ETH zürich

Active-learning-based system reliability analysis with budget constraints

P. Parisi, M. Moustapha, S. Marelli, B. Sudret

Chair of Risk, Safety and Uncertainty Quantification
Stefano-Franscini-Platz 5
CH-8093 Zürich

Risk, Safety & Uncertainty Quantification

Data Sheet

Journal:	Proc. 14th Int. Conf. Applications of Statistics and Probability in Civil Engineering (ICASP14), Dublin (Ireland)
Report Ref.:	RSUQ-2023-006
Arxiv Ref.:	
DOI:	-
Date submitted:	
Date accepted:	July 9, 2023

14th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP14
Dublin, Ireland, July 9-13, 2023

Active-learning-based system reliability analysis with budget constraints

Pietro Parisi
Graduate student, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Zurich, Switzerland

Maliki Moustapha
Senior scientist, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Zurich, Switzerland

Stefano Marelli
Senior scientist, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Zurich, Switzerland

Bruno Sudret
Professor, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Zurich, Switzerland

ABSTRACT: Structural reliability analysis provides practitioners with tools to estimate the probability of failure of various engineering systems. Failure is characterized by a so-called limit-state function that takes as input a set of uncertain variables describing the system. Due to the complexity of engineering systems, multiple limit-state functions are generally needed. The problem is known as system reliability, as opposed to component reliability, for which only a single limit-state is considered. Surrogate-based solution schemes have shown to be the most efficient methods when used in an active learning scheme. Research efforts have mainly been devoted to component reliability analysis. Extensions or adaptations to system reliability have been proposed but they lack of efficiency. In this work, we propose an active learning scheme for solving system reliability problems in an arbitrary configuration while accounting for the difference in evaluation costs of the various limit-states. We use Sobol' sensitivity analysis and clustering to identify the relevant limit-state functions to update at each iteration. We then formulate a discrete optimization problem that allows us to account for the computational budget constraints and each limit-state evaluation cost. The proposed method is validated on an analytical example.

System reliability analysis aims at computing the failure probability of a structure given the various uncertainties it is subject to. In particular, it assumes that the structure safety or failure state is jointly described by a set of m so-called *limit-state functions* $\{g_j, j = 1,\dots,m\}$. Each of them takes as inputs a random vector $\mathbf{X} \sim f_{\mathbf{X}}$ and returns a scalar. By convention, negative values of each limit-state correspond to component failure. However, system failure is obtained by the joint contribution of each component through a composition function h. More precisely, the failure domain is denoted by $\mathscr{D}_f = \{\mathbf{x} \in \mathbb{R}^M : h(g_1(\mathbf{x}),\dots,g_m(\mathbf{x})) \leq 0\}$. The failure probability then reads:

$$P_f = \int_{\mathscr{D}_f} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}. \tag{1}$$

While we consider arbitrary system configurations, common configurations such as

series or *paralell* systems can be obtained by setting h as the minimum and maximum operator respectively, i.e., $\mathscr{D}_f^{\text{series}} = \left\{\mathbf{x} \in \mathbb{R}^M : \min\left(g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})\right) \leq 0\right\}$ and $\mathscr{D}_f^{\text{parallel}} = \left\{\mathbf{x} \in \mathbb{R}^M : \max\left(g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})\right) \leq 0\right\}$.

Solving Eq. (1) is not trivial and one generally resorts to approximation-, simulation- or surrogate-based methods (Melchers and Beck, 2018). Simulation methods such as crude Monte Carlo, importance sampling or subset simulation generally yield accurate results but are computationally intensive. They are in fact not affordable when expensive-to-evaluate computational models are used in the limit-state functions. Surrogate-based methods have proven to be an effective alternative. In such approaches, an inexpensive proxy of the limit-state function is built and used instead together with traditional simulation methods. The surrogate is built by evaluating the original limit-state on carefully selected samples known as the *experimental design*. Efficiency may be enhanced by building the surrogate in an active learning scheme, that is by sequentially enriching the experimental design so as to improve the accuracy of the limit-state surface in areas of high probability density. This eventually allows one to accurately estimate the probability of failure within a relatively small computational budget. Such methods have recently been reviewed in Teixeira et al. (2021); Moustapha et al. (2022).

These contributions are generally designed for component reliability problems and are inefficient when it comes to system reliability. This is due to some particular aspects of the latter, such as the presence of multiple and disjoint failure domains or their uneven contribution of the components to system failure. However, a few active learning schemes have been proposed to address some of the specific issues related to system problems. Examples include AK-SYS (Active Kriging - Monte Carlo simulation for system reliability) by Fauriat and Gayton (2014) or ALK-TCR (active learning Kriging with truncated candidate regions) by Yang et al. (2018).

