10^{th} World Congress on Structural and Multidisciplinary Optimization
May 19 - 24, 2013, Orlando, Florida, USA

Classification-based Quantification of Constraint Violation for Efficient Global Optimization and Failure Probability Confidence Bound Calculation

Anirban Basudhar

Livermore Software Technology Corporation, Livermore, CA 94551

Samy Missoum

The University of Arizona, Tucson, AZ 85721

1. Abstract
This paper presents a model to provide probabilistic outputs for a support vector machine classifier. Given a set of evaluated samples and a trained SVM boundary separating the samples into two classes, the model provides the probability of belonging to a certain class at any point. Unlike prior probabilistic SVM models, the proposed model considers the correlation to evaluated samples for predicting the probability. The proposed model is compared to prior models using an error measure, and is used to calculate the probability of constraint violation. This probability is used in the handling of constraints in efficient global optimization; the objective function is approximated using Kriging while the constraints are defined using an SVM and a PSVM that provides probability outputs for the trained SVM. Also, the probability of constraint violation is used to define confidence bounds of a desired level on a probability of failure. The proposed constraint handling method is classification-based and does not involve response approximation; this approach alleviates issues due to discontinuous or binary responses. Multiple constraints (uncorrelated or correlated) can be defined using a single SVM. Several numerical examples and a crash problem are presented to demonstrate the method.
2. Keywords: probabilistic support vector machines, efficient global optimization, failure probability confidence interval, binary and discontinuous responses, multiple failure modes and constraints

3. Introduction
With increasing complexity of engineering designs, their evaluation, whether using a physical test or a computational code, is becoming increasing expensive. As a result the cost of optimization and reliability assessment, which require repeated function evaluations and are important steps in engineering design, has also increased. Therefore, it is important to limit the number of samples (design instances) required to locate an optimal design or to assess the reliability of a design. Much importance has been received by metamodel-based methods in the literature [1–7] to achieve that goal.
Metamodels are used to replace the actual expensive model by building an approximation of the response values. Only a few samples evaluated through the actual model can be used to construct the metamodel, which can then be used to predict the response value at any sample. If the metamodel is accurate, several standard optimization algorithms or reliability assessment methods are available that can provide accurate results by repeatedly evaluating the metamodel at various design configurations. However, the more difficult part is to construct an accurate metamodel using a few samples. Quality of the samples has an important role in the accuracy, and it is therefore important to select them strategically. Adaptive sampling techniques have been developed for that purpose [3, 8]. One such technique for optimization is efficient global optimization (EGO) [8], which is an attractive method for accurately and efficiently performing global optimization. For reliability assessment also adaptive sampling methods have been developed [9–11]. Metamodel-based methods are however hampered in certain cases. Some of the potential issues are as follows.

- presence of discontinuous responses hampers the accuracy
- presence of multiple constraints or failure modes makes the process more complex and expensive
- Presence of binary or hidden constraints. Failure region may be defined based on the inability to obtain a response. Thus, no approximation is possible in the failure or infeasible region.

To address the above issues, classification has been used instead of approximation to construct a boundary separating the feasible (or safe) and infeasible (or failure) regions. The boundaries are constructed using methods such as convex hulls [12], support vector machines (SVM) [13] etc. Instead of predicting the constraint response values over the entire design space, the goal is to simply predict whether or not the response is greater than a threshold. This subtle difference has the potential to alleviate several issues faced during response approximation using metamodels, as it only requires the class of a sample and not the actual constraint function values. For example, in problems with hidden constraints, the failed samples are assigned to the failure class even though the actual constraint function values are unavailable [14, 15].
Support vector machines [16, 17]have become popular tools for classification of data, and are used in this article. However, one of the limitations is that they do not provide probabilistic confidence measures for the class predictions. This has implications for both optimization and reliability assessment:

- For constrained EGO, constraint violation probability is required in most formulations. Not having probabilistic outputs restricts the use of such sampling strategies.
- The probability of failure calculated based on the classification boundary may not be accurate if training samples are limited. It is desirable to have confidence bounds on the probability.

The above issues can be addressed if probabilistic outputs can be obtained using the trained SVM classifiers. This is achieved using probabilistic support vector machines (PSVMs) [18, 19], which provide the probability of belonging to a certain class at any point. However, the existing PSVM models do have certain limitations due to the way in which proximity to existing samples is considered while predicting the probabilities. One of the goals of this article is to address this issue by proposing a new model. The effect of an evaluated sample at any point is assessed based on the value of correlation or kernel function. The probabilities are then calculated based on the cumulative effect of all samples.

The proposed PSVM model is used to quantify the probability of constraint violation, and is implemented within a constrained EGO formulation to select samples for locating the optimal solution [15]. The objective function values are predicted using Kriging and the constraints are handled by a single PSVM (for multiple constraints also). In another aspect of this article, the proposed PSVM model is used to provide confidence bounds on the probability of failure. If the available sampling data is limited, the proposed method can be used to provide a conservative probability of failure with desired level of confidence [19].

The organization of this article is as follows. Section 4 presents the background of different methods used in the article. In section 5, the proposed correlation based PSVM model is presented. Section 6 presents the proposed methodology for EGO using PSVM constraints. In Section 7, the calculation of confidence bounds for failure probability is explained. An error measure to quantify the accuracy of PSVM models is presented in Section 8. Finally, example problems are presented in Section 9. A comparison of the PSVM models is provided in Section 9.1. In Section 9.2, the proposed PSVM model is applied to provide confidence intervals on the probability of failure of a car crash example. In Section 9.3, examples of constrained EGO using the proposed method are provided.

