of points that are close to each other in both x direction (concentrated region) and the f direction (all having low function values). Let $f(x)$ be the $(n+1)$ component of design variable $x \in \Re^n$, and define the extended x vector as $x^e \in \Re^{n+1}$. With $c = 2$, Eq. 13 becomes:

$$J_{cd}(U, \vec{v}) = \sum_{k=1}^{m} \sum_{i=1}^{2} (u_{ik})^d \parallel x_k^e - v_i \parallel^2 \tag{16}$$

Once the clusters, along with the *degree of membership* of each point to each cluster, u_{ik}, are identified, the points can be grouped into two different clusters. If the *degree of membership* of a point to a cluster is larger than 0.5, the point is grouped with that cluster.

By comparing the minimum, median, mean, and standard deviation of the function values of the points in a cluster, an attractive cluster can be identified. If the function values in one cluster have a smaller minimum, mean, median and standard deviation than that of the other, this cluster is considered more attractive. Otherwise, which may not be often, one has to continue repeating the steps in both regions defined by the clusters. If an attractive cluster is identified, we can use

it to identify a reduced design space for further exploration. The reduced space is defined by the minimum and maximum bounds of all the points within the attractive cluster.

To illustrate the clustering process, let us assume a model is obtained after the first fitting for a $n = 1$ problem as shown in
Figure 2. Symbols 'x' and 'o' indicate a model point falling into different clusters. If the solid line is chosen as the threshold (cutting value), points above this line are deleted from the original point set. After clustering, two clusters enclosed by dotted lines can be identified. It is to be noted that since the function value, f, is treated as one of the x component participating in the clustering, the fuzzy clustering algorithm always identifies "spatially" close points. As a result, the space around the global minimum (with low function values) can be represented by one of the clusters, which is the lower cluster in Figure 2.

Insert Figure 2 about here.

The Role and Choice of Threshold

The threshold is used to reduce the number of points and speed up the process of clustering. For example, setting a high threshold value during minimization can immediately eliminate points that are far from the optimum or are infeasible if a penalty function is combined with the objective function to ensure constraint satisfaction. As seen in Figure 2, the threshold can be any value between its upper and lower limits. The sensitivity study of the threshold with test problems in later sections further proves this observation.

The threshold is chosen based on the distance between the maximum and minimum function values from the original point set. It is easy to find the minimum and maximum function values, *fMin* and *fMax*, respectively, from a point set by a simple sorting procedure. Assume $fRatio \in [0\ 1]$, the threshold value, *fCut*, is chosen as:

$$fCut = fMin + (fMax - fMin)*fRatio \tag{17}$$

When the response surface fitting is very poor, a conservative *fRatio* (>0.5) is chosen; otherwise, a smaller number (<0.5) is chosen.

Step 3. Progressively generate samples in the reduced space to build kriging model and searching for the global optimum

Steps 1 and 2 can be repeated until a design space that the designer believes to be "irreducible" is reached. This decision is based on engineering experience, the insight gained from the metamodels during earlier iterations, and the affordable time and resources. If time and computation resources are not tight, one might start this step in a relatively large design space; otherwise, one might push the limit to a very small area.

In the new design space, the sample points generated during Step 1 are inherited as long as they fall into the new space, and new sample points are generated. The new samples and the old samples together form a modified LHD sample set. For example, as illustrated by Figure 3, we assume that three points are inherited into the new space, represented by the large square defined by x_1 and x_2. If six points are desired in this space to form a LHD set, then three more LHD points have to be generated. The inheritance method will first independently generate a LHD sample set of three points as shown in the small square defined by x_1' and x_2'. It then searches the new design space, identifying the variable intervals that are not represented by inherited sample points. Their positions are recorded. Then the new three LHD points are mapped back to the space to fill in the variable intervals that are not represented. For example, Point P, which is at the *third* interval of x_1' and the *second* interval of x_2', is mapped to the space as P_m at the *third* underrepresented interval of x_1 and the *second* underrepresented interval of x_2. The other two points are mapped in the same manner. As a result, these six points together form a LHD sample set as each interval of any design variable is represented by one point. At later iterations, the design space remains the same and more samples are desired. In that case, the design space is divided to more intervals. Sample points from the previous iteration will not represent all of the variable intervals since the number of intervals has increased. The inheritance method will follow the same procedure as the first inheritance; for more details on inherited LHDs, please refer to Ref. 34.

Insert Figure 3 about here.

Since high non-linearity and many local optima may still be present in the reduced design space, the quadratic polynomial function model will have difficulty at this step. Instead, a kriging model is constructed with the formed LHD sample set. Simulated annealing (SA) method or any

other stochastic global optimization algorithm can be applied to identify the global optimum based on the kriging model.[53] The real function value of the model optimum can be found by calling the original computation-intensive simulation/analysis. This model optimum, along with all the existing points in the design space, is inherited to a new iteration when more sampling points are desired in order to get a more accurate model. It is to be noted at this step, the design space remains the same but the number of samples increases gradually. If in two consecutive iterations, the obtained optimal values are very close and the model prediction is very close to the real function value at the obtained optimum, the iteration terminates. Through this progressive process, the number of samples will be kept small, and the design optimum can be accurately secured.

III. Example Problems

In this section, the proposed methodology is applied to the six-hump camel back problem, a highly nonlinear, constrained optimization problem, and a real design problem. Detailed results and discussion for all three problems follow in Section IV.

A. The Six-Hump Camel-back Problem – Description and Results

The six-hump camel-back (SC) function given by Eq. 18 is a well-known example for testing global optimization algorithms.[54]

$$f_{sc}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4, \;\; x_{1,2} \in [-2, 2] \tag{18}$$

As the name implies, SC has six local optima with the global optimum equals to -1.0316 at (0.0898 -0.7127) and (-0.0898 0.7127). Let us assume that the SC function is unknown to the designer and that the function value can only be obtained through computation-intensive analysis. Therefore, the goal is to identify the global design optimum with a small number of function evaluations, and the following demonstrates implementation of the proposed methodology for this example using a response surface model and LHD for Step 1.

Step 1: Modeling SC in the original design space

For two variables, the minimum number of samples required to fit Eq. 2 is six as defined by the number of coefficients in the quadratic response surface model. To ensure a good sampling, the

number of samples is doubled at this step, and twelve LHD samples are generated in the design space [-2 2] for x_1 and x_2.

The two variables are normalized to [0 1], and then a second-order polynomial function is fit:

$$f = 23.875-16.680*x1-100.772*x2+31.806*x_1^2+115.862x_2^2-28.589*x_1*x_2 \quad x_{1,2} \in [0,\ 1] \qquad (19)$$

Step 2: Clustering and design space reduction

Based on the surrogate defined by Eq. 19, in total 121 uniformly distributed points are generated. The fuzzy c-means clustering was applied by setting the number of desired clusters as two. The cutting objective function value is taken as 16.96 by setting the ratio to 0.5. The remaining points are clustered to two groups. The concerned properties of the two clusters are summarized in Table 1. It is easy to see that Cluster 1 represents a more attractive design region than Cluster 2 since its four properties minimum, mean, median, and standard deviation have smaller values than those of Cluster 2. The new design space is thus reduced to x_1 [-2 2], x_2 [-1.2 1.2]. If one is not satisfied with the design space, Step 1 and Step 2 can be repeated to further reduce the design space. In this testing, the reduced space is accepted as the final design space.