In this work, we propose a generic methodology aiming at addressing the peculiarities of system reliability. Furthermore, we consider an additional constraint, more precisely budget constraints. We consider the case when a pre-defined computational budget is allocated for solving the reliability problem and when the different limit-state functions have different computational costs. We then aim to find the most effective enrichment scheme to accurately estimate the probability of failure given the available computational budget. To this end, we propose an active learning scheme that relies on a constrained optimization problem to find the optimal distribution of samples at each enrichment iteration.

The paper is organized as follows. In Section 1, we set up the problem and introduce various notations. We then present the proposed methodology in details in Section 2. Finally, we illustrate and validate the algorithm using an analytical benchmark function in Section 3.

1. PROBLEM SET-UP

Let us consider a set of m limit-state functions $\{g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})\}$ and assume that each of them has a computational cost $c_j, \{j = 1, \ldots, m\}$. The difference in cost is generally due to different types of models and may often range orders of magnitudes when different physics are considered.

In engineering practice, the resources allocated to a design stage are generally limited in time. When active learning reliability analysis is carried out, this may translate into a given allowed budget. Let us denote such a total budget by B^T. The goal is then to optimally use this budget within an enrichment strategy that allows one to find an accurate estimate of the probability of failure. For simplicity, let us assume that the initial experimental design is not included in this budget. Further assuming that the enrichment converges after K iterations, the following inequality must hold:

$$\sum_{j=1}^{m} c_j \left(\left| \mathscr{D}_j^{(K)} \right| - \left| \mathscr{D}_j^{(0)} \right| \right) \leq B^T, \tag{2}$$

where $\mathscr{D}_j^{(K)}$ is the experimental design corresponding to the j-th component limit-state at the K-th enrichment iteration and $|\bullet|$ denotes the cardinality of a set $\bullet$.

To accurately estimate the failure probability while satisfying Eq. (2), it is necessary to optimally allocate the computational resources to areas of the input space that contribute to failure as well as only evaluating the relevant limit-states. To this end, we devise an enrichment strategy that combines constrained optimization, sensitivity analysis and clustering.

2. METHODOLOGY

2.1. General workflow

The proposed methodology follows the traditional workflow of active learning reliability with specific steps taken within the enrichment step. This can be summarized as follows:

1. Build an initial experimental design $\mathscr{D}_j^{(0)}$ for each limit-state function.

2. Using the current experimental design, build m surrogate models, one for each limit-state function, say $\left\{\widehat{g}_j, j = 1,\dots,m\right\}$.

3. Using the surrogate models, estimate the failure probability with the approximate system limit-state $h\left(\widehat{g}_1(\mathbf{x}),\dots,\widehat{g}_m(\mathbf{x})\right)$.

4. Check for convergence: convergence is assumed if the relative variation of the reliability index within 3 consecutive iterations is below a given threshold $\left|\widehat{\beta}^{(k)} - \widehat{\beta}^{(k-1)}\right| / \widehat{\beta}^{(k-1)} < \varepsilon$, where $\widehat{\beta}^{(k)} = \Phi^{-1}\left(\widehat{P}_f^{(k)}\right)$ denotes the generalized reliability index computed from the probability of failure $\widehat{P}_f^{(k)}$. If convergence is achieved proceed to Step 6.

5. Enrich the experimental design by selecting one or more samples and go to Step 2.

6. Compute a final probability of failure using a reliability estimation algorithm setting that allows for a more precise estimate.

2.2. Kriging surrogates

In this work, we consider Kriging (Santner et al., 2003) as surrogate model. Kriging a.k.a. Gaussian process modelling is an interpolation method based on Gaussian processes, which casts the function to approximate as

$$\widehat{\mathscr{M}}(\mathbf{x}) = f(\mathbf{x})^T\mathbf{C} + \sigma^2 Z(\mathbf{x},\omega), \tag{3}$$

where the first summand, known as the trend, is cast here as a polynomial with unknown coefficients $\mathbf{C}$. The deviation from the trend is captured by the zero-mean unit variance Gaussian process $Z(\mathbf{x},\omega)$ fully characterized by the auto-correlation function $R(\mathbf{x},\mathbf{x}';\theta)$ and the process variance σ^2.