4. Background

4.1. Efficient Global Optimization

Efficient global optimization (EGO) is an attractive tool for optimization that balances the search for sparse region in the space and regions that are already known to yield a good solution (low objective function). It was originally proposed for unconstrained optimization, and shown to work very effectively to locate global optima. In more traditional metamodel-based optimization, an approximation of the objective function is created and it is then used to predict the objective function values at other points and to locate the optimum. In EGO, the basic idea is to not only consider the predicted values of the response functions, but also their predicted variance at a point. A balance is then obtained, using an expected improvement (EI), between the need to exploit the regions of space with good prediction values and to explore sparse regions with lack of data that may have inaccurate current metamodel predictions, and can potentially provide better solutions. The metamodel used generally for EGO is Kriging, more details of which can be found in [20–23]. Kriging provides a measure of the prediction mean as well as the variance of a response function $w_{act}(\mathbf{x})$. At any point $\mathbf{x}$, the mean prediction $\mu_w(\mathbf{x})$ and variance $\sigma_w^2(\mathbf{x})$ of the response are:

$$\mu_w(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T\boldsymbol{\beta} + \mathbf{r}(\mathbf{x})^T\mathbf{R}^{-1}(\mathbf{w} - \mathbf{F}\boldsymbol{\beta}) \tag{1}$$

$$\sigma_w^2(\mathbf{x}) = \sigma_Z^2 - \begin{bmatrix} \mathbf{h}(\mathbf{x})^T & \mathbf{r}(\mathbf{x})^T \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{R} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{h}(\mathbf{x}) \\ \mathbf{r}(\mathbf{x}) \end{bmatrix} \tag{2}$$

where σ_Z^2 is the variance of the Gaussian process Z used for the approximation, $\mathbf{r}(\mathbf{x})$ is the vector of covariance between $\mathbf{x}$ and all the training samples, $\mathbf{R}$ is a matrix consisting of the pairwise covariances between all the samples, $\mathbf{w}$ is the vector of response values at the N samples, and $\mathbf{F}$ is a matrix with its i^{th} row given by the trend $\mathbf{h}(\mathbf{x}_i)^T$ (e.g, a linear trend as used in this work) calculated at the i^{th} sample and $\boldsymbol{\beta}$ is the vector of trend coefficients. A widely used form for the correlation function R [24] used to give the correlation between any two points is the Gaussian correlation function:

$$R(\mathbf{a}, \mathbf{b}) = e^{-\sum_{i=1}^{m} \theta_i |a_i - b_i|^2} \tag{3}$$

where m is the number of dimensions and θ_i, determined based on maximum likelihood [8, 22], is the scaling parameter for the i^{th} dimension.
The expected improvement (EI) used in EGO is the expected value by which the predicted objective function value is lower than the current minimum:

$$EI(\mathbf{x}) = E\left[I(\mathbf{x})\right] = E\left[\max\left(0, w^\star - w(\mathbf{x})\right)\right] = \int_{-\infty}^{w^\star} (w^\star - w) f_{\hat{w}} dw \tag{4}$$

where $w^\star$ is the current minimum and w is a realization of $\hat{w}$, $f_{\hat{w}}$ is the normal probability density function of the Kriging model $\hat{w}$ at point $\mathbf{x}$. Figure 1 provides an example of "probability of improvement". For a Gaussian process model, EI can be expressed analytically [23]:

$$EI(\mathbf{x}) = (w^\star - \mu_w(\mathbf{x}))\Phi\left(\frac{w^\star - \mu_w(\mathbf{x})}{\sigma_w(\mathbf{x})}\right) + \sigma_w(\mathbf{x})\phi\left(\frac{w^\star - \mu_w(\mathbf{x})}{\sigma_w(\mathbf{x})}\right) \tag{5}$$

where ϕ and Φ are the standard normal probability density function and cumulative density function respectively.
The maximization of EI balances exploration of sparse regions and exploitation of regions with low objective function values, and presents an efficient scheme to select samples for unconstrained global optimization. The point with the maximum expected improvement is evaluated sequentially to update the Kriging model.
Several variations and improvements of the EGO sampling criterion can be found in the literature [25–31]. These works introduce notions such as generalized expected improvement [27], parallel evaluation of samples [30, 31], or use of EGO with other metamodels than Kriging [32]. The handling of constrained

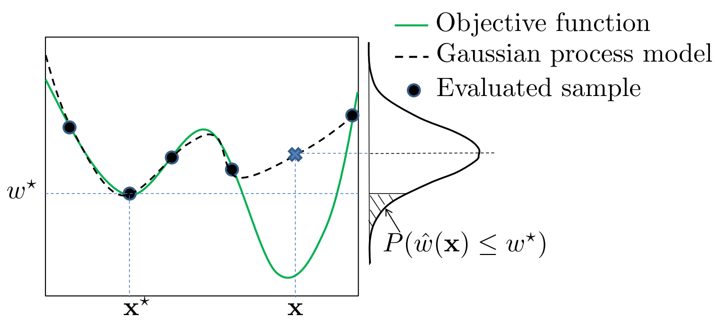

Figure 1: Depiction of the probability of improving the current minimum $\mathbf{x}^\star$.

optimization problems with EGO is more involved, as it has the added complexity of approximating the constraints (which can be multiple), and is an area which has scope for improvement. In order to include constraints in the optimization, several EGO formulations have been proposed, such as the multiplication of EI with the probability of feasibility calculated using a Kriging approximation for each constraint [25], the penalty method [26], the maximization of EI with samples constrained to lie in the feasible space defined based on the mean values of the Kriging models for the constraints, and the use of the "expected violation" [33].

One of the issues that impacts constrained EGO methods adversely arises when multiple constraints are present. When the constraints are correlated, the computation of the probability of feasibility requires information about the correlation that is often not available. In many practical applications, this is a problem as the constraints (e.g., responses of a system) might indeed be correlated. Another issue is the presence of discontinuous or binary constraint functions. In some cases the constraint function returns values only for the feasible space. Also, sometimes only pass or fail information is available instead of a constraint function value. These issues can resolved if constraint boundaries are handled as binary classifiers. Recently, indicator Kriging, random forest classifiers and PSVM have been used to address these issues [14, 15, 34, 35]. In this article, these issues are addressed using a PSVM based approach. A modified PSVM model is presented in Section 5 for this purpose. Section 6 presents the sampling strategy for PSVM-based EGO.