Insert Table 1 about here.

Step 3. Searching for the global optimum in the reduced space

First, all of the samples points used to generate the first polynomial metamodel that fall into the new design space are inherited. For the SC problem, the sampling-modeling-optimization iterations are recorded in Table 2. In Table 2, the first row is the number of samples used to obtain the kriging model at each iteration, bearing in mind that each group of samples is a combination of inherited and new samples. The second row records the number of new samples generated at that iteration to form a LHD sample set. The third row is the total function evaluations up to that iteration, which includes the samples used to build the polynomial function in Step 1. The total number of function evaluations (samples) indicates the extent of computational cost for all of the computation-intensive analyses invoked during the optimization. At Iteration 4, the obtained model optimum of -1.0240 is very close to the one obtained at Iteration 3, which is -1.0242; it is also very close to the real function value, -1.0303. Therefore,

the process terminates. The obtained optimum value of -1.0303 is very close to the analytic solution, -1.0316, with an error of 0.13%. The optimal design (-0.0774 0.7211) is also close to the analytical solution (-0.0898 0.7127). Given the high accuracy, the total number of function evaluations is only 45.

Insert Table 2 about here.

The final obtained kriging model is plotted in comparison to the original SC function in the reduced design space x_1 [-2 2], x_2 [-1.2 1.2]. In addition to yielding one global optimum, the kriging model is also able to approximate complex functions well. As we can see from Figure 4, though the kriging model did not map exactly all six "humps" in the SC function, it does capture five "humps" at the neighborhood of the analytical local optima. Additional studies using this example are discussed in Section IV.

Insert Figure 4 about here.

B. A One-dimensional Constrained Optimization Problem – Description

The presence of constraints further complicates any optimization problem, especially when the constraints are computation-intensive as well. If the objective function and constraints are to be approximated using metamodels to save computation time/cost, finding the best tradeoff between modeling and optimization is a challenging issue. If either the objective function or the constraint is a second-order polynomial function in nature, a standard response surface method can be applied to find the intrinsic model accurately for the objective or constraint. In that case, one only needs to concentrate on the functions that cannot be accurately modeled by a low-order polynomial function. Thus the problem can be simplified, but in many cases, this is not possible.

In this work, we consider the situation where both the objective and constraint functions are highly nonlinear and computation-intensive. In order to find the constrained optimum, accurate metamodels for both functions are demanded. To obtain accurate metamodels for both functions in a large design space, the number of samples (function evaluations) will be large. The proposed methodology, in combination with a penalty function approach, provides a method to eliminate design regions in which constraints are most likely violated. Therefore, a smaller

design region can be focused upon, in which accurate metamodels for both objective and constraints can be constructed at a reduced computational cost. This method is demonstrated using the following example:

$$\begin{aligned} &\min \quad f(x) = e^{-2x}\sin(5\pi x) \\ &\text{subject to} \\ &\quad g_1(x) = e^{-4x}\cos(3\pi x) \le e^{-2x}\sin(5\pi x) \\ &\quad g_2(x) = -0.8x + 0.6 \le e^{-2x}\sin(5\pi x) \\ &\quad x \in [0, 1] \end{aligned} \tag{20}$$

This problem is a one-dimensional constrained (OC) problem, which is chosen for the ease of visualization (see

Figure 5). The optimization problem is complex since there are highly nonlinear functions in both the objective and the constraints. Because of the highly entangled non-linearity, many local constrained optima exist as can be seen in

Figure 5.

Insert

Figure 5 about here.

To ensure feasibility, the constraints are aggregated to the objective function by virtue of the penalty factor, λ. Thus the optimization problem is transformed to the following unconstrained optimization problem:

$$\min \quad Af(x) = f(x) + \lambda \sum_{i=1}^{2} [< g_i(x) - f(x) >]^2$$

where

$$< g_i(x) - f(x) > = \begin{cases} g_i(x) - f(x) & for \quad g_i(x) \geq f(x) \\ 0 & for \quad g_i(x) < f(x) \end{cases}$$

$$f(x) = e^{-2x} \sin(5\pi x) \tag{21}$$
$$g_1(x) = e^{-4x} \cos(3\pi x)$$
$$g_2(x) = -0.8x + 0.6$$
$$x \in [0, 1]$$

where the original objective function defined in Eq. 20 is aggregated by a term that "penalizes" it whenever constraints are violated. The new objective function is thus referred to as the aggregated function and is also plotted in

Figure 5 when $\lambda = 50$ with high peaks being chopped off. In the testing of OC, function $g_2(x)$ is perfectly fit at Step 1 and functions $f(x)$ and $g1(x)$ are continued to be approximated. The aggregated function $Af(x)$ participates in the clustering instead of $f(x)$. Results for this example are discussed in Section IV.

C. The Sandwich Beam Problem – Description

The third example involves the design of a sandwich beam (SB) developed by Messac.[55] We are to design the sandwich beam shown in Figure 6 to support a vibrating motor. The beam consists of three layers of different materials: the mass density (ρ_i), Young's Modulus (E_i), and cost per unit volume (C_i) for each of the three material types are provided in Table 3.

Insert Figure 6 about here.

Insert Table 3 about here.

The design objective is to minimize the cost while satisfying fundamental frequency and other constraints. The cost function, f_c, and frequency function, f_f, are described as follows:

$$f_c = 2bL[c_1 d_1 + c_2(d_2 - d_1) + c_3(d_3 - d_2)]$$

$$f_f = \frac{\pi}{2L^2}\sqrt{\frac{EI}{\mu}}$$

where:

$$EI = \frac{2b}{3}\left[E_1 d_1^3 + E_2(d_2^3 - d_1^3) + E_3(d_3^3 - d_2^3)\right]$$

$$\mu = 2b\left[\rho_1 d_1 + \rho_2(d_2 - d_1) + \rho_3(d_3 - d_2)\right]$$

Several geometric constraints are imposed: an upper bound for the total mass of the beam; minimum thickness for layers two and three; and ranges for each geometric parameters, d_1, d_2, d_3, b, and L. The particular problem instantiation used for this study is:

$$\begin{aligned}
&\min f_c = 2x_4 x_5(c_1 x_1 + c_2 x_2 + c_3 x_3) \\
&\text{Subject to} \\
&f_f = \frac{\pi}{2x_5^2}\sqrt{\frac{EI}{\mu}} \geq 150,\ EI = \frac{2x_4}{3}[(E_1 - E_2)x_1^3 + (E_2 - E_3)(x_2 + x_1)^3 + E_3(x_1 + x_2 + x_3)^3] \\
&\mu = 2x_4(\rho_1 x_1 + \rho_2 x_2 + \rho_3 x_3) \\
&x_{2,3} \geq 0.01 \\
&\mu x_5 \leq 2700 \\
&x_1 \in [0\ 1],\ x_2 \in [0\ 0.05],\ x_3 \in [0\ 0.05],\ x_4 \in [0.5\ 2],\ x_5 \in [3\ 10]
\end{aligned} \tag{22}$$

where x_1: d_1, x_2: (d_2-d_1), x_3: (d_3-d_2), x_4: b and x_5: L. Based on Eq. 22, the initial design space is:

x_1: [0 1] x_2: [0 0.05] x_3: [0 0.05] x_4: [0.5 2] x_5: [3 10]

In the test of the SB problem, the constraints are also aggregated with $\lambda = 100$ for the frequency constraint. Results for this example are discussed in the next section.

