Non-parametric stochastic subset optimization utilizing multivariate boundary kernels and adaptive stochastic sampling

Gaofeng Jia and Alexandros A. Taflanidis¹

Department of Civil & Environmental Engineering & Earth Sciences
University of Notre Dame, USA

Abstract

The implementation of NP-SSO (Non-Parametric Stochastic Subset Optimization) to general design under uncertainty problems and its enhancement through various soft computing techniques is discussed. NP-SSO relies on iterative simulation of samples of the design variables from an auxiliary probability density, and approximates the objective function through kernel density estimation (KDE) using these samples. To deal with boundary correction in complex domains, a multivariate boundary KDE based on local linear estimation is adopted in this work. Also, a non-parametric characterization of the search space at each iteration using a framework based on support vector machine is formulated. To further improve computational efficiency, an adaptive kernel sampling density formulation is integrated and an adaptive, iterative selection of the number of samples needed for the KDE implementation is established.

Keywords: non-parametric stochastic subset optimization, optimization under uncertainty, kernel density estimation, adaptive kernel sampling density, stochastic sampling.

1 Introduction

In any engineering design application, the performance predictions for the system under consideration involve some level of uncertainty, stemming from our incomplete knowledge about the system itself and its environment (representing future excitations). Explicitly accounting for these uncertainties is important for providing optimal configurations that exhibit robust-performance [1-3]. A probabilistic approach provides a rational and consistent framework for performing this task [4], employing probability models to characterize different possible model-parameters values. In this setting, the design objective is typically related to the expected value of a system performance measure, such as failure probability or expected life-cycle cost [2, 5-8]. For applications involving complex system models this expected value can rarely be calculated or accurately approximated analytically [9]. For such applications stochastic simulation techniques, which pose no constraints on the complexity of the adopted numerical and probability models, are frequently the only applicable generalized approach [10, 11] for

^{*}Corresponding author: Associate Professor, University of Notre Dame, 156 Fitzpatrick Hall, Notre Dame, IN, 46556 email:a.taflanidis@nd.edu

1 estimating the objective function. This approach, though, entailing a large number of
2 evaluations of the system performance for each objective function estimation, might
3 impose a computational cost that is prohibitive for applications involving computationally
4 expensive models.

5 To address this challenge, this paper investigates an alternative optimization algorithm
6 for system design optimization under uncertainty, termed Non-parametric Stochastic
7 Subset Optimization (NP-SSO). NP-SSO, proposed recently [12] for design problems
8 utilizing the system reliability as the objective function, also relies on stochastic
9 simulation, but rather than calculating the performance objective for specific values of the
10 design variables, establishes a global approximation for it. It is an extension of the SSO
11 algorithm [13], and like SSO it relies on simulation of a sufficient number of samples of
12 the design variables from an auxiliary probability density function, treating them
13 artificially as uncertain. Rather than using the information in these samples to establish an
14 approximation for the average value of the objective function over preselected subsets, as
15 established in SSO, NP-SSO utilizes kernel density estimation (KDE) [14] to approximate
16 the objective function and identify candidate points for the global minimum. An iterative
17 approach is also established within NP-SSO to improve computational efficiency: at each
18 iteration a new compact domain is identified as candidate subset for the optimal design
19 variables and the search is then confined within this domain. A parametric description for
20 that subset, corresponding to a box-bounded domain, was adopted in [12] to support the
21 proposed KDE implementation.

22 In this paper, based upon [15], NP-SSO is extended to general design under uncertainty
23 problems (not constrained to reliability-optimization), and more importantly it is coupled
24 with various soft-computing techniques to improve its non-parametric characteristics and
25 its numerical efficiency. Specifically, the following advances are introduced here for NP-
26 SSO. A versatile, non-parametric characterization of the subset identified at each iteration
27 is investigated through adoption of KDE with multivariate boundary kernels (KDE-MBK)
28 [16]. To achieve this goal, a support vector machine (SVM) [17] is additionally adopted to
29 facilitate (i) efficient simulation of samples and (ii) estimation of boundary correction
30 terms within the proposed iterative scheme. Furthermore, an adaptive kernel sampling
31 density (AKSD) is adopted to improve the efficiency of the stochastic sampling stage that
32 is required within NP-SSO to generate the necessary samples from the design variables.
33 The characteristics for the AKSD are chosen here to explicitly optimize the anticipated
34 sampling efficiency, utilizing readily available information to perform this optimization.
35 Finally, an adaptive selection of the number of samples needed for the KDE
36 approximation is proposed. This is established through a multi-stage process; a new set of
37 samples is obtained at each stage and then a new KDE approximation is established and a
38 new domain identified (within the same always iteration of NP-SSO). If this domain does
39 not differ significantly from the previous one (obtained utilizing a smaller number of
40 samples) the KDE approximation has sufficient accuracy and the multi-stage sampling
41 stops. The advances proposed here, especially the AKSD and the adaptive selection of the
42 samples for the KDE approximation, contribute greatly to the efficiency of NP-SSO.

43 In the next section the general design under uncertainty problem is reviewed and in
44 Section 3 the NP-SSO framework is presented for such problems. The proposed advances
45 for facilitating complex subset selection are developed in Section 4 whereas in Section 5
46 an adaptive stochastic sampling implementation is seamlessly integrated in the framework.
47 In Section 6 the adaptive selection of the number of samples for KDE-MBK is presented
48 and the overall adaptive NP-SSO algorithm is reviewed. Finally, Section 7 presents an
49 illustrative example.

2 Design under uncertainty optimization

Consider a system that involves some controllable parameters that define its design, referred to also as design variables and let $\mathbf{x}=[x_1 \ x_2 \ \dots \ x_{n_x}] \in X \subset \mathfrak{R}^{n_x}$ be the design vector where X denotes the bounded admissible design space. Let $\boldsymbol{\theta}=[\theta_1 \ \theta_2 \ \dots \ \theta_{n_\theta}]$ lying in $\Theta \subset \mathfrak{R}^{n_\theta}$ be the vector of uncertain model parameters for the system, where Θ denotes the set of their possible values. A PDF (probability density function) $p(\boldsymbol{\theta})$, which incorporates our available knowledge about the system and its excitation, is assigned to these parameters. The performance measure characterizing the augmented system model is $h(\mathbf{x}, \boldsymbol{\theta}) \geq 0: \mathfrak{R}^{n_x} \times \mathfrak{R}^{n_\theta} \rightarrow \mathfrak{R}^+$. Herein $h(\mathbf{x}, \boldsymbol{\theta})$ will be assumed to be a positive function; design problems for which this is not true can be converted to this type of applications by simply adding a sufficient positive constant [11]. Also the convention that smaller values of $h(\mathbf{x}, \boldsymbol{\theta})$ correspond to more favorable performance is adopted, i.e. $h(\mathbf{x}, \boldsymbol{\theta})$ corresponds to a certain cost function. Different choices for $h(\mathbf{x}, \boldsymbol{\theta})$ facilitate different design approaches, with reliability-based design [12], for which $h(\mathbf{x}, \boldsymbol{\theta})$ corresponds to an indicator function, being a specific special case. Here we focus on general robust-to-uncertainties design, i.e. no specific form is assumed for $h(\mathbf{x}, \boldsymbol{\theta})$.

The probabilistic performance for this design problem is given by the expected value

$$C(\mathbf{x}) = E_{p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})] = \int_{\Theta} h(\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (1)$$

where $E_{p(\boldsymbol{\theta})}[\cdot]$ denotes expectation with respect to $p(\boldsymbol{\theta})$. We are then interested in the minimization problem:

$$\mathbf{x}^* = \underset{\mathbf{x} \in X}{\operatorname{argmin}} C(\mathbf{x}) \quad (2)$$

where any constraints are taken into account by appropriate definition of X . If the models considered are complex and include a large number of uncertain model parameters the multi-dimensional integral (1) involved in this optimization cannot be calculated, or even accurately approximated, analytically. An efficient alternative approach is to estimate the integral by stochastic simulation [18]. Using a finite number, n_t , of samples of $\boldsymbol{\theta}$ drawn from some chosen density $q_s(\boldsymbol{\theta})$, with $\boldsymbol{\theta}^j$ denoting the j^{th} sample, the objective function for (2) can be calculated as

$$\hat{C}(\mathbf{x}) = \frac{1}{n_t} \sum_{j=1}^{n_t} h(\mathbf{x}, \boldsymbol{\theta}^j) \frac{p(\boldsymbol{\theta}^j)}{q_s(\boldsymbol{\theta}^j)} \quad (3)$$

Though many efficient methodologies and stochastic optimization algorithms [11, 19] have been developed for the resultant design problem, it is widely acknowledged as a challenging one due to the significant computational cost for evaluating the objective function (3), which involves n_t system evaluations. This is particularly true for problems involving complex, black-box models, for which the numerical evaluation of the system response to calculate $h(\mathbf{x}, \boldsymbol{\theta})$ can be high and gradient information cannot be easily obtained. NP-SSO is an efficient alternative stochastic optimization algorithm for such exact problems, involving computational expensive numerical models for estimation of $h(\mathbf{x}, \boldsymbol{\theta})$. It is also based on stochastic simulation and stochastic sampling concepts but avoids direct estimation of (3) for specific design variables.

3 NP-SSO for general design under uncertainty problems

This section discusses the extension of NP-SSO [12] for problems like (1). The basics of the extension follows the approach in [11] (established for SSO), though various additions are required here to facilitate the direct approximation to the objective function required within NP-SSO. The focus is on these latter characteristics.

3.1 NP-SSO formulation

An augmented design problem is considered, for which the design variables are artificially considered as uncertain with uniform distribution $p(\mathbf{x})=1/V_I$ over some bounded region I that defines the search space, with V_I denoting the volume of set I . For the first iteration of the algorithm this region will be simply the admissible design space X . In this context, the following auxiliary PDF is defined

$$\pi_I(\mathbf{x}, \boldsymbol{\theta}) = \frac{h(\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) p(\mathbf{x})}{\int_I \int_{\Theta} h(\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) p(\mathbf{x}) d\boldsymbol{\theta} d\mathbf{x}} = \frac{h(\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) p(\mathbf{x})}{E_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_I} \propto h(\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) \quad (4)$$

where $\propto$ denotes proportionality, the subscript in π_I is used to denote the fact that $\mathbf{x}$ belongs in I and the normalization constant $E_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_I$ denotes expectation with respect to $p(\boldsymbol{\theta})$ and $p(\mathbf{x})$ when $\mathbf{x}$ belongs in I . The latter expression is also equal to the average value of the objective function over I , $\int_I C(\mathbf{x}) d\mathbf{x} / V_I$ [11].

Utilizing this auxiliary PDF, the objective function $C(\mathbf{x})$ is expressed as [11]:

$$C(\mathbf{x}) = \pi_I(\mathbf{x}) \left[\frac{E_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_I}{p(\mathbf{x})} \right] = \pi_I(\mathbf{x}) \left[E_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_I V_I \right] \quad (5)$$

where $\pi_I(\mathbf{x})$ corresponds to the marginal PDF for $\mathbf{x}$, $\pi_I(\mathbf{x}) = \int_{\Theta} \pi_I(\mathbf{x}, \boldsymbol{\theta}) d\boldsymbol{\theta}$ and the quantity in the brackets in (5) is simply a proportionality constant.