4.2. Probability of failure

Probability of failure calculation, like optimization, requires repeated function evaluations. It is given by the integral of the joint probability density function of the random variables over the failure domain:

$$P_f = P(g(\mathbf{x}) \leq 0) = \int_{\Omega_f} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \tag{6}$$

Calculation of the above integral is not straightforward as the failure domain can be irregular and is not known explicitly in general. One of the most basic methods to calculate failure probabilities is Monte Carlo Simulations (MCS), in which a large number of samples based on the probability density functions of the variables are generated. The fraction of Monte Carlo samples in the failure region gives the probability of failure. The accuracy of the probability of failure depends on the level of the true probability and the number of Monte Carlo samples. The confidence interval of failure probability is given as :

$$\text{CI} = \left[\left(P_f - \kappa P_f \sqrt{\frac{1 - P_f^t}{N_{MC} P_f^t}} \right), \left(P_f + \kappa P_f \sqrt{\frac{1 - P_f^t}{N_{MC} P_f^t}} \right) \right] \tag{7}$$

where κ is a factor obtained from the standard Normal distribution table. $\kappa = 1.96$ for 95% confidence. In general N_{mc} needs to be a large number, and it is not practical to evaluate all the Monte Carlo samples using the actual function evaluator. Therefore, the failure boundary is often approximated based on fewer samples, and the class of Monte Carlo samples is determined based on the approximation. In this article, the boundaries are approximated using SVM presented in Section 4.3. A method to calculate confidence bounds of the probability of failure is presented in Section 7.

4.3. Support Vector Machines

Support vector machines are statistical learning tools that can be used for classification or regression. In the context of defining a boundary (e.g. constraint or failure boundary), one can simply use classification without approximating the response values over the entire space. An SVM classification boundary separates the design space into two regions labeled as +1 and -1 (e.g. failure and safe region) . The boundary is given as:

$$s(\mathbf{x}) = b + \sum_{i=1}^{N} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) = 0, \tag{8}$$

where N is the number of training samples with known class, $\mathbf{x}_i$ is the i^{th} training sample, λ_i is the corresponding Lagrange multiplier, y_i is the class that can take values +1 or -1, K is a kernel function, and b is the bias. λ_i and b are obtained from the the solution of a nonlinear SVM training optimization. The class of any point in the space is determined by the sign of $s(\mathbf{x})$. All training samples have $|s(\mathbf{x})| \geq 0$, unless no such solution exists and a relaxed optimization is solved.

One of the commonly used kernel functions Gaussian radial basis function is used in this work:

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}_i||^2}{2\sigma^2} \right) \tag{9}$$

For more details on SVMs, the reader is referred to [16, 17, 36, 37].

4.4. Probabilistic support vector machines

The $s(\mathbf{x}) = 0$ isocontour represents the SVM classification boundary separating the =1 and -1 classes. Also, $|s(\mathbf{x})| < 0$ regions are usually empty, i.e. do not have any training samples. Apart from that, the value of SVM does not provide much intuitive interpretation. PSVMs, unlike SVMs, provide probability outputs between 0 and 1 that represent the probability of belonging to a certain class at any point in the design space. The most common PSVM model is based on the sigmoid function using SVM values [16,18]. For a given sample $\mathbf{x}$, the probability of belonging to the +1 class is:

$$P(+1|\mathbf{x}) = \frac{1}{1 + e^{As(\mathbf{x})+B}} \tag{10}$$

The parameters A ($A < 0$) and B of the sigmoid function are found such that the likelihood is maximized [18,38].

The model in Eq. (10) has some obvious limitations that were addressed in [15,19] using a distance based PSVM.

$$P(+1|\mathbf{x}) = \frac{1}{1 + e^{As(\mathbf{x})+B\left(\frac{d_-(\mathbf{x})}{d_+(\mathbf{x})+\delta} - \frac{d_+(\mathbf{x})}{d_-(\mathbf{x})+\delta}\right)}} \qquad A < \frac{-3}{\min(s_{max}, -s_{min})}, B < 0 \tag{11}$$

where $d_-(\mathbf{x})$ and $d_+(\mathbf{x})$ are the distances to the closest -1 and $+1$ samples. δ is a small quantity (set to 10^{-10} in this work) added in order to avoid numerical issues at the evaluated training samples. s_{max} and s_{min} are the maximum and minimum values of the SVM calculated at the training samples. The distance-based PSVM model is therefore dependent on the SVM values, but it does not disregard the spatial distribution of the samples. Unlike Eq. (10) it does not predict a non-zero $P(+1|\mathbf{x})$ for training samples belonging to the -1 class.

In this work, an improvement over the distance-based PSVM model is proposed in Section 5, which is then used for two purposes - to calculate the probability of feasibility for use in a constrained EGO solution, and to provide confidence bounds on probabilities of failure.

5. Correlation-based probabilistic support vector machine

The distance-based PSVM in Eq. (11) provides a way to consider both the SVM values and spatial positions of training samples. However, one of the limitations lies in the way in which the distances are considered in the model; it only considers the ratio of the distances to the closest +1 and -1 samples from the point under consideration. It does not take into account the cumulative effects of all the samples in the proximity. As a result, the probability of misclassification may not vanish even in heavily sampled region belonging to a single class. To overcome this issue and to provide more realistic probability outputs, a modified PSVM is presented in this section that uses a correlation function, the same as SVM kernel, to measure the effect of a sample at any point in the design space. The correlation, given by the Gaussian radial basis function, is highest at the basis center or the sample and reduces as the distance increases. The cumulative effect of all the samples is considered in the PSVM model:

$$P(+1|\mathbf{x}) = \frac{1}{1 + e^{As(\mathbf{x})+B\rho(\mathbf{x})}} \qquad A < \frac{-3}{\min(s_{max}, -s_{min})}, B < 10^{-3}$$

$$\rho(\mathbf{x}) = \sum_{i=1}^{N} y_i \tan\left(\frac{\pi}{2} K(\mathbf{x}, \mathbf{x}_i)\right) \tag{12}$$

At any +1 sample, $P(+1|\mathbf{x}) = 1$ and at any -1 sample, $P(+1|\mathbf{x}) = 0$.

$$\begin{gathered} \mathbf{x} \to \mathbf{x}_i \Rightarrow K(\mathbf{x}, \mathbf{x}_i) \to 1 \Rightarrow \rho(\mathbf{x}) \to y_i\infty \\ P(+1|\mathbf{x}) = \frac{1}{1 + \exp As(\mathbf{x}) + B(y_i\infty)} = \frac{1}{1 + \exp -(y_i\infty)} \\ y_i = 1 \Rightarrow P(+1|\mathbf{x}) = 1 \qquad y_i = -1 \Rightarrow P(+1|\mathbf{x}) = 0 \end{gathered} \tag{13}$$