IV. Results and Discussion

All three examples have been randomly chosen to test the proposed methodology. These examples include both unconstrained and constrained optimization problems, having highly nonlinear functions and multiple local optima. The solution strategy follows exactly the process illustrated in Figure 1. To demonstrate the flexibility in generating inexpensive points from the

surrogate, grid points are used for the first two examples while LHD samples are used for the sandwich beam problem since grid points are not ideal for high dimensional problems since the number of points grows exponentially. The results are presented and discussed as follows.

A. Comparison of the Proposed Methodology with Methods Involving No Space Reduction

To test the proposed methodology, it should first be compared with metamodeling strategies that do not involve design space reduction. As a fair comparison, a K-All strategy is chosen wherein the metamodeling and optimization are carried out in the original design space using inherited LHD points with no design space reduction. In the K-All strategy, the kriging model is built and used iteratively, following a procedure identical to Step 3 of the proposed method. For the proposed methodology, both kriging and quadratic polynomial function models are used at Step 1 where KCK denotes Kriging-Clustering-Kriging and PCK denotes Polynomial-Clustering-Kriging for Steps 1, 2, and 3, respectively. Table 4 lists the comparison of the optimization results for these three different strategies, K-All, KCK, and PCK for all three examples.

Insert Table 4 about here.

From Table 4, it can be seen that these three strategies converge to almost the same optimum for the respective test problems. For SC and OC, the obtained optimum is the analytical global optimum; while for SB, the obtained optimum is close to the analytical global optimum, f = 320.77 at (0.168 0.01 0.01 0.5 3.0). Secondly, both KCK and PCK, in general, require much fewer real computation-intensive function evaluations than the K-All strategy to reach the global optimum. For the SB problem, K-All needs fewer points than KCK. It is because that the SB is a constrained problem and the density of sampling points is coarse. The KCK process converges to a near optimum solution rather early with only 78 function evaluations, but the frequency constraint is slightly violated until 180 function evaluations. Sampling points generated during the K-All process, however, yield the optimum satisfying the frequency constraint. If assuming $\pm 5\%$ of the constraint, i.e., $f_f \geq 150(1 \pm 5\%)$, is allowed in engineering practice, in that case K-All yields f_c = *381.5* with 109 function evaluations and KCK can stop at f_c = *370.2* with 78 evaluations, in which case KCK performs better for the SB problem. In terms of the number of design iterations, no significant variation is observed for all three strategies. It is to be noted that

for both KCK and PCK, the number of design iterations in Table 4 includes the first iteration in the original design space.

B. Comparison of Kriging and Quadratic Polynomial Response Surfaces in Step 1

To further explore which modeling technique might be better suited for Step 1, a more detailed study is performed using all three examples, and the results are summarized in Table 5. The number of metamodel points generated for the clustering, shown in the first row of Table 5, is the same for both strategies for all three test problems. The reduced design spaces are very similar for SC and OC, while for SB the total reduced area is about the same for both methods. The third row is the minimum and maximum function values of the selected set of metamodel points for clustering. This information is given as a reference for the sensitivity of the threshold value to the space reduction. The last row lists the range of cutting values that lead to the "same" reduced space as recorded in the second row of each column, using a value of 0.1 to differentiate between the variation in the normalized upper and lower bounds of any two design variables.

Insert Table 5 about here.

Based on Table 4 and Table 5, both KCK and PCK work well for all three problems. The PCK strategy needs slightly fewer function evaluations than KCK for these three examples. They both yield similar reduced design spaces with a fairly big range of threshold values, but no method is better based on the choice of threshold. Based on this comparison with these examples, there is no clear-cut choice as to which metamodeling method should be used during Step 1. If the designer wants to "screen off" design variables, then the polynomial response surface model would be a better choice. If the computation cost is more of a concern or if the dimensionality is known to be irreducible, then kriging might be used instead to reduce the number of samples needed to approximate the unknown function(s).

C. Sensitivity of the Threshold

The sensitivity of the design space reduction to the threshold value is also examined and is listed in the fourth row of Table 5. It can be seen that to obtain the reduced design space, the range of threshold values is rather large. For example, for the SC problem using KCK, the cutting value can be any value from 18 to 35, and the fuzzy c-means clustering will still yield the same

reduced design space defined by x_1:[-2 2], x_2:[-1.2 1.2]. For the SB problem using PCK, the range of cutting values can be as large as [4e3 1e6], and the reduced space is about the same with variation on any bounds less than 0.1. This study indicates that the space reduction is insensitive to the choice of the threshold value. This can be generally true because if the threshold is relatively conservative, the threshold only changes the size of the undesired cluster (with high function values) and has little influence on the desired cluster (with low function values) as seen in Figure 2. It also should be noted that if the threshold is too small (i.e., close to *fMin*), the reduced design space will be too small and one risks missing good regions.

D. Metamodeling for Constrained Problems

To explore the influence of the constrained optimization problem, additional tests are performed on the SB problem. First, second-order polynomial functions are obtained using the response surface method with 42 Latin Hypercube samples for both the objective function (cost) and the nonlinear constraint function (frequency). The fraction of the variation in the sample data is captured by the R^2 value. The R^2 values are 0.9981 and 0.9700 for f_c^m and f_f^m, respectively, where f_c^m and f_f^m denote the metamodels for f_c and f_f. The average differences between the predicted values and the real values at the sampling points for these two functions are 144.42 and 15.44, respectively. If f_c^m and f_f^m are used in the optimization model described by Eq. 22, the obtained global optimum is at [1, 0.05, 0.01, 0.5, 3] with the real cost value of \$1,749 and a frequency of 787.4 Hz, which is far from the known global optimum. However, if we minimize the cost in the given design space without imposing any constraints, then the optimum based on f_c^m is the same as the real f_c, though the obtained cost values differ significantly. If f_c^m and the real frequency function f_f are used in Eq. 22, the obtained optimum is also far from the real optimum of Eq. 22. These trials indicate that for *unconstrained* optimization problems, the optimum could be obtained as long as the surrogate approximates the real function well, i.e., R^2 value is large. However, for *constrained* optimization problems with surrogate objective functions, no matter whether the constraints are approximated or not, the solution obtained can be erroneous even when R^2 value is very high as seen in this case with R^2=0.9981. Only when R^2 is extremely close to 1 and the average fitting difference on the sampling points are close to zero, can the constrained optimization problem be solved accurately.