Based on (5) the variations of $C(\mathbf{x})$ and $\pi_I(\mathbf{x})$ with respect to $\mathbf{x} \in I$ are proportional and ultimately NP-SSO is established by exploiting the information in samples from the latter PDF, $\{\mathbf{x}_h\} = \{\mathbf{x}_h^j; j=1, \dots, n_s\}$ with $\mathbf{x}_h^j$ denoting the j^{th} sample and n_s denoting the total number of samples (adaptive selection of n_s is addressed later in Section 6). Such samples can be obtained by initially simulating samples $\{\mathbf{x}_h, \boldsymbol{\theta}_h\} = \{\mathbf{x}_h^j, \boldsymbol{\theta}_h^j; j=1, \dots, n_s\}$ of the pair $\{\mathbf{x}, \boldsymbol{\theta}\}$ from the joint distribution $\pi_I(\mathbf{x}, \boldsymbol{\theta})$ in (4) using any appropriate stochastic sampling algorithm. The $\mathbf{x}$ component of these samples corresponds then to samples from the marginal distribution $\pi_I(\mathbf{x})$. Different stochastic sampling algorithms can be used for this purpose, with requirement in the context of NP-SSO, though, that they produce independent samples. An approach for efficiently generating such samples will be discussed in detail in Section 5. Note that in this process the system response will be evaluated in a large set $\{\mathbf{x}_c, \boldsymbol{\theta}_c\} = \{\mathbf{x}_c^j, \boldsymbol{\theta}_c^j; j=1, \dots, N\}$ of samples with $\mathbf{x}_c^j$ and $\boldsymbol{\theta}_c^j$ corresponding to the j^{th} sample of $\mathbf{x}$ and $\boldsymbol{\theta}$, respectively, and N to the total number of trials established in the stochastic sampling process. Figure 1 illustrates this concept for an example with a two dimensional $\mathbf{x}$: part (a) shows the objective function whereas part (b) samples obtained from $\pi_I(\mathbf{x})$.

Using KDE, an approximation to $\pi_I(\mathbf{x})$ is established within NP-SSO, denoted $\tilde{\pi}_I(\mathbf{x})$ herein (more details on this will be provided in Section 4.1), and is then used to approximate the objective function as

$$\tilde{C}(\mathbf{x}) = \tilde{\pi}_I(\mathbf{x}) \left[\hat{E}_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_I V_I \right] \quad (6)$$

This approximation requires additionally an estimate for the proportionality constant in the brackets. As suggested in [12] this can be established through stochastic simulation; using the information already available from getting the samples from $\pi_I(\mathbf{x}, \boldsymbol{\theta})$ [through set $\{\mathbf{x}_c^j, \boldsymbol{\theta}_c^j\}$] the approximation is

$$\hat{E}_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_I = \frac{1}{N} \sum_{j=1}^N h(\mathbf{x}_c^j, \boldsymbol{\theta}_c^j) \frac{p(\boldsymbol{\theta}_c^j)}{q(\boldsymbol{\theta}_c^j)} \quad (7)$$

where $\mathbf{x}_c^j \sim p(\mathbf{x})$, $\boldsymbol{\theta}_c^j \sim q(\boldsymbol{\theta})$ with $q(\boldsymbol{\theta})$ being the proposal density chosen for $\boldsymbol{\theta}$ in the stochastic sampling algorithm for $\boldsymbol{\theta}$, and as discussed above N is the total number of evaluations performed within this algorithm (see also discussion in Section 5). Part (c) of Figure 1 shows $\tilde{C}(\mathbf{x})$ using the samples shown in Part (b) of that figure. In this example, NP-SSO provides a good overall approximation of the objective function around both local minima.

For establishing approximation (6) the most challenging component is obtaining the samples from (4), at least for problems involving complex models for which the computational cost for establishing the KDE approximation is small compared to the cost for evaluating the system-model response. Advances for enhancing the efficiency for this task will be offered in Sections 5 and 6 of this manuscript.

3.2 Iterative approach

The identification of local optimal within NP-SSO is established by using the approximation of the objective function in (6). To improve the efficiency of NP-SSO the iterative framework established in [12] is adopted, restricting the search at each iteration to the subset that was identified in the previous iteration to correspond to the lower values of the objective function. This is motivated by the fact that relatively smaller number of samples will be available in the regions close to the global minimum requiring a large n_s (and thus model simulations) to gather accurate information for the objective function in these regions if the search is constrained within the initial design domain X . For this purpose, the following optimal subset is identified

$$I^* = \{I^* \subset I \mid \tilde{C}(\mathbf{x}) \leq c^h\} \quad (8)$$

corresponding to the design vectors with objective function values smaller than threshold c^h . Details for characterization of I^* will be discussed in Section 4.2. The choice of threshold c^h determines the relative size of the subset I^* and can be adaptively chosen to achieve a specific size (volume) ratio ρ between I^* and I

$$\delta(I^* \mid I) = V_{I^*} / V_I = \rho \quad (9)$$

Selection of δ is a trade-off between the number of samples required in each step and the number of steps required to converge to the optimal design choice [12, 13]. Part (d) of Figure 1 earlier illustrates this identification; it shows the region I^* that would be identified by NP-SSO for a target volume reduction of either 10% or 20%. From Figure 1 it is evident that NP-SSO can efficiently identify the regions (contours) of small objective function values close to both local minima in this example.

Through this iterative implementation NP-SSO converges gradually to regions closer to the local optima, trying to establish a higher accuracy approximation for the objective function in these regions. This is ultimately facilitated by building a higher accuracy KDE approximation within these regions [12]. For assessing the convergence of this iterative implementation the ratio of the average values for the objective function between I^* and I is used [11, 12], given by

$$H(I^* | I) = \frac{\int_{I^*} C(\mathbf{x}) d\mathbf{x} / V_{I^*}}{\int_I C(\mathbf{x}) d\mathbf{x} / V_I} = \frac{E_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_{I^*}}{E_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_I} \quad (10)$$

This ratio characterizes the average sensitivity of the objective function at the current iteration with values close to 1 indicating convergence to regions close to the minimum for the objective function, where the objective function has almost a constant value (small sensitivity). It can be estimated through stochastic simulation [12] as

$$\hat{H}(I^* | I) = \frac{\frac{1}{n_o} \sum_{j=1}^{n_o} h(\mathbf{x}_o^j, \boldsymbol{\theta}^j) \frac{p(\boldsymbol{\theta}^j)}{q(\boldsymbol{\theta}^j)}}{\frac{1}{n_u} \sum_{j=1}^{n_u} h(\mathbf{x}_u^j, \boldsymbol{\theta}^j) \frac{p(\boldsymbol{\theta}^j)}{q(\boldsymbol{\theta}^j)}} \quad (11)$$

where $\boldsymbol{\theta}^j \sim q(\boldsymbol{\theta})$ and $\mathbf{x}_u^j$ and $\mathbf{x}_o^j$ uniform samples in I and I^* , respectively. For avoiding new evaluations of the system response for calculation of (11) the sample set $\{\mathbf{x}_u\}$ can be chosen as $\{\mathbf{x}_c\}$ with their counterpart sample of $\boldsymbol{\theta}$ in (11) the corresponding values in set $\{\boldsymbol{\theta}_c\}$. This way estimation (11) is based on model response evaluations already performed.

4 Extensions for NP-SSO to accommodate complex search domains

For the description of the search space within the iterations of NP-SSO an enhanced approach that can support characterization of complex domains is established in this work. This has two components; adoption of KDE for the approximation $\tilde{\pi}_I(\mathbf{x})$ that can facilitate boundary correction in such complex domains, and efficient description of the optimal subsets I^* through SVM.

4.1 Advances in KDE approximation of $\pi_I(\mathbf{x})$ through multivariate boundary kernel

The necessity to consider boundary correction in the KDE is the fact that the objective function [and thus $\pi_I(\mathbf{x})$] is expected to take large values in the boundary of the search space (assuming optimum is an interior point) and the search space can have a complex

boundary. In [12] a reflection approach was suggested for this purpose that relied, though, on I being a box-bounded region, which created significant challenges in efficient reduction of the search space in each iteration of NP-SSO, especially for high-dimensional problems. By contrast, here KDE with multivariate boundary kernels (KDE-MBK) based on local linear estimation is adopted. KDE-MBK can be applied as nonparametric multivariate density estimation in domains with arbitrary boundary regions [16]. This is demonstrated in Figure 2, which shows a KDE-MBK approximation to a density with an elliptical boundary. It is evident that the approximation without the correction (i.e. regular KDE) significantly underestimates the density close to the boundary and would lead to erroneous results if adopted. KDE-MBK provides the necessary correction, and it also facilitates adoption of such complex boundaries (elliptical in this case) which was not possible with the reflection methods used previously [12].

If n_s samples $\{\mathbf{x}_h\}$ are available for $\mathbf{x}$ from $\pi_I(\mathbf{x})$ with $\mathbf{x}_{hi}^j$ denoting the j^{th} sample of the i^{th} design variable, the KDE-MBK estimator [16] is given by utilizing a corrected Kernel $\underline{K}$

$$\begin{aligned}\tilde{\pi}_I(\mathbf{x}) &= \frac{1}{n_s} \sum_{j=1}^{n_s} \underline{K}_m(\mathbf{x} - \mathbf{x}_h^j) \\ \underline{K}_m(\mathbf{x} - \mathbf{x}_h^j) &= \frac{K_m(\mathbf{x} - \mathbf{x}_h^j) - K_m(\mathbf{x} - \mathbf{x}_h^j) [\mathbf{x} - \mathbf{x}_h^j]^T \mathbf{D}^{-1}(\mathbf{x}) \mathbf{c}_1(\mathbf{x})}{c_0(\mathbf{x}) - \mathbf{c}_1^T(\mathbf{x}) \mathbf{D}^{-1}(\mathbf{x}) \mathbf{c}_1(\mathbf{x})}\end{aligned}\quad (12)$$

where for the uncorrected multivariate kernel, K_m , a product kernel is adopted

$$K_m(\mathbf{x} - \mathbf{x}_h^j) = \prod_{i=1}^{n_x} \frac{1}{w_i} K\left(\frac{x_i - x_{hi}^j}{w_i}\right) \quad (13)$$

with $K(\cdot)$ some chosen univariate kernel and w_i the bandwidth parameter for the i^{th} design variable defining the spread of the kernel. The support region for K_m (region that it influences based on its bandwidth) is denoted by $S(\mathbf{x})$. The constant $c_0(\mathbf{x})$, vector $\mathbf{c}_1(\mathbf{x}) = [c_{11}(\mathbf{x}) \cdots c_{1n_x}(\mathbf{x})]^T$ and matrix $\mathbf{D}(\mathbf{x})$ with elements $d_{kl}(\mathbf{x})$ in (12) are given, respectively, by

$$\begin{aligned}c_0(\mathbf{x}) &= \int_{\Delta_I(\mathbf{x})} K_m(\mathbf{x} - \boldsymbol{\xi}) d\boldsymbol{\xi}, \quad c_{1i}(\mathbf{x}) = \int_{\Delta_I(\mathbf{x})} K_m(\mathbf{x} - \boldsymbol{\xi}) (x_i - \xi_i) d\boldsymbol{\xi} \\ d_{kl}(\mathbf{x}) &= \int_{\Delta_I(\mathbf{x})} K_m(\mathbf{x} - \boldsymbol{\xi}) (x_k - \xi_k)(x_l - \xi_l) d\boldsymbol{\xi}\end{aligned}\quad (14)$$

Here $\Delta_I(\mathbf{x})$ is the part of the support region $S(\mathbf{x})$ for the Kernel $K_m(\mathbf{x})$ that also belongs to I . If the whole part of $\Delta_I(\mathbf{x})$ is inside I then $\mathbf{x}$ is an interior point and no correction is necessary $\underline{K} = K_m$; otherwise, it belongs to the boundary region and estimation of $\underline{K}$ requires the variables in (14) which for complex boundaries can be performed using Monte Carlo integration and a uniform proposal density in $\Delta_I(\mathbf{x})$. Numerical characteristics related to this approach will be discussed later.