6. EGO using PSVM constraints

In this section the methodology for EGO using PSVM-based constraints is presented. Because the method is based on classification, it overcomes the issues of discontinuous or binary responses. Multiple constraints can be treated with a single PSVM.
The basic methodology is similar to [15]; except for modifications in the sampling criteria and the PSVM model. One sample is added based on the objective function improvement function in each iteration. In addition, samples having high probability of misclassifiction by the SVM are added locally in sparse regions around these ones. All the samples selected during an iteration are evaluated in parallel. Overall four types of samples are used, three of them being based on the objective function improvement function and evaluated alternatively:

1. Constrained EI maximization:

$$\begin{aligned} \max_{\mathbf{x}} \quad & EI(\mathbf{x}) \\ s.t. \quad & P(+1|\mathbf{x}) \geq \delta_{pp} \end{aligned} \tag{14}$$

   This criterion is used at alternate iterations along with Eq. (15). The first and third samples in a cycle of five iteration are selected using this criterion. The initial value of δ_{pp} is 0.5. It is relaxed by 0.05 is no solution is found, until a limit of probability of feasibility equal to 0.2.

2. Maximization of the product of EI and $P(+1|\mathbf{x})$:

$$\max_{\mathbf{x}} \quad EI(\mathbf{x})P(+1|\mathbf{x}) \tag{15}$$

   The second and fourth samples in a cycle of five iteration are selected using this criterion. This criterion is not constrained by the probability of feasibility, and can explore regions in the infeasible space with high EI, potentially leading to a faster location of the optimal solution. On the other hand, this may lead to too much sampling away from the SVM boundary and not improve the solution - if the EI term dominates, the infeasible region may be heavily sampled and if the probability term dominates then the feasible region may be heavily sampled away from the boundary. Therefore, this criterion should be used along with other criteria, such as Eq. (14) and Eq. (16).

3. Maximization of the product of probability of improvement and $P(+1|\mathbf{x})$:

$$\max_{\mathbf{x}} \quad P(I(\mathbf{x}) > 0)P(+1|\mathbf{x}) \tag{16}$$

   Every fifth sample in a cycle of five is selected using this criterion. This sampling criterion has a more intuitive interpretation compared to Eq. (16). It represents the mximization of probabilities of improving the objective function and locating a feasible sample at the same time (considering the two are independent).

4. Auxiliary sample for local SVM boundary refinement - maximum minim distance with misclassification probability greater than 0.5: These samples are added to investigate the local refinement of the SVM constraint boundary approximation around the current best predicted solution. After a sample is found using one of the above criteria, n_p additional samples are added in its vicinity in a hyperspherical update region. In this work, n_p is equal to the problem dimensionality m. At iteration k, the radius R_u^k of the update region is selected such that it consists of at least one sample from either class:

$$R_u^k = \max(d_+, d_-) \tag{17}$$

   where d_+ and d_- are the distances to the closest +1 and −1 samples from the center of the hypersphere.

   The optimization problem to locate an auxiliary sample is:

$$\begin{aligned} \max_{\mathbf{x}} \quad & d(\mathbf{x}) \\ s.t. \quad & P_m(\mathbf{x}) \geq 0.5 \\ & ||\mathbf{x} - \mathbf{x}_c|| - R_u^k \leq 0 \end{aligned} \tag{18}$$

where d is the distance to the closest training sample and $\mathbf{x}_c$ is the center of the hypersphere. P_m is the probability of misclassification defined as:

$$P_m(\mathbf{x}) = \begin{cases} 1 - P(+1|\mathbf{x}) & s(\mathbf{x}) \geq 0 \\ P(+1|\mathbf{x}) & s(\mathbf{x}) < 0 \end{cases} \tag{19}$$

The value of 0.5 used for the probability of misclassification corresponds to samples on or close to the SVM.

7. PSVM-based confidence intervals for failure probability

The confidence interval of MCS in Eq. (7) is calculated based on the assumption that the N_{mc} Monte Carlo samples have been evaluated using the actual function (e.g. finite element analysis, physical test etc.), and their class (safe or failed) is known without any uncertainty. However, when using approximations, the accuracy of the failure boundary depends on the quality of the samples and the surrogate model. As a result, the boundary approximation stage introduces additional uncertainty, and Eq. (7) is no longer valid. In this section, a modified estimate of the probability of failure bounds is presented in the context of support vector machine boundaries. The bounds can be calculated for any desired level of confidence. This is made possible through the use of PSVM. To find the $\alpha_u\%$ upper confidence bound of the failure probability a probability of misclassification is considered for any point for which $s(\mathbf{x}) > 0$ and $P(+1|\mathbf{x}) < \frac{\alpha_u}{100}$:

$$P_{ub} = \frac{N(s(\mathbf{x}) \leq 0)}{N_{mc}} + \sum_{i=1}^{N_{mc}} \psi(\mathbf{x})$$
$$\psi(\mathbf{x}) = \begin{cases} 1 - P(+1|\mathbf{x}) & P(+1|\mathbf{x}) < \frac{\alpha_u}{100} \cap s(\mathbf{x}) > 0 \\ 0 & otherwise \end{cases} \tag{20}$$

To find the $\alpha_l\%$ lower confidence bound of the failure probability, a probability of misclassification is considered for any point for which $s(\mathbf{x}) < 0$ and $P(+1|\mathbf{x}) > 1 - \frac{\alpha_u}{100}$.

8. Error measure of PSVM models

An error measure to compare PSVM models is presented in this section. In the case where the actual limit-state function is known, P(+1—x) is known for any point and is equal to 0 or 1. Therefore the error of the PSVM model can be calculated at any point. A large number of uniform test samples are used for this purpose. The probability of misclassification for the i^{th} test sample is:

$$P_{misc}(\mathbf{x}_i) = \begin{cases} 1 - P(+1|\mathbf{x}_i) & y_i = +1 \\ P(+1|\mathbf{x}_i) & y_i = -1 \end{cases}, \tag{21}$$

where $\mathbf{x}_i$ represents the i^{th} test point with class label y_i. A low probability of misclassification for the test points is an indicator of a good PSVM model. The error E_{test} is defined as the mean probability of misclassification for all the test points [15]:

$$E_{test} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} P_{misc}(\mathbf{x}_i) \tag{22}$$

9. Examples

In this section, numerical examples are presented to demonstrate the efficacy of the proposed PSVM model and associated optimization and failure probability calculation methods. In section 9.1, the proposed PSVM model is compared to its predecessor. Section 9.2 presents a crash example for which the PSVM is used to provide lower and upper confidence bounds on the probability of failure. In Section 9.3, two numerical examples are presented to demonstrate the efficacy of the PSVM-based EGO method.