E. Improvements to the Adaptive Response Surface Method

As mentioned in Section I, the Adaptive Response Surface Method (ARSM) applies a threshold gradually on the original design space.[34,36] The reduced design space is identified by searching the two extreme values of each x component on the hyper-plane formed by the intersection of the cutting plane and the original function. The ARSM features the systematic reduction of design space considering the interrelationship between all design variables, providing an efficient and effective search method for the global optimum. However, ARSM makes an assumption that the constraints are low-order polynomials that can be approximated accurately by a response surface or are known to the designer. This assumption is made because the ARSM always uses the polynomial function to approximate functions in any region. The polynomial function does not interpolate all the sampling points but approximates the overall trend of the function. If objective and constraint functions of high non-linearity are both approximated by polynomial functions, the error at the constrained optimum may be high. In addition, because the ARSM uses quadratic polynomial functions and design space reduction is based on the unconstrained objective function, it always gives the best modeling accuracy in the convex region that entails the unconstrained global optimum. If the constraints are active in a convex region other than that of the unconstrained global optimum, the ARSM may yield a mediocre solution.

The proposed methodology inherits the merits of the ARSM in that Steps 1 and 2 can be used to gradually and efficiently reduce the space to an "irreducible region." More importantly, it overcomes several of the ARSM's limitations. First, the proposed methodology makes no assumption on the form of the constraints. As demonstrated with the OC and SB problems, it can deal with highly non-linear constraints, and these constraints can be approximated functions as well. Secondly, because kriging is used repeatedly in the last "irreducible region," the methodology converges to the constrained optimum rather than the unconstrained optimum. Thirdly, the proposed methodology uses an intuitive and reliable method, namely, fuzzy clustering, to reduce the design space. In light of these findings, the proposed methodology is a significant improvement over ARSM and has much stronger potential to be used in real practice.

F. Limitations of the Proposed Method

The proposed method is essentially an intuitive method. It is based on the assumption that the initial design space is usually large, given the engineer' limited understanding of the design problem at the beginning of design optimization. The sampling in the initial design space should give a reasonable coverage of the entire space so that the model built at Step 1 can be used to locate the attractive sub-region in the design space. What is the minimum number of sample points in the initial sampling, unfortunately, is problem dependent. At Step 3, other methods such as the one in Ref. 33, might be used in substitution of the proposed Step 3 strategy. The essences of this work include the proposition of the fuzzy clustering method to reduce the design space, and the use of a framework of hierarchical metamodeling for design optimization. It does not dictate the exclusive use of any sampling method, modeling method, or combination of both. In stead, engineers can gain more benefits by integrating new, better, methods into the two essential ideas proposed in this work.

V. Closing Remarks and Future Work

To explore an unknown system using metamodels, engineers are always faced with the challenge of the tradeoff between the number of expensive samples and the accuracy of surrogate model. The logical process is to build an accurate metamodel first and then perform optimization on the premise of a good metamodel. However, to accurately model a complex system across a relatively large design space, the number of required samples (i.e., function evaluations) might be prohibitive. The number of samples is also related to the granularity of the approximation, i.e., the minimum distance between sample points. The granularity determines the metamodeling accuracy and thus the optimization accuracy. One way to further reduce the number of samples while increasing the optimization accuracy is to discriminate the original design space. That is to say, some regions will have small granularity and others have larger granularity. Any such discrimination strategy, under the constraint of limited number of samples, would most likely be intuitive. The proposed methodology in this work is one such intuitive method. It learns from a metamodel of large granularity and then focuses on a small region to build a metamodel of small granularity, from which an accurate optimum can be obtained. The contributions from the proposed methodology are summarized as follows.

- The fuzzy-clustering based method for space reduction. The combination of screening of points using the threshold and the point clustering leads to a smaller design space. By applying a threshold, areas violating constraints can be excluded from the point set before clustering. As a result, the constraints are more likely inactive in the reduced space and the optimization problem can be simplified. The clustering method is intuitive, conceptually simple and computationally reliable and inexpensive. The choice of threshold is proved to be insensitive.
- The hierarchical metamodeling strategy. This strategy distinguishes between the task of learning in a large design space and creating metamodels with high accuracy in a small design space. At the learning phase, the conventional RSM can be used for variable screening and function behavior learning. Kriging models can also be used at this phase if the number of function evaluations is more of concern or the function is known not to be a low-order polynomial. In the reduced design space, kriging is used for accurate metamodeling in order to solve constrained optimization problems. The progressively sampling strategy with inherited LHD samples ensures consistently improving kriging models in a given design space. The total number of samples can also be kept small.

It is found through the testing of the proposed methodology that a constrained optimization problem demands very accurate surrogates of both the objective function and constraint functions. Any discrepancy between the model and the real objective and/or constraint function will lead to an erroneous solution. Similar results have also been observed during multi-objective optimization using metamodels.[47] Finally, a reliable global optimization algorithm is needed to solve the final optimization problem with accurate kriging models, especially when n is larger than ten. Tests of the proposed methodology on large-scale design problems with n larger than 10 might also be helpful, and future work might include studying the use of other types of metamodels and metamodeling strategies within the proposed methodology.

Acknowledgement

The first author is funded through the Natural Science and Engineering Research Council (NSERC) of Canada. The second author acknowledges support from the National Science Foundation under NSF Grant DMI-0084918.

References

[1]Gu, L., "A Comparison of Polynomial Based Regression Models in Vehicle Safety Analysis," *ASME Design Engineering Technical Conferences - Design Automation Conference (DAC) (Diaz, A., ed.),* Pittsburgh, PA, ASME, September 9-12, 2001, Paper No. DETC2001/DAC-21063.

[2]Haftka, R., Scott, E. P. and Cruz, J. R., "Optimization and Experiments: A Survey," *Applied Mechanics Review,* Vol. 51, No. 7, 1998, pp. 435-448.

[3]Simpson, T. W., Peplinski, J., Koch, P. N. and Allen, J. K., "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," *Engineering with Computers,* Vol. 17, No. 2, 2001, pp. 129-150.

[4]Sobieszczanski-Sobieski, J. and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Structural Optimization,* Vol. 14, No. 1, 1997, pp. 1-23.

[5]Box, G. E. P., Hunter, W. G. and Hunter, J. S., *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*, John Wiley & Sons, New York, 1978.

[6]Myers, R. H. and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, New York, 1995.

[7]Matheron, G., "Principles of Geostatistics," *Economic Geology,* Vol. 58, 1963, pp. 1246-1266.

[8]Jin, R., Chen, W. and Simpson, T. W., "Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria," *8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization,* Long Beach, CA, AIAA, September 6-8, 2000, AIAA-2000-4801.

[9]Lin, Y., Krishnapur, K., Allen, J. K. and Mistree, F., "Robust Concept Exploration Method in Engineering Design: Metamodeling Techniques and Formulations," *ASME 2000 Design Engineering Technical Conferences - Design Automation Conference (Renaud, J. E., ed.),* Baltimore, MD, ASME, September 10-13, 2000, Paper No. DETC-2000/DAC-14283.