The main interest of Kriging is that it provides a measure of its own accuracy on top of the prediction. Given an experimental design, $\mathscr{D}_j = \left\{\left(\mathscr{X}_j^{(i)},\mathscr{G}_j^{(i)}\right) : \mathscr{G}_j^{(i)} = g_j\left(\mathscr{X}^{(i)}\right) \in \mathbb{R}, \mathscr{X}^{(i)} \in \mathbb{R}^M, i = 1,\dots,N_j\right\}$, Kriging assumes, for any unknown sample $\mathbf{x}$, that $\widehat{G}(\mathbf{x})$ follows a Gaussian distribution $\mathscr{N}\left(\mu_{\widehat{G}(\mathbf{x})}, \sigma^2_{\widehat{G}(\mathbf{x})}\right)$ with

$$\begin{aligned}\mu_{\widehat{G}(\mathbf{x})} =& \mathbf{f}^T(\mathbf{x})\widehat{\mathbf{C}} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}\left(\mathscr{G} - \mathbf{F}\hat{\beta}\right),\\ \sigma^2_{\widehat{G}(\mathbf{x})} =& \widehat{\sigma}^2\left(1 - \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) + \mathbf{u}^T(\mathbf{x})(\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{u}(\mathbf{x})\right),\end{aligned} \tag{4}$$

where $\mathbf{F}$ is the observation matrix with $F_{i,j} = f_j(\mathbf{x}^{(i)})$; $i = 1,\dots,N$; $j = 1,\dots,P$, $\mathbf{r}(\mathbf{x})$ is the vector of the cross-correlations with $r_i = R(\mathbf{x},\mathbf{x}^{(i)};\theta)$, $i = 1,\dots,N$ and $\mathbf{R}$ is the correlation matrix with elements $R_{i,j} = R(\mathbf{x}^{(i)},\mathbf{x}^{(j)};\theta)$, $i,j = 1,\dots,N$, $\widehat{\mathbf{C}} = (\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{R}^{-1}\mathscr{G}$ is the weighted least-square estimate of the regression coefficients and $\mathbf{u}(\mathbf{x}) = \mathbf{F}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) - \mathbf{f}(\mathbf{x})$ is introduced for convenience.

The Kriging prediction is eventually given by $\mu_{\widehat{G}(\mathbf{x})}$ while the variance $\sigma^2_{\widehat{G}(\mathbf{x})}$ gives indications on the local error associated to the prediction.

Once the surrogates are built, they can be used to replace the original limit-state function in the reliability analysis. We consider in this work subset simulation (Au and Beck, 2001) as a reliability estimation algorithm. The samples generated by subset simulation are considered as candidate points for enrichment in the next iteration. The enrichment step, which is the core of the proposed method, is now described in details.

2.3. Enrichment scheme

Let us assume that we are at the k-th enrichment iteration and denote by $\mathscr{X}_{\mathrm{SP}}^{(k)}$ the search space for

enrichment, which corresponds to the samples generated by subset simulation in the previous iteration. To find the points that are most likely to improve the system limit-state surface and henceforth the estimate of the failure probability, we solve the following constrained optimization problem:

$$\begin{aligned} \arg \max_{\mathscr{X} \in \mathscr{P}(\mathscr{X}_C)} \quad & \mathscr{I}(\mathscr{X}) \\ \text{s.t.} \quad & \mathscr{C}(\mathscr{X}) \leq B^{(k)}, \\ & \mathscr{X}_C = \mathscr{E}(\mathscr{X}_{SP}^{(k)}), \end{aligned} \tag{5}$$

where:

- $\mathscr{I}(\mathscr{X})$ is an improvement function that quantifies the improvement on the knowledge of the limit-state surface that can be attained by adding the set $\mathscr{X}$ to the experimental design;
- $\mathscr{C}(\mathscr{X})$ is the cost of evaluating the samples in $\mathscr{X}$ using the appropriate limit-state functions;
- $\mathscr{E}(\mathscr{X}_{SP}^{(k)})$ is the enrichment candidate function that selects from the search space the candidates $\mathscr{X}_C$ that will actually be considered for optimization;
- $\mathscr{P}(\mathscr{X}_C)$ is the power set of $\mathscr{X}_C$, that is the set of all subsets of $\mathscr{X}_C$ ranging from the empty set to $\mathscr{X}_C$ itself. It forms the search space for the optimization problem.

Let us now go through each of the ingredients of this optimization problem.