9.1. Examples to compare PSVM models

In this section the proposed correlation-based PSVM is compared to the closest distance-based PSVM. Two numerical examples are presented. The errors are quantified using Eq. (21). The errors are plotted for varying design of experiment (DOE) sizes. Optimized Latin hypercube sampling is used to select the DOE.

9.1.1. Example 1. Three constraint two island problem

The constraints for this example are defined for an optimization problem in [27]. The problem consists of two variables x_1 and x_2 with identical ranges $[0, 1]$. The feasible space is an intersection of the three following regions:

$$\begin{aligned} g_1(\mathbf{x}) &= ((x_1-3)^2+(x_2+2)^2)e^{(-x_2^7)}-12 \leq 0 \\ g_2(\mathbf{x}) &= 10x_1+x_2-7 \leq 0 \\ g_3(\mathbf{x}) &= (x_1-0.5)^2+(x_2-0.5)^2-0.2 \leq 0 \end{aligned} \tag{23}$$

The constraint boundaries are plotted in Figure 2. The values of E_{misc} are plotted in Figure 3 for the

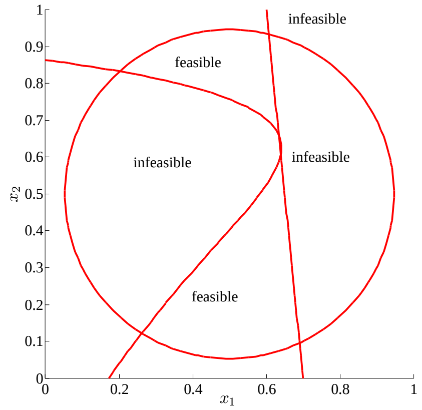

Figure 2: Example 1. Three constraint boundaries.

proposed correlation-based PSVM and the closest distance-based PSVM. The study for varying DOE sizes indicates slightly lower errors for the proposed model in most cases, and a lower average error.

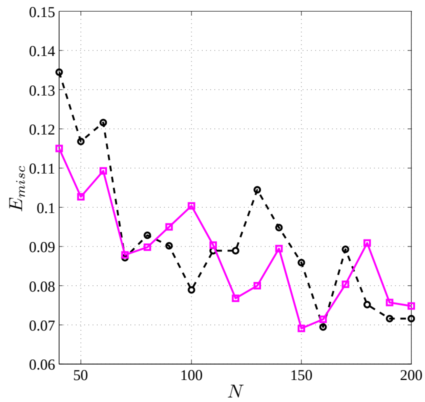

Figure 3: Example 1. Comparison of the error of PSVM models. Black represents the closest distance-based PSVM and magenta is the proposed correlation-based PSVM.

9.1.2. Example 2. Two constraint problem

The constraints for this two variable problem are taken from [27]. The variable ranges are between −2 and 2. The feasible region is an intersection of the following regions:

$$g_1(\mathbf{x}) = -3x_1 + (-3x_2)^3 \leq 0$$
$$g_2(\mathbf{x}) = x_1 - x_2 - 1 \leq 0 \tag{24}$$

The actual constrained boundary is shown in Figure 4 The errors are plotted in Figure 5. For this ex-

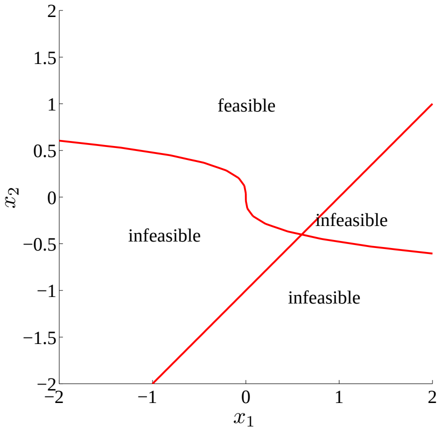

Figure 4: Example 2. Two constraint boundaries.

ample, the correlation-based PSVM shows a definite improvement.

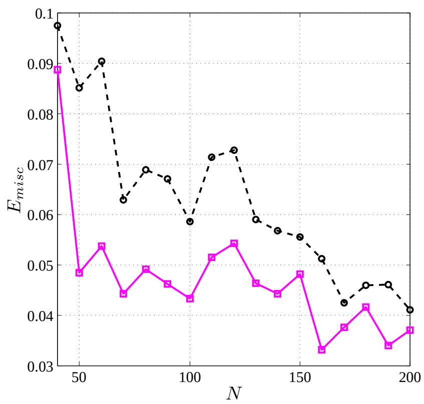

Figure 5: Example 2. Comparison of the error of PSVM models. Black represents the closest distance-based PSVM and magenta is the proposed correlation-based PSVM.

9.2. PSVM-based failure probability confidence intervals - Example 3. Pole crash

In this section, a crash example, from LS-OPT user's manual, is presented to illustrate the calculation of probability of failure confidence intervals due to uncertainty in the failure boundary approximation, in addition to the variable uncertainties. The vehicle model, which crashes into a pole, has five design variables, which are the thicknesses of the hood, roof, bumper, front rail and back rail. These components are displayed in Figure 6. Failure is defined based on three criteria - 15 ms Head Injury Criterion (HIC) calculated at node 432, maximum x-acceleration at node 432, and the intrusion, which is defined as the difference of x-displacement between two nodes (184 and 432).