[10]Meckesheimer, M., Barton, R. R., Simpson, T. W., Limayem, F. and Yannou, B., "Metamodeling of Combined Discrete/Continuous Responses," *AIAA Journal,* Vol. 39, No. 10, 2001, pp. 1955-1959.

[11]Simpson, T. W., Mauery, T. M., Korte, J. J. and Mistree, F., "Comparison of Response Surface and Kriging Models for Multidisciplinary Design Optimization," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization,* St. Louis, MO, AIAA, Vol. 1, September 2-4, 1998, pp. 381-391.

[12]Koch, P. N., Simpson, T. W., Allen, J. K. and Mistree, F., "Statistical Approximations for Multidisciplinary Optimization: The Problem of Size," *Special Multidisciplinary Design Optimization Issue of Journal of Aircraft,* Vol. 36, No. 1, 1999, pp. 275-286.

[13]Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J. and Morris, M. D., "Screening, Predicting, and Computer Experiments," *Technometrics,* Vol. 34, No. 1, 1992, pp. 15-25.

[14]Box, G. E. P. and Draper, N. R., *Evolutionary Operation: A Statistical Method for Process Management*, John Wiley & Sons, Inc., New York, 1969.

[15]Balabanov, V. O., Giunta, A. A., Golovidov, O., Grossman, B., Mason, W. H. and Watson, L. T., "Reasonable Design Space Approach to Response Surface Approximation," *Journal of Aircraft,* Vol. 36, No. 1, 1999, pp. 308-315.

[16]Giunta, A. A., Balabanov, V., Kaufmann, M., Burgee, S., Grossman, B., Haftka, R. T., Mason, W. H. and Watson, L. T., "Variable-Complexity Response Surface Design of an HSCT Configuration," *Multidisciplinary Design Optimization: State of the Art - Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization (Alexandrov, N. M. and Hussaini, M. Y., eds.),* Hampton, VA, SIAM, March 13-16, 1996, pp. 348-367.

[17]Chen, W., Allen, J. K., Schrage, D. P. and Mistree, F., "Statistical Experimentation Methods for Achieving Affordable Concurrent Systems Design," *AIAA Journal,* Vol. 35, No. 5, 1997, pp. 893-900.

[18]Wujek, B. A. and Renaud, J. E., "New Adaptive Move-Limit Management Strategy for Approximate Optimization, Part 1," *AIAA Journal,* Vol. 36, No. 10, 1998, pp. 1911-1921.

[19]Wujek, B. A. and Renaud, J. E., "New Adaptive Move-Limit Management Strategy for Approximate Optimization, Part 2," *AIAA Journal,* Vol. 36, No. 10, 1998, pp. 1922-1934.

[20]Toropov, V., van Keulen, F., Markine, V. and de Doer, H., "Refinements in the Multi-Point Approximation Method to Reduce the Effects of Noisy Structural Responses," *6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization,* Bellevue, WA, AIAA, Vol. 2, September 4-6, 1996, pp. 941-951.

[21]Alexandrov, N., Dennis, J. E., Jr., Lewis, R. M. and Torczon, V., "A Trust Region Framework for Managing the Use of Approximation Models in Optimization," *Structural Optimization,* Vol. 15, No. 1, 1998, pp. 16-23.

[22]Alexandrov, N. M., "On Managing the Use of Surrogates in General Nonlinear Optimization and MDO," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization,* St. Louis, MO, AIAA, Vol. 2, September 2-4, 1998, pp. 720-729.

[23]Rodriguez, J. F., Renaud, J. E. and Watson, L. T., "Trust Region Augmented Lagrangian Methods for Sequential Response Surface Approximation and Optimization," *Advances in Design Automation (Dutta, D., ed.),* Sacramento, CA, ASME, September 14-17, 1997, Paper No. DETC97/DAC-3773.

[24]Renaud, J. E. and Gabriele, G. A., "Approximation in Nonhierarchic System Optimization," *AIAA Journal,* Vol. 32, No. 1, 1994, pp. 198-205.

[25]Renaud, J. E. and Gabrielle, G. A., "Sequential Global Approximation in Non-Hierarchic System Decomposition and Optimization," *Advances in Design Automation - Design Automation and Design Optimization (Gabriele, G., ed.),* Miami, FL, ASME, Vol. 32-1, September 22-25, 1991, pp. 191-200.

[26]Wujek, B. A., Renaud, J. E., Batill, S. M. and Brockman, J. B., "Concurrent Subspace Optimization Using Design Variable Sharing in a Distributed Computing Environment," *Advances in Design Automation (Azarm, S., et al., eds.),* Boston, MA, ASME, Vol. 82, September 17-21, 1995, pp. 181-188.

[27]Perez, V. M., Renaud, J. E. and Watson, L. T., "Adaptive Experimental Design for Construction of Response Surface Approximations," *42nd AIAA/ASME/ASC/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit,* Seattle, WA, AIAA, Vol. 5, April 16-19, 2001, pp. 3221-3232.

[28]Osio, I. G. and Amon, C. H., "An Engineering Design Methodology with Multistage Bayesian Surrogates and Optimal Sampling," *Research in Engineering Design,* Vol. 8, No. 4, 1996, pp. 189-206.

[29]Alexandrov, N. M., Nielsen, E. J., Lewis, R. M. and Anderson, W. K., "First-Order Model Management with Variable-Fidelity Physics Applied to Multi-Element Airfoil Optimization,"

8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, AIAA, September 6-8, 2000, AIAA-2000-4886.

[30]Booker, A. J., Dennis, J. E., Jr., Frank, P. D., Serafini, D. B., Torczon, V. and Trosset, M. W., "A Rigorous Framework for Optimization of Expensive Functions by Surrogates," *Structural Optimization,* Vol. 17, No. 1, 1999, pp. 1-13.

[31]Markus, S., Houstis, E. N., Catlin, A. C., Rice, J. R., Tsompanopoulou, P., Vavalis, E., Gottfried, D., Su, K. and Balakrishnan, G., "An Agent-based Netcentric Framework for Multidisciplinary Problem Solving Environments (MPSE)," *International Journal of Computational Engineering Science,* Vol. 1, No. 1, 2000, pp. 33-60.

[32]Serafini, D., "Software for Managing Approximation Models in Multidisciplinary Optimization," *6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization,* Bellevue, WA, AIAA, Vol. 2, September 4-6, 1996, pp. 1063-1067.

[33]Schonlau, M., Welch, W. J. and Jones, D. R., "Global Versus Local Search in Constrained Optimization of Computer Models," *New Developments and Applications in Experimental Design (Flournoy, N., Rosenberger, W. F., and Wong, W. K., eds.),* Institute of Mathematical Statistics, 1998, pp. 11-25.

[34]Wang, G., "Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points," *ASME Journal of Mechanical Design,* 2002, to appear.