In this study the Epanechnikov kernel [10] is chosen

$$K(r) = \begin{cases} \frac{3}{4}(1-r^2) & \text{if } -1 \leq r \leq 1 \\ 0 & \text{else} \end{cases} \quad (15)$$

with bandwidth [12] for each component selected as

$$w_i^{opt} = \left[25 \left(\frac{3}{5} \right)^{n_x} (2\sqrt{\pi})^{n_x} \right]^{1/(n_x+4)} \left[\frac{4}{(n_x+2)n_s} \right]^{1/(n_x+4)} \sigma_i \quad (16)$$

where σ_i corresponds to the standard deviation for the samples $\{\mathbf{x}_h\}$ for x_i .

4.2 Optimal subset characterization and approximation by support vector machine (SVM)

The identification (8) and its utilization as search space in the next iteration require some means to efficiently characterize the subset I^* . To avoid a parametric description for I^* a characterization based on samples is adopted here. This is established through the following approach: first generate n_u samples of $\mathbf{x}$ from the uniform distribution $p(\mathbf{x})$, denoted $\{\mathbf{x}_u\} = \{\mathbf{x}_u^j; j=1, \dots, n_u\}$ with $\mathbf{x}_u^j$ corresponding to the j^{th} sample, then evaluate $\tilde{\pi}_l(\mathbf{x})$ for them through (12), order these samples based on the values of $\tilde{C}(\mathbf{x})$, select c^h as the value corresponding to the ρ -percentile, and keep only these lower ρ -percentile samples. These samples, denoted $\{\mathbf{x}_o\} = \{\mathbf{x}_o^j; j=1, \dots, n_o\}$, with $\mathbf{x}_o^j$ corresponding to the j^{th} sample and $n_o = \rho n_u$, are then uniformly distributed in I^* whereas $\delta(I^* | I) = \rho$. They ultimately correspond to points uniformly distributed within I^* and constitute candidate optimal solutions for the design problem. Figure 3 illustrates this concept for the same example as in Figure 1. Part (a) shows uniform samples in I , whereas part (b) samples in I^* for selection of $\rho = 10\%$. Note that the evaluation of $\tilde{C}(\mathbf{x}_u)$ for samples $\{\mathbf{x}_u\}$ can be performed at a small computational cost, as it requires no additional structural analyses [only KDE for (12)], indicating that a large number can be used for n_u . Thus the characterization of I^* through candidate optimal points can be efficiently established.

In [12], and constrained by the KDE approach adopted, a box-bounded superset of I^* was used as search space for the subsequent iteration, contributing to a reduction of efficiency in high-dimensional problems. As discussed earlier, the KDE-MBK approach utilized here allows for directly using I^* as the search space, facilitating a truly non-parametric description (I can take any possible form), based only on samples following the constraint (8). This non-parametric description though, creates an additional computational challenge for deciding if any new design vector $\mathbf{x}$ belongs in I^* since it involves estimation of $\tilde{C}(\mathbf{x})$, rather than a simple geometric description. This makes evaluation of the integrals in (14) intensive since this is established through Monte Carlo integration, requiring a decision on whether each of these samples [used within the Monte Carlo (MC) integration] will belong in the search space. To avoid estimation of $\tilde{C}(\mathbf{x})$ for each of these samples, a SVM [17] is utilized here to classify whether a sample is inside I^* or not. This is established through the following process:

Step 1: After identification of set I^* , a box bounded region (superset of I^*) B is established with lower and upper bounds corresponding to the minimum and maximum value (in each dimension) of the uniform samples in I^*

Step 2: Then samples from the set $\{\mathbf{x}_u\}$ are identified that are within B but outside I^* . This sample set is denoted by $\{\mathbf{x}_{-o}\}$

Step 3: A membership $\{\mathbf{x}_o, \mathbf{x}_{-o}\} \Rightarrow \{1, 0\}$ is assigned to the total uniform samples $\{\mathbf{x}_{u,B} := \mathbf{x}_u^j \in B\} = \{\mathbf{x}_o, \mathbf{x}_{-o}\}$ in B , with 1 meaning inside I^* and 0 outside of I^*

Step 4: Finally a SVM is developed for classification of any new candidate sample based on this membership information. This idea is demonstrated in part (c) of Figure 3.

This SVM can be then utilized to decide the membership of any new sample $\mathbf{x}$ that will be used in the next iteration of the algorithm (stochastic sampling) and for estimation of the integrals in (14). The latter is established following the approach presented in detail in Appendix A.

Another topic that needs to be addressed in the characterization of I^* is existence of multiple local minima. In such cases the samples $\{\mathbf{x}_o\}$ will be separated in multiple clusters, and KDE considering all of them together will lead to inaccurate results due to bad estimation of optimal bandwidth caused by the multimodality of the PDF. To avoid this problem, and following the approach in [12], a clustering algorithm is adopted that can identify the existence of multiple clusters and separate the total samples $\{\mathbf{x}_o\}$ to each of these clusters. Each of the clusters can be then treated as an independent search space, including a potential local minimum. A SVM can be built for each of the clusters to facilitate KDE-MBK as discussed above, and the KDE results will be more accurate within each cluster. For identification of the global minimum information from all clusters should be combined before performing identification (8). In this study the k-means clustering algorithm [20] is used for this purpose and a gap-statistic approach [21] is adopted for selecting the number of clusters.

5 Adaptive stochastic sampling

The computationally most demanding component of NP-SSO is the simulation of samples from $\pi_I(\mathbf{x}, \boldsymbol{\theta})$ within subset I for $\mathbf{x}$, with this subset corresponding to either the initial domain X (for the 1st iteration of the algorithm) or the optimal subset I^* identified in the previous iteration (in case multiple clusters exist, this will correspond to the search domain of a particular cluster). Improvement of the efficiency of this stochastic sampling can thus greatly improve the overall efficiency of NP-SSO. For this purpose an adaptive kernel sampling density (AKSD) [22] is integrated within NP-SSO, utilizing samples from $\pi_I(\mathbf{x}, \boldsymbol{\theta})$ available from the previous iteration.

5.1 Rejection sampling and proposal density formulation

The specific sampling algorithm chosen to perform the stochastic sampling and provide independent samples (as required for the KDE) is rejection sampling [18], which is summarized as follows: first choose an appropriate proposal PDF $q(\boldsymbol{\theta})$ for the model parameters (this selection will be the focus here) and generate samples by:

Step 1: Simulate candidate sample $\{\mathbf{x}^q, \boldsymbol{\theta}^q\}$ with $\mathbf{x}^q \sim p(\mathbf{x})$ and $\boldsymbol{\theta}^q \sim q(\boldsymbol{\theta})$, and u from uniform $(0,1)$.

Step 2: Accept $\{\mathbf{x}, \boldsymbol{\theta}\} = \{\mathbf{x}^q, \boldsymbol{\theta}^q\}$ if

$$\frac{h(\mathbf{x}^q, \boldsymbol{\theta}^q)p(\boldsymbol{\theta}^q)}{M_{ar} \cdot q(\boldsymbol{\theta}^q)} > u \quad ; \quad \text{where } M_{ar} > \max_{\mathbf{x} \in I, \boldsymbol{\theta} \in \Theta} \left[\frac{h(\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \right] \quad (17)$$

Step 3: Return to Step 1 otherwise.

This process is repeated until the required number n_s of independent samples $\{\mathbf{x}_h, \boldsymbol{\theta}_h\}$ is obtained. As discussed earlier this provides evaluation of the system response $h(\mathbf{x}, \boldsymbol{\theta})$ for the set $\{\mathbf{x}_c, \boldsymbol{\theta}_c\} = \{\mathbf{x}_c^j, \boldsymbol{\theta}_c^j; j=1, \dots, N\}$ composed of all candidate samples $\{\mathbf{x}^q, \boldsymbol{\theta}^q\}$.

The computational efficiency of this sampling algorithms, expressed as the reciprocal of the number of trials needed (on average) to obtain one sample n_s/N , is equal to $E_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_I / M_{ar}$ [18], meaning that smaller values for M_{ar} lead to greater efficiency. This efficiency depends on how closely the proposal PDF $q(\boldsymbol{\theta})$ resembles the marginal target density $\pi_I(\boldsymbol{\theta})$. In the first iteration of NP-SSO, $q(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$, since no information is available. In subsequent iterations the samples from $\pi_I(\boldsymbol{\theta})$ from the previous iterations can be used to select better proposal densities $q(\boldsymbol{\theta})$. A similar concept could have been applied for $\mathbf{x}$, i.e. introduce a proposal PDF $q(\mathbf{x})$ for it as well. But we already know that as NP-SSO evolves regions with small sensitivity are identified, for which $\pi_I(\mathbf{x})$ is efficiently approximated by a uniform PDF $p(\mathbf{x})$. As such the introduction of proposal densities for $\mathbf{x}$ is not considered here.

The focus is, then, on development of an adaptive approach to choose $q(\boldsymbol{\theta})$ by sharing information across the iterations of NP-SSO. To formalize this process, consider that we have available samples from $\pi_I(\mathbf{x}, \boldsymbol{\theta})$ from the previous iteration; these are simply the samples within $\{\mathbf{x}_h, \boldsymbol{\theta}_h\} = \{\mathbf{x}_h^j, \boldsymbol{\theta}_h^j; j=1, \dots, n_s\}$ whose $\mathbf{x}$ component belongs in the new search space, and ultimately correspond to samples from the optimal proposal density [23]. We also have samples $\{\mathbf{x}_e, \boldsymbol{\theta}_e\} = \{\mathbf{x}_e^j, \boldsymbol{\theta}_e^j; j=1, \dots, n_e\}$ for which the performance measure $h(\mathbf{x}, \boldsymbol{\theta})$ is evaluated; similarly, these are samples within $\{\mathbf{x}_c, \boldsymbol{\theta}_c\} = \{\mathbf{x}_c^j, \boldsymbol{\theta}_c^j; j=1, \dots, N\}$ whose $\mathbf{x}$ component belongs in the new search space. Note that $n_e = N \cdot n_p / n_s$ since distribution used for $\mathbf{x}$ is uniform. Based on set $\{\boldsymbol{\theta}_p\}$, an adaptive proposal density formulation is developed through kernel sampling density (KSD), utilizing also the information provided by set $\{\mathbf{x}_e, \boldsymbol{\theta}_e\}$. Rather than trying to establish efficient densities for the entire $\boldsymbol{\theta}$ vector, which is challenging in high-dimensional problems [11], the focus is on the important components of $\boldsymbol{\theta}$, the ones that have higher impact on the system behavior. The sensitivity analysis developed recently by Jia and Taflanidis [24] is adopted here to facilitate this task.