2.3.1. Improvement function

The improvement function is built using a so-called *learning function* whose goal is to provide the best next point to add to the experimental design. We consider here an adaptation of the popular deviation number U used in AK-MCS (active Kriging - Monte Carlo Simulation, Echard et al. (2011)). In this set-up, the next point to add to the experimental design is defined as:

$$\mathbf{x}_{\text{next}} = \arg\min_{\mathbf{x}} U(\mathbf{x}) = \frac{|\mu_{\widehat{G}}(\mathbf{x})|}{\sigma_{\widehat{G}}(\mathbf{x})}, \tag{6}$$

where $\mu_{\widehat{G}}(\mathbf{x})$ and $\sigma^2_{\widehat{G}}(\mathbf{x})$ are respectively the mean and variance of the Kriging model at the point $\mathbf{x}$. This simple yet powerful algorithm adds to the experimental design the points that are closest to the limit-state surface ($\mu_{\widehat{G}}(\mathbf{x}) \to 0$) and/or whose prediction is uncertain ($\sigma^2_{\widehat{G}}(\mathbf{x})$ is large).

In system reliability, there are multiple limit-states and choosing one such point for each limit-state is not efficient as they do not necessarily contribute to system failure. We therefore use a system adapted learning function that allows us to integrate the effect of the composition function h and henceforth to find the points that are likely to improve the knowledge of the system limit-state surface. This modified learning function simply reads:

$$U_{\text{sys}} = \frac{\left|\mu_{\text{sys}}(\mathbf{x})\right|}{\sigma_{\text{sys}}(\mathbf{x})}, \tag{7}$$

where $\mu_{\text{sys}}(\mathbf{x})$ and $\sigma_{\text{sys}}(\mathbf{x})$ are respectively the mean and standard deviation of the system limit-state approximation obtained through the composition function and the surrogates $h(\widehat{g}_1(\mathbf{x}), \ldots, \widehat{g}_m(\mathbf{x}))$. More precisely, they are computed empirically, i.e.,

$$\begin{aligned} \widehat{\mu}_{\text{sys}}(\mathbf{x}) &= \frac{1}{N_s} \sum_{i=1}^{N_s} h\left(\mathbf{z}^{(i)}\right), \\ \widehat{\sigma}^2_{\text{sys}}(\mathbf{x}) &= \frac{1}{N_s - 1} \sum_{i=1}^{N_s} \left(h\left(\mathbf{z}^{(i)}\right) - \widehat{\mu}_{\text{sys}}(\mathbf{x})\right)^2, \end{aligned} \tag{8}$$

where $\mathbf{z}^{(i)}$ are independent and identically distributed samples drawn from a multivariate Gaussian derived from the Kriging predictor:

$$\mathbf{Z}(\mathbf{x}) \sim f_{\mathbf{Z}}(Z) = \prod_{j=1}^{m} \mathscr{N}\left(\mu_{\widehat{G}_j}(\mathbf{x}), \sigma^2_{\widehat{G}_j}(\mathbf{x})\right). \tag{9}$$

Points that minimize U_{sys} are likely to be located near the current system limit-state surface or carry large prediction uncertainty. This information is embedded in the actual improvement function we propose, which reads:

$$\mathscr{I}(\mathscr{X}) = \sum_{i=1}^{|\mathscr{X}|} \left(\frac{1}{U_{sys}(\mathbf{x}^{(i)}) + \nu}\right)^{\alpha}, \quad \alpha \in [0, 1]. \tag{10}$$

In this equation, we introduce the cost sensitivity coefficient α that controls the effect of the learning

function on the enrichment: when $\alpha \to 0$, all candidates give the same contribution, and the dominant criterion in enrichment is actually the evaluation cost imposed through the constraint in Eq. (5). In contrast, when $\alpha \to 1$ more weight is given to the actual learning function U_{sys}. The term ν is a regularizer that allows us to avoid numerical instabilities when $U_{sys}(\mathbf{x}) \to 0$.

2.3.2. Enrichment candidate function

Before evaluating the improvement function, we reduce the samples produced by subset simulation. The idea is to keep a small number of relevant samples so as to maintain a reasonable optimization cost. To this end, we first filter the initial search space by keeping only a small percentage corresponding to those points that have the smallest U_{sys} values, i.e.,

$$\widetilde{\mathscr{X}_{\mathrm{SP}}} = \left\{\mathbf{x} \in \mathscr{X}_{\mathrm{SP}} : U_{\mathrm{sys}}(\mathbf{x}) < u_q\right\}, \tag{11}$$

where u_q is a lower q-quantile of U_{sys} as computed on $\mathscr{X}_{\mathrm{SP}}$. In this work, we set $q = 0.05$.