[35]Wang, G., "Improvement on the Adaptive Response Surface Method for High-dimensional Computation-intensive Design Problems," *ASME Design Engineering Technical Conferences - Design Automation Conference (Diaz, A., ed.),* Pittsburgh, PA, ASME, September 9-12, 2001, Paper No. DETC2001/DAC-21141.

[36]Wang, G., Dong, Z. and Aitchison, P., "Adaptive Response Surface Method - A Global Optimization Scheme for Computation-intensive Design Problems," *Journal of Engineering Optimization,* Vol. 33, No. 6, 2001, pp. 707-734.

[37]Trosset, M. W. and Torczon, V., "Numerical Optimization Using Computer Experiments," *NASA CR-201724, ICASE Report No. 97-38,* Institute for Computer Applications in Science and Engineering (ICASE) - NASA Langley Research Center, Hampton, VA, 1997.

[38]Koehler, J. R. and Owen, A. B., "Computer Experiments," *Handbook of Statistics (Ghosh, S. and Rao, C. R., eds.),* Elsevier Science, New York, 1996, pp. 261-308.

[39]Welch, W. J., Yu, T.-K., Kang, S. M. and Sacks, J., "Computer Experiments for Quality Control by Parameter Design," *Journal of Quality Technology,* Vol. 22, No. 1, 1990, pp. 15-22.

[40]Booker, A. J., Conn, A. R., Dennis, J. E., Frank, P. D., Trosset, M. and Torczon, V., "Global Modeling for Optimization: Boeing/IBM/Rice Collaborative Project," *1995 Final Report, ISSTECH-95-032,* The Boeing Company, Seattle, WA, 1995.

[41]Sacks, J., Schiller, S. B. and Welch, W. J., "Designs for Computer Experiments," *Technometrics,* Vol. 31, No. 1 (February), 1989, pp. 41-47.

[42]Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science,* Vol. 4, No. 4, 1989, pp. 409-435.

[43]Mathworks, *Matlab 6.0 User Guide: Fuzzy Logic Toolbox*, 2001.

[44]Bezdek, J. C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.s

[45]Milligan, G. W. and Copper, M. C., "Methodology Review: Clustering Methods," *Applied Psychological Measurement,* Vol. 11, No. 4, 1987, pp. 329-354.

[46]Jin, R., Chen, W. and Simpson, T. W., "Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria," *Journal of Structural and Multidisciplinary Optimization,* Vol. 23, No. 1, 2001, pp. 1-13.

[47]Wilson, B., Cappelleri, D. J., Frecker, M. I. and Simpson, T. W., "Efficient Pareto Frontier Exploration Using Surrogate Approximations," *Optimization and Engineering,* Vol. 2, No. 1, 2001, pp. 31-50.

[48]Iman, R. L. and Conover, W. J., "Small Sample Sensitivity Analysis Techniques for Computer Models, with an Application to Risk Assessment," *Communication in Statistics - Theory and Methodology, Part A,* Vol. 9, No. 17, 1980, pp. 1749-1842.

[49]Simpson, T. W., Lin, D. K. J. and Chen, W., "Sampling Strategies for Computer Experiments: Design and Analysis," *International Journal of Reliability and Applications,* Vol. 2, No. 3, 2001, pp. 209-240.

[50]Chiu, S. L., "Fuzzy Model Identification Based on Cluster Estimation," *Journal of Intelligent and Fuzzy Systems,* Vol. 2, No. 2, 1994, pp. 267-278.

[51]Yager, R. R. and Filev, D. P., "Generation of Fuzzy Rules by Mountain Clustering," *Journal of Intelligent and Fuzzy Systems,* Vol. 2, No. 2, 1994, pp. 202-219.

[52]Milligan, G. W. and Copper, M. C., "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika,* Vol. 50, No. 2, 1985, pp. 159-179.

[53]Simpson, T. W., Mauery, T. M., Korte, J. J. and Mistree, F., "Kriging Metamodels for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," *AIAA Journal,* Vol. 39, No. 12, 2001, pp. 2233-2241.

[54]Branin, F. H. and Hoo, S. K., "A Method for Finding Multiple Extrema of a Function of n Variables," *Numerical Methods for Non-Linear Optimization (Lootsma, F., ed.),* Academic Press, New York, 1972, pp. 231-237.

[55]Messac, A., "Physical Programming: Effective Optimization for Computational Design," *AIAA Journal,* Vol. 34, No. 1, 1996, pp. 149-158.

List of Tables

Table 1 Summary of the clusters' properties for SC.

	Cluster 1	**Cluster 2**
Center	(0.4886 0.4945)	(0.5182 0.4949)
Range	x1 [-2 2] x2 [-1.2 1.2]	x1[-2 2] x2[-1.6 1.6]
Minimum	-5.0819	5.5842
Mean	0.2173	10.940
Median	-0.0133	11.3745
STD	3.1473	3.4183

Table 2 Optimization iterations in the final space for the SC problem.

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**
# of samples for modeling	12	19	27	38
# of new samples	7	7	8	11
Total # of func. evaluations	**19**	**26**	**34**	**45**
Model optimum	-1.2336	-1.2324	-1.0079	-1.0240
Real func. value at the model optimum	**-0.5868**	**-0.9424**	**-1.0242**	**-1.0303**
Optimum at	(0.0632 0.443)	(0.1367 -0.6089)	(0.0563 -0.7297)	(-0.0774 0.7211)

Table 3 Material properties of beam layers.

Material Type	**ρ (Kg/m^3)**	**E (N/m^2)**	**C ($/m^3)**
1	100	$1.6 * 10^9$	500
2	2770	$70 * 10^9$	1500
3	7780	$200 * 10^9$	800

Table 4 Comparison of the optimization results for three strategies.

	SC			OC			SB		
	K-All	**KCK**	**PCK**	**K-All**	**KCK**	**PCK**	**K-All**	**KCK**	**PCK**
Optimal Func. Value	-1.031	-1.030	-1.030	0.007	0	0	358.161	355.705	330.96
Optimal Point	(-0.086 0.717)	(-0.077 0.721)	(-0.077 0.721)	0.997	1	1	(0.180 0.010 0.013 0.500 3.110)	(0.183 0.011 0.014 0.500 3.002)	(0.168 0.010 0.011 0.512 3.008)
# of Func. Eval.	78	45	45	31	16	15	148	180	97
Convergence at Iteration #	6	5	5	5	4	3	4	4	3

Table 5 Comparison of KCK and PCK on space reduction.