This sensitivity analysis is based on comparison of the marginal distributions $\pi_I(\theta_i)$ and $p(\theta_i)$ through calculation of their relative information entropy, given by the scalar integral

$$D_{re}(\pi_I(\theta_i) \parallel p(\theta_i)) = \int \pi_I(\theta_i) \log \left[\frac{\pi_I(\theta_i)}{p(\theta_i)} \right] d\theta_i \quad (18)$$

Higher values for this entropy correspond to bigger discrepancies between $\pi_i(\theta_i)$ and $p(\theta_i)$ and ultimately bigger importance towards the probabilistic performance. For efficiently calculating this entropy, based on the readily available samples from $\pi_i(\theta_i)$, from $\{\theta_p\}$, an estimate $\tilde{\pi}_i(\theta_i)$ for $\pi_i(\theta_i)$ can be established through KDE and used for estimation of the scalar integral in (18) through standard numerical integration [24].

Formulation of KSD proposal densities is then considered only for the most influential ones, denoted $\mathbf{v}$ herein, with dimension n_v (the parameters having entropy values above a certain threshold –to be determined next). For the remaining components of θ , denoted $\mathbf{y}$ (so that $\theta=[\mathbf{v} \ \mathbf{y}]$), the proposal density will be selected equal to the initial PDF $q(\mathbf{y}) = p(\mathbf{y} | \mathbf{v})$. As such $q(\theta) = p(\mathbf{y} | \mathbf{v})q(\mathbf{v})$ and

$$\frac{p(\theta)}{q(\theta)} = \frac{p(\mathbf{y} | \mathbf{v})p(\mathbf{v})}{p(\mathbf{y} | \mathbf{v})q(\mathbf{v})} = \frac{p(\mathbf{v})}{q(\mathbf{v})} \quad (19)$$

5.2 Adaptive selection of kernel characteristics

Let $\{\mathbf{v}_p\}$ and $\{\mathbf{v}_e\}$ denote the components of the $\{\theta_p\}$ and $\{\theta_e\}$ sets corresponding to $\mathbf{v}$. The KSD based on the available samples $\{\mathbf{v}_p^j; j=1, \dots, n_p\}$ is developed using a multivariate adaptive product Kernel [25] as

$$q(\mathbf{v} | \mathbf{d}) = \frac{1}{n_p} \sum_{j=1}^{n_p} \left(\prod_{i=1}^{n_v} \left\{ \frac{1}{\lambda^j t_i} \left[K_G \left(\frac{v_i - v_{pi}^j}{\lambda^j t_i} \right) \right] \right\} \right) \quad (20)$$

where K_G is the chosen kernel, $\mathbf{d}$ is the vector of kernel-characteristics (defined later), λ^j is a local bandwidth factor, t_i is the fixed-window bandwidth parameter defining the spread of the kernel for the i^{th} variable v_i (i^{th} component of $\mathbf{v}$). For λ^j equal to 1 (or to any constant) the KSD in (20) corresponds to a fixed-bandwidth kernel. The introduction of λ^j establishes the adaptive kernel introduced by [25], aimed at improving the approximation at the tails of the distribution by using an adaptive bandwidth, selected larger for the regions with lower values for the probability density. This factor is given by

$$\lambda^j = \left\{ \left[\prod_{s=1}^{n_p} f_{pilot}(\mathbf{v}^s) \right]^{1/n_p} / f_{pilot}(\mathbf{v}^j) \right\}^\alpha = (\omega^j)^\alpha \quad (21)$$

where $0 < \alpha < 1$ is a sensitivity factor, ω^j is defined as

$$\omega^j = \left[\prod_{s=1}^{n_p} f_{pilot}(\mathbf{v}^s) \right]^{1/n_p} / f_{pilot}(\mathbf{v}^j) \quad (22)$$

and f_{pilot} is a pilot estimate for the KSD established using the fix-bandwidth kernel approximation, which is given by (20) but with $\lambda^j = 1$. The bandwidth for establishing

f_{pilot} has small impact on the final adaptive kernel [25], and common values suggested in the literature can be used for it. Such a bandwidth choice is appropriate for formulating the pilot estimate, but may not be necessarily appropriate for $q(\mathbf{v}|\mathbf{d})$ when utilized as proposal density.

The vector $\mathbf{d}$ for the kernel characteristics within this approach includes the sensitivity factor α and the bandwidth parameter t_i for each component of $\mathbf{v}$. The remaining, and very important step for the efficiency of the AKSD, is the adaptive selection of $\mathbf{d}$. Since AKSD is intended to be used as proposal density for efficient stochastic sampling, an explicit optimization is introduced here for $\mathbf{d}$, rather than relying on common values suggested in the literature (the pilot estimate utilizes such values). Because as discussed previously this efficiency is inversely proportional to M_{ar} the goal is to select $\mathbf{d}$ so that M_{ar} is minimized. Since M_{ar} is not known exactly, an approximation for it is established utilizing the information (evaluations of performance measure) available within set $\{\mathbf{x}_e, \boldsymbol{\theta}_e\}$. This leads to the optimization problem

$$\mathbf{d}^* = \arg \min_{\{t_i\} \in \mathbb{R}^{n_v}, 0 \leq \alpha \leq 1} \left(\max_{j=1, \dots, n_e} \left[h(\mathbf{x}_e^j, \boldsymbol{\theta}_e^j) \frac{p(\mathbf{v}_e^j)}{q(\mathbf{v}_e^j | \mathbf{d})} \right] \right) \quad (23)$$

where the quantity in the brackets is an approximation for M_{ar} based on readily available information. Note that the evaluation of the objective function in this problem simply requires estimation of $\{q(\mathbf{v}_e^j | \mathbf{d}); j=1, \dots, n_e\}$ for the new values of vector $\mathbf{d}$, something that involves minimal computational burden. As such, obtaining a local optimum for $\mathbf{d}^*$ should be considered as a straightforward numerical optimization.

5.3 Selection of number of model parameters for $\mathbf{v}$

The remaining question is, then, the selection of the exact number of model parameters corresponding to the vector $\mathbf{v}$. Rather than preselecting this number an adaptive selection can be further established to yield the best anticipated efficiency, using the sensitivity analysis provide only a relative prioritization [22]. The adaptive selection is established as follows (also summarized in Figure 4).

Step 1: The model parameters are first ordered based on their relative importance, i.e. the respective relative entropy values, estimated through (18). Based on some threshold D_{min} the maximum number of model parameters n_{max} for which KSD will be considered is then defined. Parameters having $D_{re}(\cdot)$ lower than D_{min} are ignored as potential candidates for the KSD formulation. D_{min} may be adaptively selected based on the highest entropy value among all parameters and some pre-defined allowable percentage reduction $s_e < 1$.

$$D_{min} = s_e \max_i \left[D_{re}(\pi_I(\theta_i) \| p(\theta_i)) \right] \quad (24)$$

Through this definition the selection is always normalized with respect to the importance of the most influential parameter.

Step 2: For different values of n_v up to n_{max} , we define $\mathbf{v}$ to consist of the n_v -most influential components of $\boldsymbol{\theta}$, and perform optimization (23) to find the

proportionality constant for the anticipated efficiency [inverse of quantity in the parenthesis in (23)] along with the corresponding optimal bandwidth and sensitivity factor.

Step 3: Finally, n_v is selected as the one yielding the best efficiency, corresponding to the smallest value for the quantity in the parenthesis in (23). This ultimately corresponds to the subset of $\boldsymbol{\theta}$ yielding the highest efficiency in formulating the AKSD in (20), and having explicitly optimized kernel characteristics.

6 Adaptive NP-SSO algorithm

6.1 Adaptive selection of number of samples for KDE-MBK

A critical last component of NP-SSO is the proper selection of the number of samples n_s used for the KDE approximation. This selection proportionally impacts the computational efficiency of the algorithm as it influences the total number of simulation required for the system response. It also impacts the accuracy of the KDE approximation and as such it needs to be carefully selected. Rather than preselecting n_s , as was established in [12], an adaptive multi-stage approach is developed here, based on accuracy criteria for the KDE approximation. An important question is then how should this accuracy be defined? Since the ultimate goal at each iteration of the NP-SSO is to identify the optimal subset I^* based on the KDE approximation, a proper criterion for selection of n_s is how does the identified subset changes for different selections of n_s . Thus, instead of looking at the impact of n_s on the KDE approximation accuracy, something that means focus over the entire search space I , its influence on the set identified by NP-SSO is utilized here. To facilitate this task a gradual, multi-stage increase in n_s is considered until the variation of the identified subset is below some chosen threshold.

This is ultimately established through the following approach. Select first the number of samples to increase/add at each stage, n_{sa} , the threshold ρ_δ defining the accuracy of the subset identification, the maximum number of allowable samples $n_{s,max}$, and generate n_u samples of $\mathbf{x}$ from the uniform distribution $p(\mathbf{x})$, denoted $\{\mathbf{x}_u\} = \{\mathbf{x}_u^j; j=1, \dots, n_u\}$, to use for all stages. Then choose adaptively n_s through the following procedure.

Step 1: Set counter $n_b=1$ and start the first stage of the selection. Simulate n_{sa} samples from $\pi_I(\mathbf{x}, \boldsymbol{\theta})$, $\{\mathbf{x}_h\} = \{\mathbf{x}_h^j; j=1, \dots, n_{sa}\}$, using the approach discussed in Section 5 based on information from the previous iteration.

Step 2: Using these samples establish approximation $\tilde{\pi}_I(\mathbf{x})$ through KDE and utilize it to calculate the objective function through (6) for all samples $\{\mathbf{x}_u\}$. Perform identification (8) and obtain set $\{\mathbf{x}_o\}_1$ composed of $n_o = \rho n_u$ samples within I^* .

Step 3: Set counter $n_b=n_b+1$ and proceed to the next stage. Simulate n_{sa} additional samples from $\pi_I(\mathbf{x}, \boldsymbol{\theta})$ using the approach discussed in Section 5 based on information from the previous iteration, to obtain a total $n_b n_{sa}$ samples, $\{\mathbf{x}_h\} = \{\mathbf{x}_h^j; j=1, \dots, n_b n_{sa}\}$, from $\pi_I(\mathbf{x}, \boldsymbol{\theta})$.

Step 4: Using the total number of samples establish approximation $\tilde{\pi}_l(\mathbf{x})$ through KDE and utilize it to calculate the objective function through (6) for all samples $\{\mathbf{x}_u\}$. Perform identification (8) and obtain set $\{\mathbf{x}_o\}_{n_b}$ composed of $n_o = \rho n_u$ samples within I^* .

Step 5: Count the number of samples that belong in $\{\mathbf{x}_o\}_{n_b}$ but do not belong in $\{\mathbf{x}_o\}_{n_{b-1}}$ (the sample set identifying I^* from the previous stage) and denote it by δn_o . This corresponds to the difference between the identified subsets using $n_s = n_b n_{sa}$ samples and $n_s = (n_b - 1) n_{sa}$ samples for the KDE. If the difference $\delta n_o / n_o$ is above the threshold selected for the accuracy of the identification ρ_δ and the total number of samples obtained $n_s = n_b n_{sa}$ is still smaller than the maximum number of allowable samples $n_{s,max}$ then go back to step 3.