All the variables have Normal distributions. The mean configuration for calculating the failure probability has thood = 3, troof = 4, tbumper = 3, trailb = 3 and trailf = 3. The variable troof has a standard

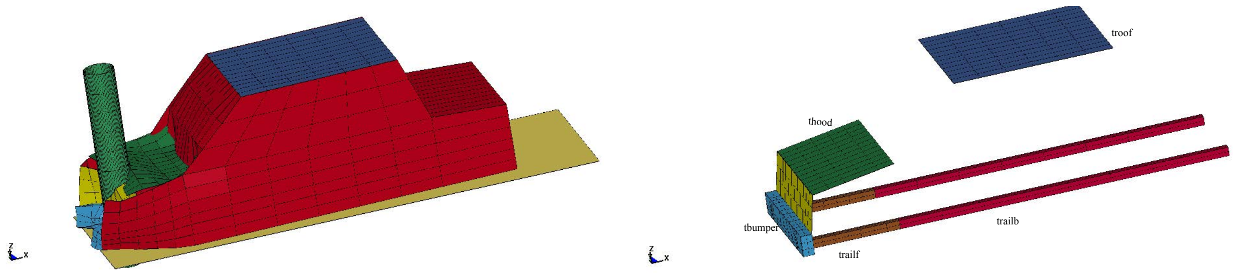

Figure 6: Example 3. Car crashing into a pole (left). Design variables for the car (right).

deviation of 0.05. All other variables have standard deviation equal to 0.1. The constraints are:

$$\begin{aligned} \text{HIC (15 ms) at node 432} &\leq 3800 \\ \text{Maximum Acceleration at node 432} &\leq 6.4 \times 10^5 \\ \text{Intrusion} &\leq 545 \end{aligned} \tag{25}$$

The best known probability of failure is obtained as 0.0828 using direct Monte Carlo simulations. 10^4 LS-DYNA simulations are run at the Monte Carlo samples through LS-OPT for calculating this reference probability.
The SVM boundaries are obtained using 100, 200, 300 and 400 space-filling samples generated and evaluated using LS-OPT and LS-DYNA. The probabilities of failure using SVM are shown with the green curve in Figure 7. The SVM-based failure probability has significant error. The probability of failure is also calculated with the proposed PSVM model (magenta). The failure probability confidence bounds, calculated as explained in section 7, are also shown. Most of the times the failure probability is within the displayed confidence bounds.

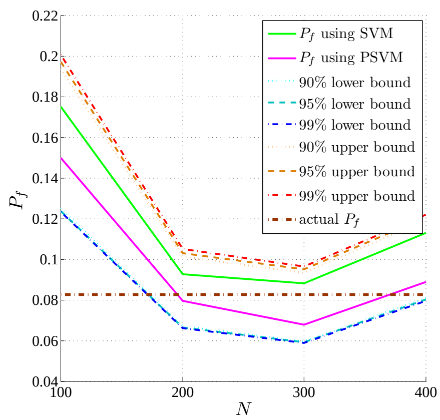

Figure 7: Example 3. Probability of failure and confidence bounds.

9.3. Examples of EGO using PSVM constraints

In this section, results of EGO using the proposed PSVM-based method are presented for two analytical problems. The constraint are the same as those defined in Section 9.1. The results are compared to [15]. For both examples, 10 initial CVT samples are used and the update is run for a 50 iterations to study the convergence. Each iteration consists of 3 samples evaluated in parallel (one sample based on the objective function improvement function and 2 auxiliary samples selected within a local update region using 18). A genetic algorithm (GA) is used to solve the global optimization subproblems for locating the samples (Eq. (15), (14), (16) and (18)).
The following notation is used:

- $\mathbf{x}^\star$ and $f^\star$ are the current predicted optimum (best solution among all evaluate samples) and the corresponding objective function value obtained by the algorithm.

- $\mathbf{x}^{\star}_{actual}$ and $f^{\star}_{actual}$ are the actual optimum (analytical solutions) and the corresponding objective function value.
- ϵ_k is the relative percentage error of the optimum objective function value at the k^{th} iteration.

$$\epsilon_k = \frac{|f^* - f^*_{actual}|}{|f^*_{actual}|} \times 100 \tag{26}$$

9.3.1. Example 4. Three constraint two island problem optimization
The constraints for this example are same as Probem 1. The design variables x_1 and x_2 have identical ranges $[0, 1]$. The optimization problem is:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = -(x_1 - 1)^2 - (x_2 - 0.5)^2 \\ s.t. \quad & g_1(\mathbf{x}) = ((x_1 - 3)^2 + (x_2 + 2)^2)e^{(-x_2^7)} - 12 \leq 0 \\ & g_2(\mathbf{x}) = 10x_1 + x_2 - 7 \leq 0 \\ & g_3(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.2 \leq 0 \end{aligned} \tag{27}$$

The problem has two optima. The global optimum is at $(0.2017, 0.8332)$ with an objective function value -0.7483. The objective function contours, constraint and the optimum are shown in Figure 8. The errors using the proposed method and the method in [15] are plotted in Figure 9. The results of both PSVM-based methods are compared to Kriging in Table 1. The results presented for Kriging-based

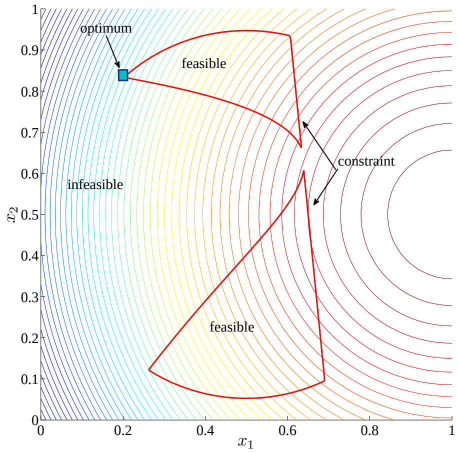

Figure 8: Example 4. Three constraint two island problem - objective function contours, feasible space and optimum.

Method	$\|\|\mathbf{x}^{\star} - \mathbf{x}^{\star}_{actual}\|\|$
Kriging $Probability_v$ [27]	2.2×10^{-4}
Kriging $Probability_s$ [27]	2.2×10^{-4}
Kriging $mean_v$ [27]	2.8×10^{-5}
Kriging $mean_s$ [27]	2.2×10^{-4}
Kriging EV_v [27]	1.8×10^{-1}
Kriging EV_s [27]	2.5×10^{-1}
SVM scheme 1 [15]	9.4×10^{-3}
SVM scheme 2 [15]	4.8×10^{-3}
Proposed method with correlation-based PSVM	2.49×10^{-3}

Table 1: Ex. 4. Comparison of EGO methods. Closest distance to $x^{\star}_{actual}$ after 60 samples.