	SC		OC		SB	
	KCK	**PCK**	**KCK**	**PCK**	**KCK**	**PCK**
# of model points for clustering	121	121	201	201	600	600
Reduced Space	x_1:[-2 2] x_2:[-1.2 1.2]	x_1:[-2 2] x_2:[-1.2 1.2]	[0.485 1]	[0.540 1]	x_1:[0.018 0.942] x_2[0 0.047] x_3[0 0.042] x_4[0.506 1.034] x_5[3.017 9.803]	x_1:[0.074 0.976] x_2[0 0.048] x_3[0 0.043] x_4[0.511 1.996] x_5[3.008 5.944]
[fMin fMax]	[-3.6 35.1]	[-5.1 39]	[0.01 65]	[0.02 4.4]	[516 2e10]	[197 2e10]
Range of Cutting Values	[18 35]	[14 25]	[0.6 1.65]	[3.5 4.4]	[5e3 2e4]	[4e3 1e6]

List of Figures

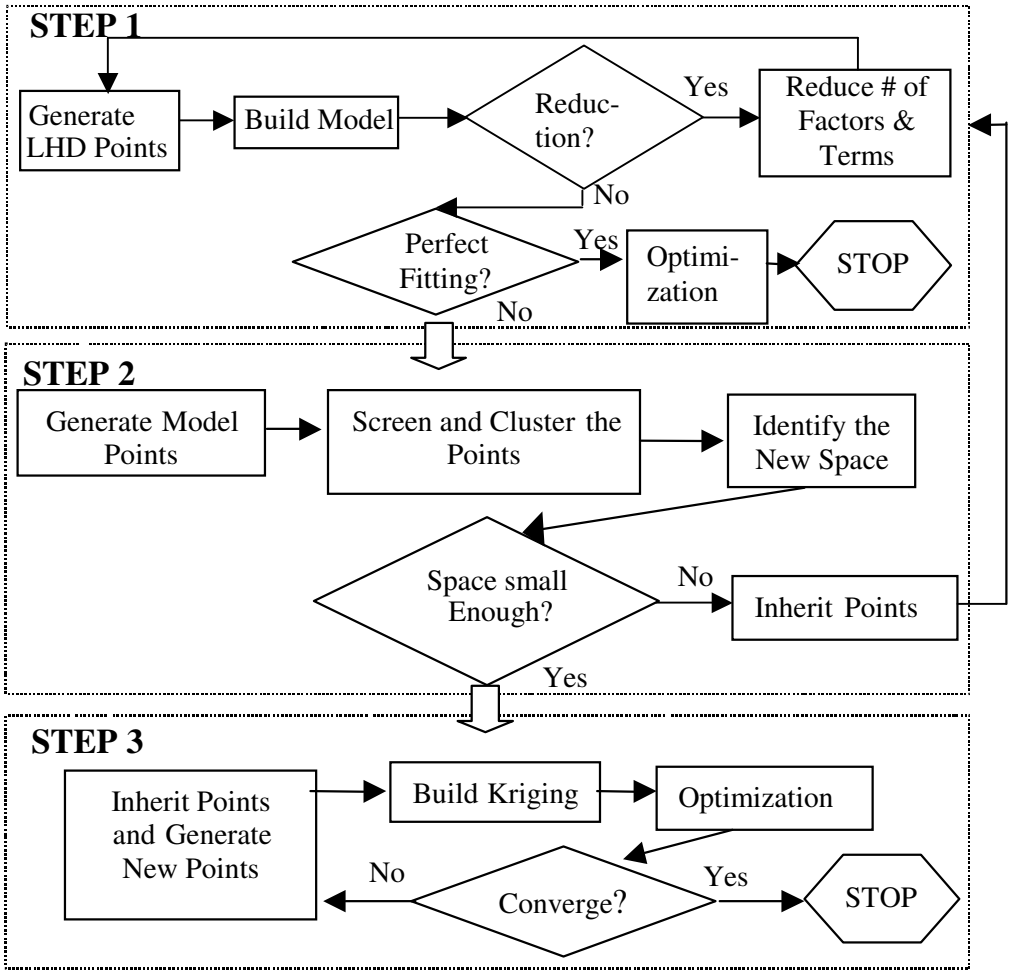


Figure 1 The flowchart of the proposed method.

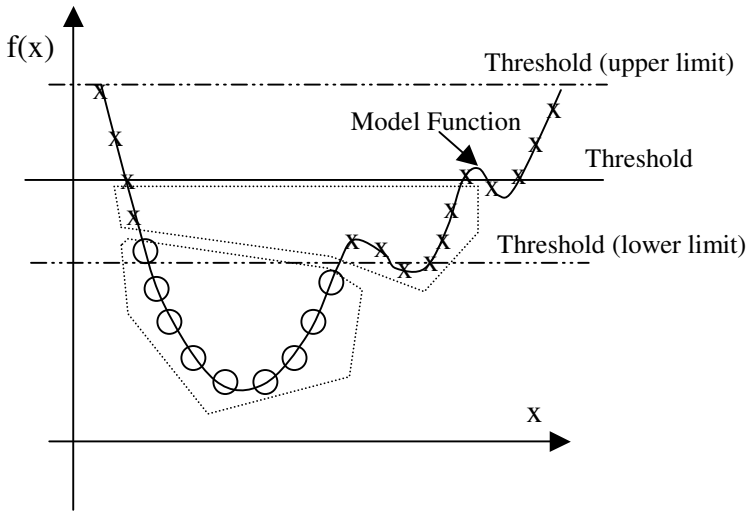


Figure 2 An illustration of point screening and fuzzy clustering.