Step 6: If $\delta n_o / n_o < \rho_\delta$ or $n_s = n_b n_{sa} > n_{s,max}$ then stop the multi-stage approach. Then the choice for n_s is finally $n_s = n_b n_{sa}$.

In the first iteration of the NP-SSO algorithm, and since no information is available to establish AKSD at stage one, i.e for $n_b=1$, (since no previous iteration exists) such densities can be formulated immediately before the second stage ($n_b=2$) utilizing the information from the first stage. This is not necessary for other iterations of NP-SSO since there is sufficient information from the previous iteration for the AKSD implementation.

6.2 Adaptive NP-SSO algorithm

Combining the previous concepts we establish the adaptive NP-SSO algorithm. Figure 5 illustrates some of the key steps of the algorithm, with the stochastic sampling details presented in Figure 4 earlier and the adaptive selection of n_s presented in Section 6.1.

Define first the design space X , the number of samples per stage for the multi-stage stochastic sampling, n_{sa} , the maximum number of samples, $n_{s,max}$, the accuracy threshold for the subset identification ρ_δ , the number of samples for characterization of the subsets n_u , the stopping threshold c^s , and the values for the volume reduction ρ . For the stochastic sampling define the percentage reduction s_e to determine the maximum value for n_v for which the formulation of AKSD will be considered.

At iteration k of the algorithm, suppose $I_k^m, m=1, \dots, M_{k-1}$ subsets (clusters) have been identified from the previous iteration (where for the first iteration only one cluster exists $I_1 = X$) with each one having volume $V_{I_k^m}$ for a total volume of $V_{I_k} = \sum_{m=1}^{M_{k-1}} V_{I_k^m}$. The subscript k is used herein to denote the iteration of the algorithm, and the superscript m to denote the specific subset (cluster). Note that the search space corresponds to the union of all clusters $I_k = \bigcup_{m=1}^{M_{k-1}} I_k^m$, whereas since the distribution for the design variables is uniform the probability of belonging to each cluster is $V_{I_k^m} / V_{I_k}$. Also a number of samples from $\pi_{I_k^m}(\mathbf{x}, \boldsymbol{\theta})$, $\{\mathbf{x}_{p(k-1)}^m, \boldsymbol{\theta}_{p(k-1)}^m\} = \{\mathbf{x}_{p(k-1)}^{mj}, \boldsymbol{\theta}_{p(k-1)}^{mj}; j=1, \dots, n_{p(k-1)}^m\}$, as well as samples $\{\mathbf{x}_{e(k-1)}^m, \boldsymbol{\theta}_{e(k-1)}^m\} = \{\mathbf{x}_{e(k-1)}^{mj}, \boldsymbol{\theta}_{e(k-1)}^{mj}; j=1, \dots, n_{e(k-1)}^m\}$ for which $h(\mathbf{x}, \boldsymbol{\theta})$ is known, will be readily available from the previous iteration of the algorithm (this will not be true of course in the first iteration). The k^{th} iteration of NP-SSO requires the following steps.

Step 1: Simulate $n_s - n_{p(k-1)}^m$ additional samples from $\pi_{I_k^m}(\mathbf{x}, \boldsymbol{\theta})$ using the approach discussed in Section 5 (presented also in Figure 4) to develop AKSD, while n_s is

adaptively selected using the approach discussed in Section 6.1. Use the sample set $\{\boldsymbol{\theta}_{p(k-1)}^m\}$ to formulate the kernel density and the set $\{\mathbf{x}_{e(k-1)}^m, \boldsymbol{\theta}_{e(k-1)}^m\}$ to optimally select the characteristics of the kernel as well as the number of model parameters n_{vk}^m defining vector $\mathbf{v}_k^m$. A total of n_s samples from $\pi_{I_k^m}(\mathbf{x}, \boldsymbol{\theta})$, $\{\mathbf{x}_{hk}^m, \boldsymbol{\theta}_{hk}^m\} = \{\mathbf{x}_{hk}^{mj}, \boldsymbol{\theta}_{hk}^{mj}; j=1, \dots, n_s\}$ will be available, and the projection onto the design variable space gives samples $\{\mathbf{x}_{hk}^m\} = \{\mathbf{x}_{hk}^{mj}, j=1, \dots, n_s\}$ from $\pi_{I_k^m}(\mathbf{x})$. Also do maintain the information for all trials $\{\mathbf{x}_{ck}^m, \boldsymbol{\theta}_{ck}^m\}$ within the sampling process.

Step 2: Using all samples $\{\mathbf{x}_{hk}^m\}$ from $\pi_{I_k^m}(\mathbf{x})$, estimate $\tilde{\pi}_{I_k^m}(\mathbf{x})$ through (12) and using all trials $\{\mathbf{x}_{ck}^m, \boldsymbol{\theta}_{ck}^m\}$ available from the stochastic sampling algorithm from Step 1, estimate $\hat{E}_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_{I_k^m}$ through (13). Establish approximation for the pair $\mathbf{x} - \tilde{C}(\mathbf{x})$ through (6) for $\text{Int}(n_u V_{I_k^m} / V_{I_k})$ number of samples of the design variables uniformly distributed in I_k^m , denoted $\{\mathbf{x}_{uk}^m\} = \{\mathbf{x}_{uk}^{mj}; j=1, \dots, \text{Int}(n_u V_{I_k^m} / V_{I_k})\}$, where Int denotes the integer function. The SVM established in the previous iteration is used in this step to estimate the coefficients in (14), needed for (12).

Step 3: After performing steps one and two for each cluster, combine all samples $\{\mathbf{x}_{uk}\} = \bigcup_{m=1}^{M_{k-1}} \{\mathbf{x}_{uk}^m\}$ (will end up with a total of n_u samples uniformly distributed in I_k), order them based on the values of $\tilde{C}(\mathbf{x})$, adaptively select c_k^h as the value corresponding to the lower ρ -percentile and keep only the samples for $\mathbf{x}$ that correspond to the lower values of $\tilde{C}(\mathbf{x})$. These samples $\{\mathbf{x}_{ok}\}$ are then uniformly distributed in the new optimal set I_k^* .

Step 4: Using (11) calculate $\hat{H}(I_k^* | I_k)$. If $\hat{H}(I_k^* | I_k) < c^s$ proceed to Step 5, else stop as the algorithm has converged.

Step 5: Separate the samples $\{\mathbf{x}_{ok}\}$ into the clusters identified at the beginning of the iteration and use any appropriate clustering algorithm to identify new clusters, if they exist, within each subset. This ultimately separates $\{\mathbf{x}_{ok}\}$ into M_k total clusters, $\{\mathbf{x}_{ok}^m\}, m=1, \dots, M_k$.

Step 6: For each of the clusters, build a SVM through the process outlined in Section 4.1 to characterize the respective subset and classify points in the next iteration. The new search region is then $I_k^* = \bigcup_{m=1}^{M_k} I_k^{*m}$.

Step 7: For the n_s samples from $\pi_{I_k^m}(\mathbf{x}, \boldsymbol{\theta})$ generated in Step 1, keep those samples whose $\mathbf{x}$ component belongs in I_k^{*m} . These samples define set $\{\mathbf{x}_{pk}^m, \boldsymbol{\theta}_{pk}^m\}$. Similarly the samples $\{\mathbf{x}_{ck}^m, \boldsymbol{\theta}_{ck}^m\}$ whose $\mathbf{x}$ component belongs in I_k^{*m} define $\{\mathbf{x}_{ek}^m, \boldsymbol{\theta}_{ek}^m\}$. Set finally $I_{k+1}^m = I_k^{*m}$ and proceed to iteration $k+1$ of the NP-SSO algorithm, starting with Step 1.

Once the convergence criteria have been satisfied then an approximation for the optimal design choice $\mathbf{x}^*$ can be established as the value of $\mathbf{x}$ corresponding to the minimum of $\tilde{C}(\mathbf{x})$

$$\mathbf{x}_{NP-SSO}^* = \operatorname{argmin}_{\mathbf{x} \in I_k^*} \tilde{C}(\mathbf{x}) \quad (25)$$

Note that full system response evaluations are required in the proposed algorithm only in Step 1, when additional samples need to be generated. Thus, NP-SSO, explores the sensitivity of the objective function in a highly efficient manner, whereas the AKSD and adaptive selection of n_s established here greatly enhance its numerical efficiency by reducing the computational burden required within the stochastic sampling stage. The ability to consider complex design domains through the adoption of KDE-MBK and SVM further contributes to this efficiency improvement.

7 Illustrative example

As an illustrative example, the design of the isolation characteristics for a three-story base-isolated structure (Figure 6) for protection against near-fault earthquake excitations is considered. The performance of the system is analyzed through nonlinear time-history analysis incorporating complex models for describing the structural behavior as well as the seismic hazard. The characteristics of the probability and structural models are the same as the example considered in [12]. Here simply the objective function is modified to correspond to a general optimization under uncertainty problem (rather than a reliability based optimization as in [12]).

7.1 Numerical models and performance quantification

The superstructure is modeled as a planar linear shear building with uncertain inter-story stiffness, incorporating correlation between the different floors, and uncertain classical modal damping. More information can be found in [12]. The base-isolation protective system is composed of lead-rubber bilinear isolators and supplemental viscous dampers at the base level. The pre-yield stiffness, K_{pr} , and post-yield stiffness, K_p , and the yield force, F_y , are the design variables $\mathbf{x}$ for the isolation system along with the damping coefficient, c_d , for the supplemental viscous dampers. A simplified problem with only two design variables is also formulated by setting the post-yield stiffness equal to 15% of the pre-yield stiffness and the viscous dampers to 5% critical damping assuming a nominal period of 2.25 sec for the isolated structure (thus only K_{pr} and F_y are design variables in this case). The simplified problem is denoted by D_1 and the full one by D_2 .

The seismic hazard at the structural site is quantified through the acceleration time history of near-fault earthquakes following the modeling approach discussed in detail in [26] and including as seismological parameters (and the main sources of uncertainty for the excitation) the moment magnitude M and the rupture distance r_{rup} . A point-source model [27] is adopted for characterizing the broad-band component of the excitation, relying on modulation of a white noise-sequence $\mathbf{Z}_w$ through functions dependent upon M and r_{rup} . This broadband component is coupled with a long-period, pulse component [28] representing the near-fault characteristics of ground motions. This includes as additional model parameters the amplitude, A_p , frequency, f_p , number of half-cycles γ_p and phase, ν_p for the pulse. The probability models for these parameters are selected based on predictive

relationships available in the literature [29] that relate them to the seismological parameters. Further details on this excitation model can be found in [12].

For quantifying the performance of the isolated system, the fragility of three different response quantities is considered: (i) the maximum inter-story drifts $d_m(\mathbf{x}, \boldsymbol{\theta})$, (ii) the maximum base displacement $b_m(\mathbf{x}, \boldsymbol{\theta})$ and (iii) the maximum shear force at the first story $F_m(\mathbf{x}, \boldsymbol{\theta})$. The fragility function for each is taken as a lognormal distribution with some chosen median and coefficient of variation and the overall system performance measure is taken as the average of the three fragilities

$$h(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{3} \left[\Phi_g \left(\frac{\ln(d_m / \mu_d)}{\beta_d} \right) + \Phi_g \left(\frac{\ln(b_m / \mu_b)}{\beta_b} \right) + \Phi_g \left(\frac{\ln(F_m / \mu_F)}{\beta_F} \right) \right] \quad (26)$$

where $\Phi_g(\cdot)$ is the standard Gaussian cumulative distribution function. The characteristics for the median of the fragility curves are $\mu_d=0.0105\text{m}$, $\mu_b=0.25\text{m}$ and μ_F equal to $0.24g$ of structural mass, whereas the coefficient of variation is set for all of them equal to 20%.