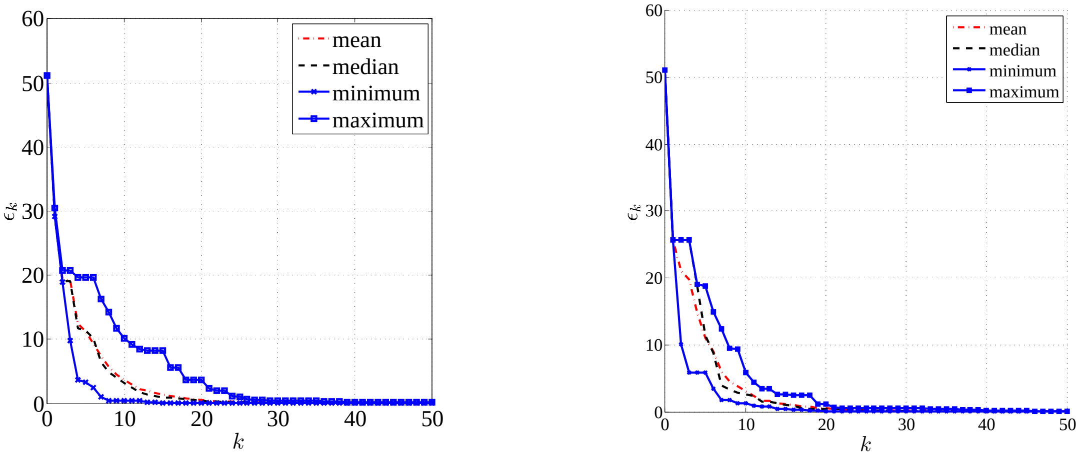

Figure 9: Ex 4. ϵ_k using the closest distance-based PSVM (left) and the proposed method (right).

constraints in [27] were generated using several different methods and the reader is referred to [27] for a more detailed description of the approaches. SVM scheme 1 and SVM scheme 2 in [15] were both based on the closest distance-based PSVM. For each case, the distance between the actual optimum and the optimum found by the algorithms is provided after 50 adaptive sample evaluations. These values are compared to those using the proposed and previous SVM-based methods after the same number of evaluations. This corresponds to the 17^{th} iteration of these methods. It should be noted that the samples are evaluated in parallel in these methods.

9.3.2. Example 5. Two constraint problem optimization

This two variable optimization problem has identical ranges [-2,2] for both variables. The objective function, the constraints and the optimum solution are depicted in Figure 10. The actual optimum is at $(0.5955, -0.4045)$ with an objective function value of 289.85. The optimization problem is defined as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = (1 + A(x_1 + x_2 + 1)^2)(30 + B(2x_1 - 3x_2)^2) \\ \text{where} \quad & A = 19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2, \\ \text{and} \quad & B = 18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = -3x_1 + (-3x_2)^3 \leq 0 \\ & g_2(\mathbf{x}) = x_1 - x_2 - 1 \leq 0 \end{aligned} \tag{28}$$

The errors in objective function prediction are plotted in Figure 11.

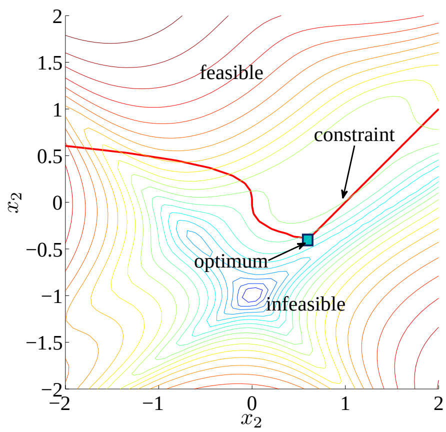

Figure 10: Example 5. Two constraint problem - objective function contours, feasible space and optimum.

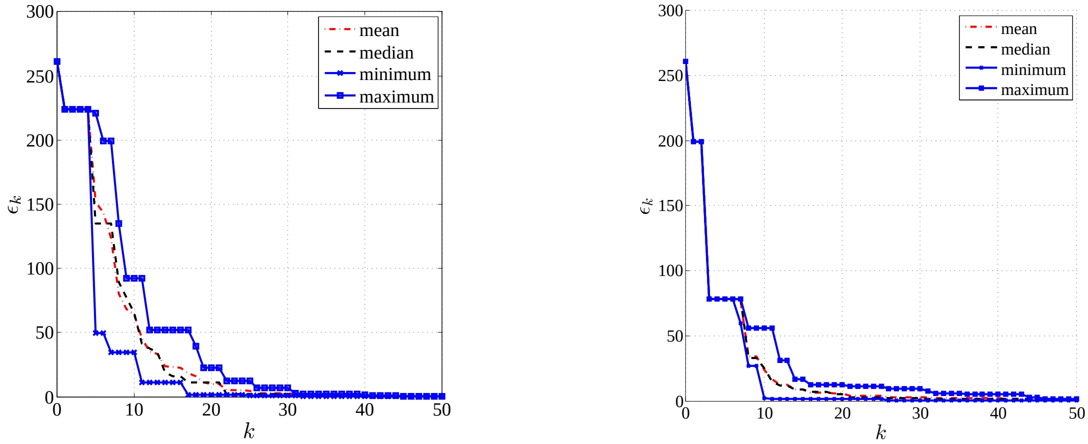

Figure 11: Ex. 5. ϵ_k using the closest distance-based PSVM (left) and the proposed method (right).

9. Conclusion

A classification-based method to quantify constraint violations in terms of probability of violation is presented. More specifically, an improved probabilistic support vector machine model is presented to give probability outputs for an SVM classification. Because the method is classification-based, it alleviates some of the issues associated with discontinuous and binary responses, and multiple failure modes. The main change in the PSVM compared to the previous closest distance-based PSVM is that it considers the effect of all samples to predict the probability of belonging to certain class, instead of just considering the closest sample from each class. The effect of a sample on the probability reduces as the distance increases, based on a Gaussian correlation function. The proposed PSVM model is seen to provide lower errors, based on two analytical examples. The PSVM model is used in an EGO formulation to define the constraints. Two examples are presented for comparison to the previous approach. The proposed method has slightly less average error, although not remarkable. In addition, a method to calculate corrected failure probability confidence intervals for MCS is presented for the case when failure boundaries are approximated using SVM. The efficacy of the method is demonstrated using a car crash example with five random design variables.