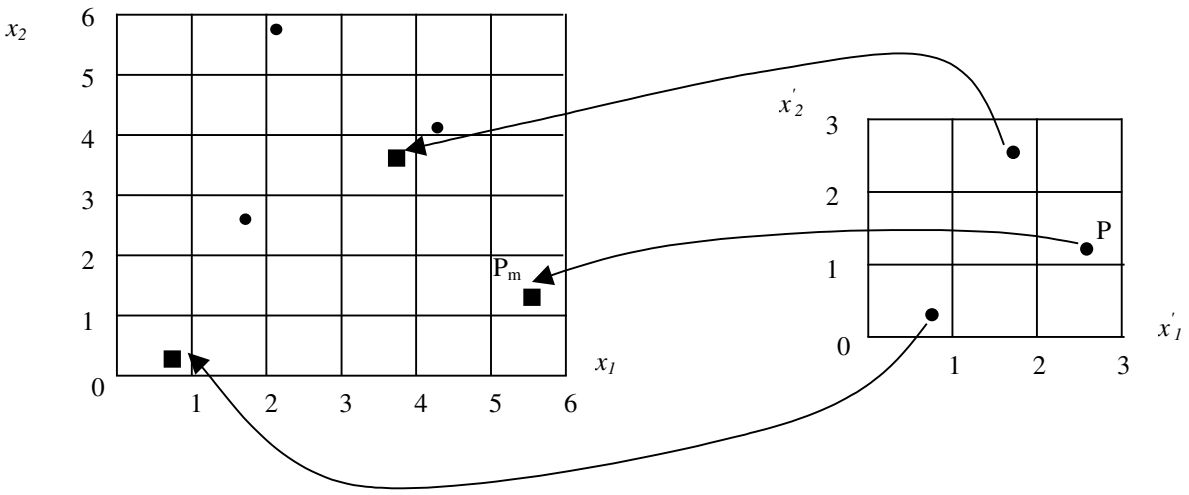
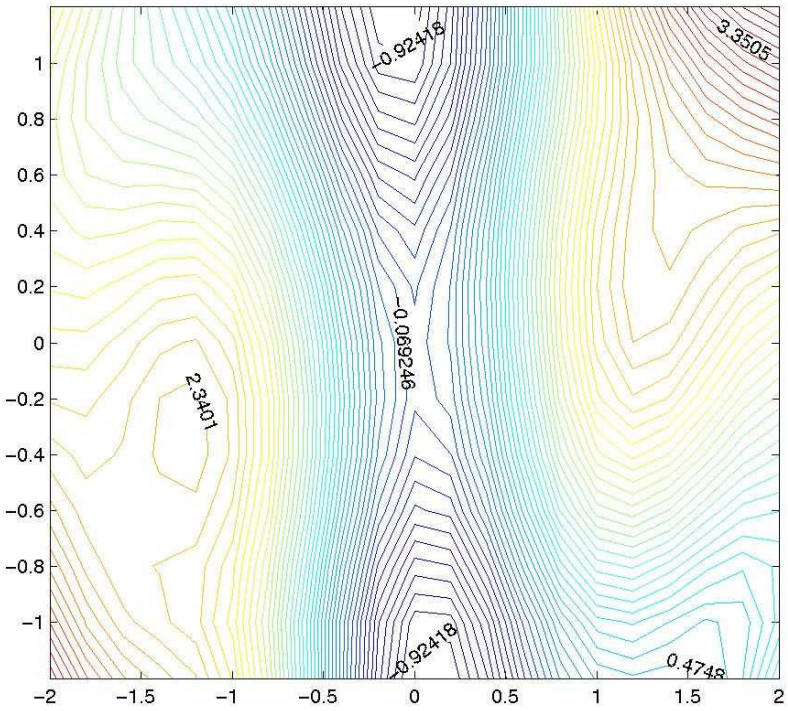
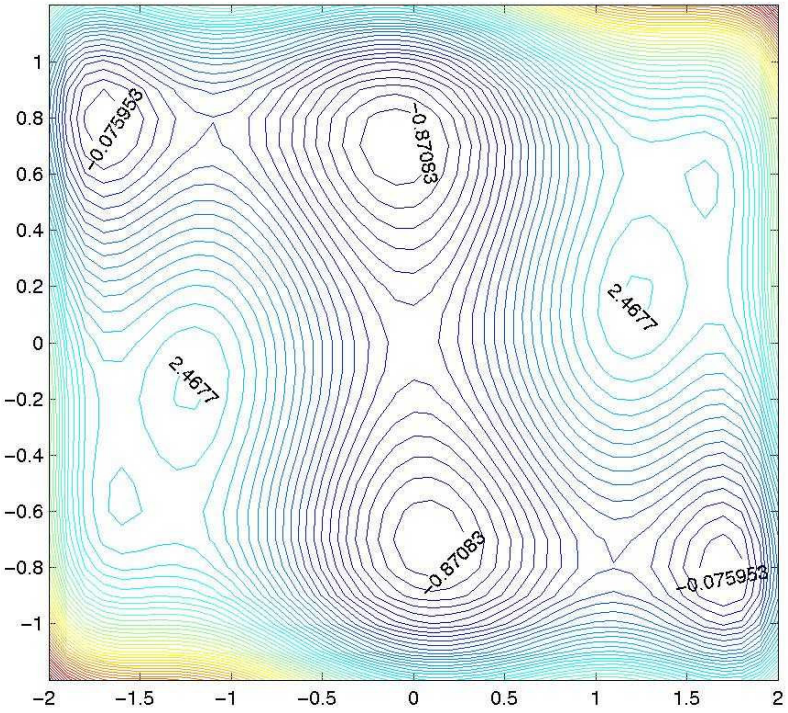


Figure 3 A LHD sample set formed by inherited and new points.



a) Contour plot of the obtained kriging model at iteration #4



b) Contour plot of the SC function

Figure 4 Comparison of the fitted kriging model with the original SC function.

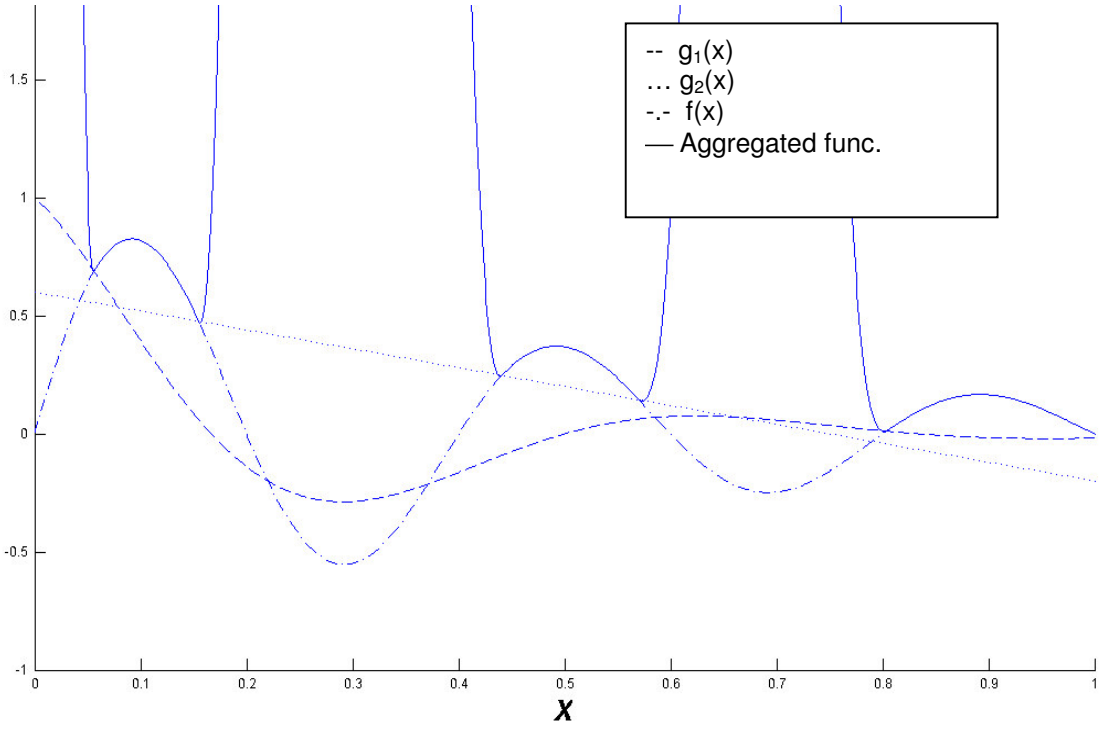


Figure 5 Plot of the constrained optimization functions.

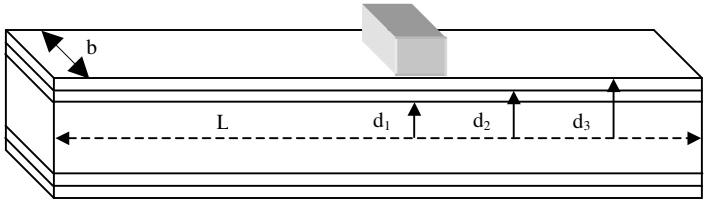


Figure 6 The sandwich beam design problem.