The uncertain parameter vector in this design problem then consists of the structural model parameters $\boldsymbol{\theta}_s$, the seismological parameter $\boldsymbol{\theta}_g=[M, r_{rup}]$, the additional parameters for the near-fault pulse $\boldsymbol{\theta}_p=[A_p, f_p, \gamma_p, v_p]$ and the white-noise sequence, $\mathbf{Z}_w$ (in other words a new ground motion time-history is created for each considered sample), so $\boldsymbol{\theta}=[\boldsymbol{\theta}_s, \boldsymbol{\theta}_g, \boldsymbol{\theta}_p, \mathbf{Z}_w]$. Note that $\mathbf{Z}_w$ corresponds to a 5001-dimensional vector in this case (partitioning of a 50 sec time window into intervals of 0.01sec). The samples for all these variables within NP-SSO are simulated within the standard Gaussian space. This is chosen to avoid boundary correction problems within the AKSD formulation (a Gaussian Kernel is chosen, as discussed previously). The initial design interval for each design variable is defined as [25 150]MN/m for K_{pr} , [1 5] MN for F_y , [0.1 5]MN/m for K_p and [0.1 10]MN·s/m for c_d .

7.2 Optimization results and discussions

The NP-SSO is implemented with choices number of uniform samples for characterization of subsets $n_u=10000$, volume reduction of $\rho=0.2$ and convergence criterion $c^s=0.97$. For the adaptive selection of n_s , n_{sa} is selected as 300 for D_1 and 600 for D_2 , with $n_{s,\max}=3000$ and 6000 respectively, while ρ_δ is selected as 0.2 for both problems. Larger values are chosen for the higher-dimensional problem D_2 as it is anticipated that the accuracy of KDE will decrease.

For problem D_1 NP-SSO converged in 2 iterations to an optimal solution, whereas for problem D_2 in 4 iterations. Table 1 presents the results per iteration for a sample run of NP-SSO as well as the optimal solutions provided by NP-SSO for both design problems. In particular it includes the ratio of average values $\hat{H}(I_k^* | I_k)$, the final size reduction per design variable $\sqrt{n_s} \sqrt{V_{I_{NP-SSO}}} / V_X$, the average value of the objective function over the search space $\hat{E}_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_{I_k}$, as well as the optimal solution $\mathbf{x}_{NP-SSO}^*$ if the algorithm was to stop at each iteration, along with the objective function at that optimal solution evaluated through stochastic simulation $\hat{C}(\mathbf{x}_{NP-SSO}^*)$ as in (3). The latter is calculated with high accuracy (corresponding to coefficient of variation below 5%) whereas the same samples are used in all evaluations to facilitate a more consistent comparison (reduction of relative influence of estimation error).

The results in the table verify the efficiency of NP-SSO. In a small number of iterations a subset with small sensitivity for the objective function [large value for $\hat{H}(I_k^* | I_k)$] is

identified corresponding to a significant average reduction per design variable (given in last row of Table 1). Through the proposed here MBK-based KDE, supported by SVM for computational efficiency, complex, non-parametric domains can be utilized as candidate search spaces and ultimately the pre-selected volume reduction is achieved (20% in this example) at each iteration regardless of the dimensionality of the problem. This was not feasible in the previously adopted KDE, utilizing reflection approaches for the boundary corrections, which was forced to approximate the search space as a box-bounded superset of the target domain, contributing to large increase of its volume (compared to the target volume-reduction) for high-dimensional problems [12].

As anticipated, $\hat{H}(I_k^* | I_k)$ exhibits an increasing trend as regions with lower sensitivity are gradually identified. The results also demonstrate the efficiency of the iterative NP-SSO approach for gradually converging to the optimal solution, rather than trying to pinpoint it in a single iteration (which would lead to sub-optimal results in both examples). This is evident by the fact that the quality of the optimal solution for each iteration, evaluated through $\hat{C}(\mathbf{x}_{NP-SSO}^*)$, improves per iteration. The small sensitivity within the identified set I_k^* upon convergence is also evident by the proximity between the average value $\hat{E}_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_{I_k}$ and the identified optimal solution $\hat{C}(\mathbf{x}_{NP-SSO}^*)$, whereas it should be also stressed that even though the optimal solution for problem D_2 is on the boundary for one of the design variables (F_y) NP-SSO successfully identifies that solution.

The quality of the solution obtained is further verified by solving the problem through an approach relying on stochastic simulation, utilizing (3) to calculate the objective function. An exterior sampling [19] approach is obtained with $n_r=3000$ to reduce the influence of the prediction error (coefficient of variation close to 3% was established using appropriate important sampling densities [11]), whereas a gradient-free optimization algorithm relying on surrogate modeling was adopted to address the high computational burden associated with each objective function evaluation (costly estimation). This is implemented through the TOMLAB optimization environment [30]. The solutions obtained correspond to objective function values of 0.0257 and 0.0192 for design problems D_1 and D_2 , respectively, which are similar to the ones identified through NP-SSO, and provide a further validation of the optimality of solution obtained. The number of iterations required for convergence to the optimal solution was 100 for D_1 and 200 for D_2 (these were the maximum allowable iterations numbers chosen for each problem) leading to a total of 300,000 and 600,000 evaluations of the objective function (note, though, that no special effort was given for reducing this burden).

7.3 Efficiency of adaptive sampling and adaptive selection of samples

The computational efficiency of the AKSD and of the adaptive selection of the number of samples n_s for the NP-SSO implementation is investigated in detail in this Section. First the benefits stemming from the adaptive selection of the number of dimensions n_v for formulating the AKSD are discussed. For this purpose the variation of sampling efficiency (percentage of accepted samples) with respect to n_v is shown in Figure 7 for a representative iteration of both design problems (D_1 and D_2). Similar trends hold for all other iterations examined. It is evident that formulating proposal densities for a large number of the model parameters is not beneficial as it encounters numerical challenges in selecting appropriate bandwidth characteristics. A distinct optimum exists for n_v (in this case equal to 3) for the AKSD implementation, whereas for values beyond that optimum, the efficiency quickly drops. The influential model parameters, and the optimal ones for

the AKSD, were found in this application to correspond consistently to the hazard characteristics: the moment magnitude, the rupture distance and the pulse amplitude.

The efficiency of the overall AKSD scheme is further demonstrated in Table 2, which presents the acceptance rate established through implementing three different sampling schemes for the two design problems. The first one, DRS, corresponds to direct implementation of rejection sampling, with no effort to formulate new proposal densities [in other words $q(\boldsymbol{\theta})$ is selected as $p(\boldsymbol{\theta})$]. The second one, AKSD, corresponds to the adaptive formulation discussed in Section 5, which entails relative prioritization of the model parameters and then an explicit optimization to adaptively select the kernel characteristics as well as the number of model parameters for the development of the sampling density, to obtain the best overall anticipated efficiency. Finally the third one, KSD, considers no explicit optimization for the kernel characteristics, but rather common values from the literature are adopted for bandwidth (the one minimizing the asymptotic mean integrated square error [14]) and sensitivity factor (0.5). The relative prioritization is still adopted for this scheme as well (since it was found to provide great benefits), with $\mathbf{v}$ selected the same as the AKSD case (corresponding to the hazard characteristics mentioned above).

As the iterations proceed, and regions with lower objective function values (see $\hat{E}_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_{I_k}$ in Table 1), and ultimately higher probabilistic sensitivity with respect to $\boldsymbol{\theta}$ are identified (note that the sensitivity with respect to $\mathbf{x}$ is decreased as these regions are close to local minima), the efficiency of DRS decreases, as expected. This is not true for AKSD, which maintains a high efficiency. It is interesting to note that even at the last iteration of NP-SSO, when algorithm is approaching regions with high sensitivity to $\boldsymbol{\theta}$ (as evident by the small efficiency of the DRS approach) the AKSD scheme provides high computational efficiency; this shows that the sharing of information across the iterations of NP-SSO is greatly beneficial. Note that for design problem D_2 multiple clusters are identified at iterations two and three of the algorithm and the numerical details are presented with respect to each of these clusters. The overall sampling efficiency for these iterations can be taken as the average efficiency over the individual clusters. The increasing trend for the ratio of sampling efficiency between AKSD and DRS clearly demonstrates the computational benefits established through the proposed AKSD scheme, with computational burden reduced in this case by a factor of up to 8 for D_2 . The comparison between AKSD and KSD is also very interesting: the explicit optimization of the bandwidth characteristics contributes to a considerable improvement of the stochastic sampling computational-efficiency. For problems with complex numerical models, where the computational burden for estimating the model performance is significantly higher than the one for optimization (23), this feature justifies the additional numerical cost for performing this optimization.

This overall discussion validates the efficiency of the proposed scheme for sharing information across the iterations of the algorithm to, (i) prioritize model parameters to focus on [problems when trying to formulate proposal densities for entire $\boldsymbol{\theta}$], (ii) establish good proposal densities [great improvements over DRS], (iii) properly select the kernel characteristics by performing an explicit optimization [advantages over the KSD case]. Since this AKSD formulation is established with small additional computational burden (using readily available information), it ultimately provides a great improvement in the numerical efficiency of NP-SSO.

Table 2 also shows details for the overall numerical burden for simulating the necessary samples. In particular it includes the number n_s needed (adaptive choice) per iteration to

establish the desired accuracy in the identification of the candidate optimal subset, the number of simulations of the model response needed to obtain these samples, as well as the total computational burden (total number of evaluation of the system response) till convergence of the algorithm. Note that for the first iteration the first n_{sa} samples are obtained without an AKSD implementation, which contributes to an increased computational burden. The number of samples required per iteration remains significantly below the maximum number of samples considered, something that contributes to a large improvement in computational efficiency compared to previous implementations of the NP-SSO algorithm [12, 15], where this number was pre-selected. Coupled with the sampling efficiency provided by the AKSD formulation this leads to a small overall computational burden, close to just 10000 simulations of the system model for problem D_1 and 50000 for problem D_2 . This corresponds to great computational savings compared to a traditional optimization approach relying on stochastic simulation; assuming, for example, that 3000 samples are used for the estimation within such a setting (this is a sufficient number of samples to establish a small coefficient of variation close to the optimum for this specific application), one would have to use only 4 and 16 objective function evaluations for problems D_1 and D_2 , respectively, to converge to the optimal solution with the same computational burden. The details of such an approach can, of course, greatly vary depending on the algorithm selected whereas adaptive choices could have been also made [31, 32]. The overall discussion, though, provides an important validation of the computational efficiency of the NP-SSO algorithm, stemming from the ability to explore globally the sensitivity of the objective function to the design variables, and further enhanced by the developed here AKSD formulation and adaptive selection of the necessary number of samples.

7.4 Different choices of volume reduction

The dependence of the algorithmic efficiency on the targeted volume reduction is finally explored in this Section, by considering a value of $\rho=0.1$. Results are presented in Tables 3 and 4 in a similar format as in Tables 1 and 2, respectively.