References

[1] T. W. Simpson, V. V. Toropov, V. O. Balabanov, and F. A. C. Viana. Design and analysis of computer experiments in multidisciplinary design optimization: A review of how far we have come - or not. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Colombia, Canada, September 2008.

[2] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P.K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28, 2005.

[3] D. Huang, T.T. Allen, W.I. Notz, and R.A. Miller. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32(5):369–382, 2006.

[4] F.A.C. Viana, C. Gogu, and R.T. Haftka. Making the most out of surrogate models: tricks of the trade. In *ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Montreal, Canada, 2010.

[5] G.G. Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4):370–380, 2007.

[6] J.P.C. Kleijnen. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707–716, 2009.

[7] R. Jin, X. Du, and W. Chen. The use of metamodeling techniques for optimization under uncertainty. *Structural and Multidisciplinary Optimization*, 25(2):99–116, 2003.

[8] D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[9] G.G. Wang, L. Wang, and S. Shan. Reliability assessment using discriminative sampling and metamodeling. *SAE Transactions, Journal of Passenger Cars - Mechanical Systems*, 114:291–300, 2005.

[10] B.J. Bichon, M.S. Eldred, L.P. Swiler, S. Mahadevan, and J.M. McFarland. Multimodal reliability assessment for complex engineering applications using efficient global optimization. In *Proceedings of the* 48^{th} *AIAA/ASME/ASCE/AHS/ASC conference on Structures, Dynamics and Materials. Paper AIAA-2007-1946.*, Honolulu, Hawaii, April 2007.

[11] B.J. Bichon, J.M. McFarland, and S. Mahadevan. Applying egra to reliability analysis of systems with multiple failure modes. In *Proceedings of the* 51^{st} *AIAA/ASME/ASCE/AHS/ASC conference on Structures, Dynamics and Materials. Paper AIAA-2010-2852.*, Orlando, Florida, April 2010.

[12] S. Missoum, P. Ramu, and R.T. Haftka. A convex hull approach for the reliability-based design of nonlinear transient dynamic problems. *Computer Methods in Applied Mechanics and Engineering*, 196(29):2895–2906, 2007.

[13] A. Basudhar, S. Missoum, and A. Harrison Sanchez. Limit state function identification using Support Vector Machines for discontinuous responses and disjoint failure domains. *Probabilistic Engineering Mechanics*, 23(1):1–11, 2008.

[14] R.B. Gramacy and H.K.H. Lee. Optimization under unknown constraints. *Arxiv preprint arXiv:1004.4027*, 2010.

[15] Anirban Basudhar, Christoph Dribusch, Sylvain Lacaze, and Samy Missoum. Constrained efficient global optimization with support vector machines. *Structural and Multidisciplinary Optimization*, 46(2):201–221, 2012.

[16] V.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[17] S.R. Gunn. Support vector machines for classification and regression. Technical Report ISIS-1-98, Department of Electronics and Computer Science, University of Southampton, 1998.

[18] J.C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[19] Anirban Basudhar and Samy Missoum. Reliability assessment using probabilistic support vector machines. *International Journal of Reliability and Safety*, 2013.

[20] J.P. Chiles and P. Delfiner. *Geostatistics: modeling spatial uncertainty*. Wiley-Interscience, 1999.

[21] T.J. Santner, B.J. Williams, and W. Notz. *The design and analysis of computer experiments*. Springer Verlag, 2003.

[22] J.D. Martin and T.W. Simpson. Use of kriging models to approximate deterministic computer models. *AIAA journal*, 43(4):853–863, 2005.

[23] B.J. Bichon. *Efficient Surrogate Modeling for Reliability Analysis and Design*. PhD thesis, Vanderbilt University, 2010.

[24] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[25] M. Schonlau. *Computer experiments and global optimization*. PhD thesis, Department of Statistics, University of Waterloo, Ontario, Canada, 1997.

[26] M.J. Sasena, P.Y. Papalambrose, and P. Goovaerts. Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization*, 34:263–278, 2002.

[27] M.J. Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Optimization with Kriging Approximations.* PhD thesis, Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, 2002.

[28] A.I.J. Forrester, A. Sobester, and A.J. Keane. *Engineering design via surrogate modelling: a practical guide.* Wiley, 2008.

[29] N. Henkenjohann and J. Kunert. An efficient sequential optimization approach based on the multivariate expected improvement criterion. *Quality Engineering*, 19(4):267–280, 2007.

[30] W. Ponweiser, T. Wagner, and M. Vincze. Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models. In *IEEE Congress on Evolutionary Computation*, Hong Kong, 2008.

[31] D. Ginsbourger, R. Le Riche, and L. Carraro. A multi-points criterion for deterministic parallel global optimization based on kriging. In *International Conference on Nonconvex Programming, NCP07*, December 2007.

[32] F. Viana, R. Haftka, and L.T. Watson. Why not run the efficient global optimization algorithm with multiple surrogates? In *Proceedings of the 51th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA, Orlando, FL, USA. AIAA–2010–3090.*, 2010.

[33] C. Audet, Jr. J.E. Dennis, D.W. Moore, A. Booker, and P.D. Frank. A surrogate based method for constrained optimization. In *8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization. Paper number AIAA-2000-4891*, 2000.

[34] V. Picheny, N.H. Kim, R.T. Haftka, and N.V. Queipo. Conservative predictions using surrogate modeling. In *Proceedings of the 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Schaumburg, Illinois, USA*, pages 7–10, 2008.

[35] H. Lee, R. Gramacy, C. Linkletter, and G. Gray. Optimization subject to hidden constraints via statistical emulation. Technical report, Tech. Rep. UCSC-SOE-10-10, University of California, Santa Cruz, Department of Applied Mathematics and Statistics, 2010.

[36] N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods.* Cambridge university press, 2006.

[37] B. Scholkopf and A.J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* the MIT Press, 2002.

[38] H.T. Lin, C.J. Lin, and R.C. Weng. A note on Platts probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, 2007.