NP-SSO with $\rho=0.1$ successfully converges to the optimal solution, as evident by comparing the objective function values $\hat{C}(\mathbf{x}_{NP-SSO}^*)$ between Tables 1 and 3. There are some differences in some of the design variables $\mathbf{x}_{NP-SSO}^*$ but this simply demonstrates that there is a manifold with very small sensitivity (so larger differences in the design configuration contribute to very small differences in the objective function value). Compared to $\rho=0.2$ case the algorithm converges in fewer iterations, as expected, but requires a larger number of samples per iteration to establish the required accuracy. As mentioned earlier, the choice of ρ is a compromise between the total number of iterations required and the number of evaluations required within each iteration; smaller ρ means more confidence in the KDE estimation and therefore usually requires adaptively selecting larger n_s as shown by comparison of n_s for D_1 in Tables 2 and 4 (900 compared to 600), resulting to overall more evaluations of the system model for D_1 with choice of $\rho=0.1$. For D_2 better efficiency is established with the choice of $\rho=0.1$ (30077 total function evaluations compared to 48651 for $\rho=0.2$) but this should be attributed to the fact that by identifying a smaller subset, multiple clusters are avoided (this was not the case for $\rho=0.2$) which saves computational effort (focus is only on a single cluster).

In terms of the sampling efficiency even greater savings are established by the AKSD implementation whereas a trend of increasing efficiency of AKSD with the iterations of the NP-SSO is identified. This should be attributed to the fact that the formulation of

proposal densities has focused on θ and not $\mathbf{x}$, thus it is expected to benefit from convergence to regions where the sensitivity with respect to $\mathbf{x}$ is small. This discussion also shows that for some problems, formulations of proposal densities also with respect to $\mathbf{x}$ could be beneficial.

Overall the implementation for volume reduction of $\rho=0.1$ further validates the robustness and computational efficiency of the NP-SSO algorithm, whereas it does not provide a strong preference over the $\rho=0.2$ case. This trends holds for most of the cases the authors have considered, with values of $\rho=0.1-0.2$ providing similar efficiencies and sharing same robustness in converging to the optimal solution.

8 Conclusions

In this paper the NP-SSO algorithm was extended to general design under uncertainty problems (not constrained to adopting system reliability as performance objective), but more importantly it was enhanced through the following new computational advances.

a) A truly non-parametric characterization of the subsets identified at each iteration was established, facilitating a more efficient reduction of volume of the search space, especially as the dimension of the design variable increases. This was established through introduction of a multivariate boundary kernel (KDE-MBK) that can address complex (even geometrically non-parametric) domains/boundaries.

b) For characterizing the identified subsets and efficiently calculating all numerical quantities involved in the KDE-MBK in the next iteration of the algorithm, as well as facilitating the simulation of samples for the design variables, a support vector machine (SVM) was proposed.

c) For improving the sampling efficiency within NP-SSO, an adaptive kernel sampling density (AKSD) formulation was integrated in the algorithm. This was established by selecting the most important (probabilistically) model parameters to focus on, and then by choosing the bandwidth characteristics of the kernel and the exact number of parameters considered for the AKSD with an explicit objective to maximize the anticipated efficiency. This was performed utilizing readily available system evaluations, without increasing the overall computational burden.

d) An adaptive selection of the number of samples needed for the KDE approximation was proposed. This was established through a multi-stage process; a new set of samples is obtained at each stage and then a new KDE approximation is established and a new candidate subset is identified. If this subset does not differ significantly from the previous one (obtained utilizing a smaller number of samples) the KDE approximation has sufficient accuracy and the multi-stage sampling stops.

In the numerical example considered these advances were shown to contribute significantly to the efficiency and robustness of the algorithm. Overall it was also demonstrated that NP-SSO efficiently reduces the size of the search space and efficiently identifies the optimal solutions, whereas the established iterative approach significantly improves the quality of that solution. Also, the incorporation and optimal selection of AKSD, coupled with the adaptive selection of the number of samples needed for facilitating the KDE approximation, significantly increased the stochastic sampling efficiency and ultimately the efficiency of NP-SSO in each iteration.

Appendix A

For KDE-MBK estimation of the density at a point $\mathbf{x}$, the correction coefficients defined by the integrals in (14), can be estimated through Monte Carlo (MC) integration using uniform samples in the integration domain $\Delta_I(\mathbf{x})$. Facilitated by SVM, the estimation can be established through the following steps.

Step 1: Generate n_c uniform samples $\{\xi\} = \{\xi^j; j=1, \dots, n_c\}$, with ξ^j denoting the j^{th} sample, in the support region $S(\mathbf{x})$ for $\mathbf{x}$, which for the Epanechnikov Kernel is a hyper-rectangle centred at $\mathbf{x}$ and with width $2w_i$ in the i^{th} dimension. Note that the total volume of $S(\mathbf{x})$ is $V_c = 2^{n_x} \prod_{i=1}^{n_x} w_i$.

Step 2: Use SVM to classify the n_c samples $\{\xi\}$ and identify the samples that are inside $\Delta_I(\mathbf{x})$, denoted $\{\xi_o\} = \{\xi_o^j; j=1, \dots, n_{co}\}$, corresponding to uniform samples in $\Delta_I(\mathbf{x})$. The volume for the latter is $V_o = V_c n_{co} / n_c$.

Step 3: Utilizing set $\{\xi_o\}$ use MC integration to estimate the correction coefficients as

$$\begin{aligned} c_0(\mathbf{x}) &\approx V_o / n_{co} \sum_{j=1}^{n_{co}} K_m(\mathbf{x} - \xi^j) \\ c_{1i}(\mathbf{x}) &\approx V_o / n_{co} \sum_{j=1}^{n_{co}} K_m(\mathbf{x} - \xi^j) (x_i - \xi_i^j) \\ d_{kl}(\mathbf{x}) &\approx V_o / n_{co} \sum_{j=1}^{n_{co}} K_m(\mathbf{x} - \xi^j) (x_k - \xi_k^j) (x_l - \xi_l^j) \end{aligned} \quad (27)$$

Acknowledgments

This research effort is supported by the National Science Foundation (NSF) under Grant No. CBET-1235768. This support is gratefully acknowledged.

References

- [1] Beyer H-G, Sendhoff B. Robust optimization -A comprehensive survey. Computer Methods in Applied Mechanics and Engineering 2007; 196 (3190-3218).
- [2] Beck AT, Santana Gomes WJ. A comparison of deterministic, reliability-based and risk-based structural optimization under uncertainty. Probabilistic Engineering Mechanics 2012; 12: 18-29.
- [3] Doltsinis I. Robust design of structures using optimization methods. Computer Methods in Applied Mechanics and Engineering 2004; 193 (23-26): 2221-2237.
- [4] Jaynes ET. Probability Theory: The logic of science. Cambridge University Press, Cambridge, UK; 2003.
- [5] Royset JO, Der Kiureghian A, Polak E. Optimal design with probabilistic objective and constraints. Journal of Engineering Mechanics 2006; 132 (1): 107-118.
- [6] Aoues Y, Chateauneuf A. Reliability-based optimization of structural systems by adaptive target safety - Application to RC frames. Structural Safety 2008; 30 (2): 144-161.

- [7] Taflanidis AA, Beck JL. Life-cycle cost optimal design of passive dissipative devices. *Structural Safety* 2009; 31 (6): 508-522.
- [8] Valdebenito MA, Schuëller GI. A survey on approaches for reliability-based optimization. *Structural and Multidisciplinary Optimization* 2010; 42 (5): 645-663.
- [9] Schuëller GI, Jensen HA. Computational methods in optimization considering uncertainties - An overview. *Computer Methods in Applied Mechanics and Engineering* 2008; 198 (1): 2-13.
- [10] Royset JO, Polak E. Reliability-based optimal design using sample average approximations. *Probabilistic Engineering Mechanics* 2004; 19: 331-343.
- [11] Taflanidis AA, Beck JL. An efficient framework for optimal robust stochastic system design using stochastic simulation. *Computer Methods in Applied Mechanics and Engineering* 2008; 198 (1): 88-101.
- [12] Jia G, Taflanidis AA. Non-parametric stochastic subset optimization for optimal-reliability design problems. *Computers and Structures* 2013; 126: 86-99.
- [13] Taflanidis AA, Beck JL. Stochastic subset optimization for optimal reliability problems. *Probabilistic Engineering Mechanics* 2008; 23 (2-3): 324-338.
- [14] Silverman BW. *Density Estimation for Statistics and Data Analysis*. Boca Raton, Florida; 1998.
- [15] G. Jia, A. Taflanidis, Non-Parametric Stochastic Subset Optimization for System Design Optimization under Uncertainty, in: Y. Tsompanakis (Ed.) *Proceedings of the Third International Conference on Soft Computing Technologies in Civil, Structural and Environmental Engineering*, Civil-Comp Press, Stirlingshire, UK, Paper 10, 2013.
- [16] Neilsen J. Multivariate boundary kernels from local linear estimation. *Scandinavian Actuarial Journal* 1999; 1999 (1): 93-95.
- [17] Schölkopf B, Smola AJ. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. The MIT Press, Cambridge, Massachusetts; 2002.
- [18] Robert CP, Casella G. *Monte Carlo statistical methods*. 2nd Edition ed., Springer, New York, NY; 2004.
- [19] Spall JC. *Introduction to stochastic search and optimization*. Wiley-Interscience, New York; 2003.
- [20] Hartigan JA, Wong MA. Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 1979; 28 (1): 100-108.
- [21] Tibshirani R, Walther G, Hastie T. Estimating the number of clusters in a data set via the gap statistic. *Journal of Royal Statistical Society: Series B* 2000; 63 (2): 411-423.
- [22] Jia G, Taflanidis A, Beck JL, "Adaptive stochastic sampling using kernel density approximations," in *7th International Conference on Computational Stochastic Mechanics* Santorini, Greece, 2014.
- [23] Au SK, Beck JL. A new adaptive importance sampling scheme. *Structural Safety* 1999; 21 (2): 135-158.
- [24] Jia G, Taflanidis AA. Sample-based evaluation of global probabilistic sensitivity measures *Computers and Structures* 2014; 144: 103-118.
- [25] Abramson IS. On bandwidth variation in kernel estimates - a square root law. *Annals of Statistics* 1982; 10 (4): 1217-1223.

- 1 [26] Taflanidis AA, Jia G. A simulation-based framework for risk assessment and
2 probabilistic sensitivity analysis of base-isolated structures. *Earthquake*
3 *Engineering & Structural Dynamics* 2011; 40: 1629–1651.
- 4 [27] Boore DM. Simulation of ground motion using the stochastic method. *Pure and*
5 *Applied Geophysics* 2003; 160: 635-676.
- 6 [28] Mavroeidis GP, Papageorgiou AS. A mathematical representation of near-fault
7 ground motions. *Bulletin of the Seismological Society of America* 2003; 93 (3):
8 1099-1131.
- 9 [29] Bray JD, Rodriguez-Marek A. Characterization of forward-directivity ground
10 motions in the near-fault region. *Soil Dynamics and Earthquake Engineering* 2004;
11 24: 815-828.
- 12 [30] Holmstrom K, Goran AO, Edvall MM. User's guide for TOMLAB 7. Tomlab
13 Optimization Inc. www.tomopt.com/tomlab/, San Diego, CA; 2009.
- 14 [31] Polak E, Royset JO. Efficient sample size in stochastic nonlinear programming.
15 *Journal of Computational and Applied Mathematics* 2007; 217 (2): 301-310.
- 16 [32] Medina JC, Taflanidis A. Adaptive importance sampling for optimization under
17 uncertainty problems. *Computer Methods in Applied Mechanics and Engineering*
18 2014; 279: 133–162.
- 19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Figure Captions

Figure 1: Illustration of sample-based approximation concepts for NP-SSO

Figure 2: Illustration of KDE approximation for complex boundaries

Figure 3: Identification of subsets by KDE-MBK and characterization by SVM

Figure 4: Illustration of building optimal AKSD within NP-SSO

Figure 5: Illustration of the adaptive NP-SSO algorithm

Figure 6: Structural model and a sample near-field excitation

Figure 7: Variation of sampling efficiency against the dimension (n_v) of the vector of model parameters considered for building AKSD

Table Captions

Table 1: Results per iteration from a sample run of NP-SSO ($\rho = 0.2$)

Table 2: Sampling efficiency of adaptive kernel sampling density ($\rho = 0.2$)

Table 3: Results per iteration from a sample run of NP-SSO ($\rho = 0.1$)

Table 4: Sampling efficiency of adaptive kernel sampling density ($\rho = 0.1$)

Nomenclature

AKSD	adaptive kernel sampling density	KDE	kernel density estimation
$\mathbf{x}$	design variable vector	$\mathbf{x}^*$	optimal design solution
x_i	i^{th} design variable	$\boldsymbol{\theta}$	Uncertain model parameters
n_x	dimension of $\mathbf{x}$	n_θ	dimension of $\boldsymbol{\theta}$
X	admissible design space	Θ	space of possible values for $\boldsymbol{\theta}$
$h(\mathbf{x}, \boldsymbol{\theta})$	performance measure	$p(\cdot)$	Probability density function
$C(\mathbf{x})$	objective function	$\hat{C}(\mathbf{x})$	approximation to objective function through stochastic simulation
I	bounded search space	V_I	volume of set I
$E_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_I$	Average value for objective function for $\mathbf{x}$ belonging in I	$\tilde{C}(\mathbf{x})$	approximation to objective function through NP-SSO
$\hat{E}_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_I$	Approximation to $E_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_I$ through stochastic simulation	$\pi_i(\mathbf{x})$	Auxiliary density function for $\mathbf{x}$ with $\mathbf{x}$ belonging in I
$\pi_i(\mathbf{x}, \boldsymbol{\theta})$	Auxiliary density function for $\mathbf{x}$ and $\boldsymbol{\theta}$ with $\mathbf{x}$ belonging in I	$\pi_i(\boldsymbol{\theta})$	Auxiliary density function for $\boldsymbol{\theta}$ with $\mathbf{x}$ belonging in I
$\{\mathbf{x}_\cdot^j, \boldsymbol{\theta}_\cdot^j\}$	j^{th} sample from $\mathbf{x}_\cdot$ or $\boldsymbol{\theta}_\cdot$	n_s	Number of samples available from $\pi_i(\mathbf{x})$
$\{\mathbf{x}_h, \boldsymbol{\theta}_h\}$	sample set for $\{\mathbf{x}, \boldsymbol{\theta}\}$ from $\pi_i(\mathbf{x}, \boldsymbol{\theta})$	$\{\mathbf{x}_h\}$	sample set from $\pi_i(\mathbf{x})$
$\{\mathbf{x}_c, \boldsymbol{\theta}_c\}$	sample set for which the system response is evaluated during the stochastic sampling process	N	number of samples in set $\{\mathbf{x}_c, \boldsymbol{\theta}_c\}$
$\underline{K}_m$	Multivariate kernel with boundary correction	K_m	Product multivariate kernel
$K(\cdot)$	univariate kernel for approximation of $\pi_i(\mathbf{x})$	$q(\cdot)$	proposal density for $\boldsymbol{\theta}$ for obtaining samples $\{\boldsymbol{\theta}_c\}$
c_0, c_{1i}, d_{kl}	coefficients needed for boundary correction	$S(\mathbf{x})$	Support domain for kernel K_m for $\mathbf{x}$
σ_i	standard deviation of samples $\{\mathbf{x}_h\}$ for i^{th} design variable	w_i	Bandwidth of Kernel for x_i
I^*	subset of I with $\tilde{C}(\mathbf{x}) \leq c^h$	$H(I^* I)$	ratio of average objective function values in I^* and I
$\delta(I^* I)$	Volume ratio between sets I^* and I	$\hat{H}(I^* I)$	Approximation to $H(I^* I)$ using stochastic simulation
$\{\mathbf{x}_u\}$	Uniform samples in I	$\{\mathbf{x}_o\}$	Uniform samples in I^*
ρ	target volume ratio for $\delta(I^* I)$ per iteration of NP-SSO	c^h	Threshold defining subset I^*
n_u	number of samples in $\{\mathbf{x}_u\}$	n_o	Number of samples in $\{\mathbf{x}_o\}$
$\{\mathbf{x}_p, \boldsymbol{\theta}_p\}$	samples available for formulation of proposal densities for stochastic sampling in the new search space	$\{\mathbf{x}_e, \boldsymbol{\theta}_e\}$	samples for which $h(\mathbf{x}, \boldsymbol{\theta})$ is known for guiding the formulation of proposal densities
n_p	number of samples in $\{\mathbf{x}_p, \boldsymbol{\theta}_p\}$	n_e	number of samples in $\{\mathbf{x}_e, \boldsymbol{\theta}_e\}$
D_{re}	relative entropy	$\mathbf{v}$	subset of $\boldsymbol{\theta}$ targeted for the AKSD
$K_G(\cdot)$	univariate kernel for AKSD	n_v	Size of $\mathbf{v}$
λ^j	local bandwidth factor for j^{th} sample	t_i	bandwidth for i^{th} component of $\mathbf{v}$
$\mathbf{d}$	vector with kernel characteristics	α	sensitivity factor
D_{min}	minimum relative entropy considered	s_e	percentage reduction for definition of D_{min}
c^s	stopping threshold for $H(I^* I)$	M	number of clusters for $\{\mathbf{x}_o\}$
$\cdot \cdot_k$ (subscript k)	k^{th} iteration of NP-SSO characteristics	$\cdot \cdot^m$ (superscript m)	m^{th} cluster of $\{\mathbf{x}_o\}$ characteristics
n_{sa}	number of samples per stage for adaptive selection of n_s	ρ_δ	accuracy threshold for identification of subset I^*
n_b	Stage count for adaptive selection of n_s	$n_{s,max}$	maximum allowable number of samples considered for n_s

		Problem D ₁ $n_x=2$		Problem D ₂ $n_x=4$			
		Iteration		Iteration			
		1	2	1	2	3	4
$\hat{H}(I_k^* I_k)$		0.792	0.996	0.403	0.823	0.931	0.998
$\hat{E}_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_{I_k}$		0.0378	0.0266	0.0632	0.0250	0.0214	0.0203
$\hat{C}(\mathbf{x}_{NP-SSO}^*)$		0.0281	0.0262	0.0208	0.0192	0.0195	0.0189
$\mathbf{x}_{NP-SSO}^*$	K_{pr} (MN/m)	42.63	123.98	114.81	73.66	79.89	79.47
	F_y (MN)	2.79	1.57	1.00	1.018	1.01	1.04
	K_p (MN/m)	-	-	4.16	0.16	4.73	4.30
	c_d (MN s/m)	-	-	2.46	3.93	3.81	4.01
$\sqrt[n_x]{V_{I_{NP-SSO}}}/V_X$		0.2		0.2			

Table 1: Results per iteration from a sample run of NP-SSO ($\rho = 0.2$)

	Sampling scheme	Problem D ₁ $n_x=2$ Iteration		Problem D ₂ $n_x=4$ Iteration					
		1	2	1	2 (two clusters)	3 (two clusters)	4		
Acceptance Rate (%)	DRS	5.88	4.37	6.25	2.34	2.55	2.19	2.26	2.08
	AKSD	17.11	16.85	17.1	16.91	16.86	18.03	14.36	16.89
	KSD	6.18	14.40	5.96	13.42	14.29	15.28	11.05	11.98
Ratio of AKSD to DRS rates		2.91	3.86	2.74	7.23	6.61	8.23	6.35	8.12
n_s (for AKSD)		600	600	1200	1200	1200	1800	1200	1200
Number of model evaluations per iteration (for AKSD)		6853	3176	13108	6906	6855	9029	7577	5176
Total Number of evaluations (for AKSD)		10029		48651					

Table 2: Sampling efficiency of adaptive kernel sampling density ($\rho = 0.2$)

		Problem D ₁ $n_x=2$		Problem D ₂ $n_x=4$		
		Iteration		Iteration		
		1	2	1	2	3
$\hat{H}(I_k^* I_k)$		0.718	0.999	0.335	0.918	0.997
$\hat{E}_{p(\mathbf{x})p(\boldsymbol{\theta})}[h(\mathbf{x}, \boldsymbol{\theta})]_{I_k}$		0.0358	0.0264	0.0672	0.0229	0.0211
$\hat{C}(\mathbf{x}_{NP-SSO}^*)$		0.0267	0.0261	0.0229	0.0228	0.0193
$\mathbf{x}_{NP-SSO}^*$	K_{pr} (MN/m)	122.31	121.11	58.12	141.23	38.47
	F_y (MN)	1.50	2.42	1.37	1.12	1.06
	K_p (MN/m)	-	-	1.03	0.69	1.51
	c_d (MN s/m)	-	-	2.32	1.72	5.41
$\sqrt[n_x]{V_{I_{NP-SSO}}/V_X}$		0.1		0.178		

Table 3: Results per iteration from a sample run of NP-SSO ($\rho = 0.1$)

	Sampling scheme	Problem D ₁ $n_x=2$ Iteration		Problem D ₂ $n_x=4$ Iteration		
		1	2	1	2	3
Acceptance Rate (%)	DRS	5.56	4.26	7.14	2.46	2.12
	AKSD	15.94	22.07	14.4	14.49	20.29
	KSD	5.14	20.82	6.86	11.41	11.40
Ratio of AKSD to DRS rates		2.87	5.18	2.02	5.89	9.57
n_s (for AKSD)		900	900	1200	1800	1200
Number of model evaluations per iteration (for AKSD)		9165	3820	12567	12191	5319
Total Number of evaluations (for AKSD)		12985		30077		

Table 4: Sampling efficiency of adaptive kernel sampling density ($\rho = 0.1$)

Figure 1: Illustration of sample-based approximation concepts for NP-SSO

(a) Actual density with an elliptical boundary

(b) Samples from the density

(c) KDE without boundary correction

(d) KDE-MBK approximation

Figure 2: Illustration of KDE approximation for complex boundaries

Figure 3: Identification of subsets by KDE-MBK and characterization by SVM

Figure 4: Illustration of building optimal AKSD within NP-SSO

Figure 5: Illustration of the adaptive NP-SSO algorithm

Figure 6: Structural model and a sample near-field excitation

Figure 7: Variation of sampling efficiency against the dimension (n_v) of the vector of model parameters considered for building AKSD