In the next step, we split the candidates into m subsets where each subset contains the samples that are relevant for each of the limit-state functions. The relevance is obtained through Sobol' sensitivity indices. More precisely, we consider for each sample $\mathbf{x}$ the Sobol' indices of the function $\mathbf{x} \mapsto h(\widehat{g}_1(\mathbf{x}), \ldots, \widehat{g}_m(\mathbf{x})) = h(\mathbf{Z}(\mathbf{x}))$. This can be obtained by a simple Monte carlo simulation given that $\mathbf{Z}(\mathbf{x})$ are multivariate Gaussian whose parameters are given in Eq. (9) and h is a simple analytical function.

In practice, for each sample $\mathbf{x}$ the total Sobol' indices $\left\{S_1^T(\mathbf{x}), \ldots, S_p^T(\mathbf{x})\right\}$ are computed. The sample is then added to the set $\mathscr{X}_{S_{j^*}^T}$ where j^* corresponds to the index of the largest Sobol' index, i.e.,

$$j^* = \arg \max_{j=\{1,\ldots,m\}} S_j^T(\mathbf{x}). \tag{12}$$

At the end of this process, the filtered candidate set for enrichment $\widetilde{\mathscr{X}_{\mathrm{SP}}}$ is divided into m subsets $\mathscr{X}_{S_1^T}, \ldots, \mathscr{X}_{S_m^T}$, each of them corresponding to the most relevant limit-state to system failure. To obtain the final search space for optimization, each of these subsets are clustered using a density-based clustering method (DBSCAN, (Ester et al., 1996)). The choice of this technique is explained by the fact that it does not require to know the number of clusters *a priori*. Instead, the latter is naturally derived from the data distribution and often corresponds to a specific failure modes. The points with lowest U_{sys} values in each cluster are eventually added to the set $\mathscr{X}_C$, which ultimately corresponds to the search space for the optimization problem in Eq. (5).

The entire process leading to the selection of a few samples for the optimization problem is summarized in the pseudo-algorithm described in Algorithm 1.

Algorithm 1 Enrichment candidate selection.

Input: Filtered search space $\widetilde{\mathscr{X}_{SP}^{(k)}}$
$\mathscr{X}_C \leftarrow \emptyset$
for i=1,...,m **do**
    $\mathscr{X}_{S_i^T} \leftarrow \emptyset$
for all samples $\mathbf{x} \in \widetilde{\mathscr{X}_{SP}^{(k)}}$ **do**
    Compute component Sobol' indices $S_1^T(\mathbf{x}), \ldots, S_m^T(\mathbf{x})$
    Add $\mathbf{x}$ to the set $\mathscr{X}_{S_i^T}$ of the largest Sobol' index S_i^T
for i=1,...,m **do**
    Apply clustering to $\mathscr{X}_{S_i^T}$
    for each cluster **do**
        Add $\tilde{\mathbf{x}}$, the point of the cluster with minimum U_{sys}, to $\mathscr{X}_C$

2.3.3. Enrichment cost function

The optimization problem in Eq. (5) is a combinatorial one that attempts to find the solution in $\mathscr{P}(\mathscr{X}_C)$, the power set of $\mathscr{X}_C$. Let us denote by $\mathscr{X}$ an arbitrary set in $\mathscr{P}(\mathscr{X}_C)$. $\mathscr{X}$ can be mapped to a unique total cost according to the limit-states the samples it contains are associated to. More precisely, each sample $\mathbf{x} \in \mathscr{X}$ is uniquely associated to the set $\mathscr{X}_{S_i^T}$ which corresponds to its largest Sobol' index as previously computed. The cost associated to the set of samples is therefore that of their corre-

sponding limit-state functions, i.e.,

$$\mathscr{C}(\mathscr{X}) = \sum_{i=1}^{|\mathscr{X}|} \sum_{j=1}^{m} c_j \mathbb{I}_{\left\{\mathbf{x}^{(i)} \in \mathscr{X}_{S_j^T}\right\}}, \tag{13}$$

where $\mathbb{I}$ is the indicator function, i.e., it is equal to 1 if $\mathbf{x}^{(i)} \in \mathscr{X}_{S_j^T}$ and to 0 otherwise.

2.4. Summary of the proposed cost-aware active learning scheme

Once samples are selected through the process described in the previous section, they are added to the appropriate experimental design. The surrogates are then updated and the probability of failure estimated again. The process is repeated until convergence. This is summarized in the following workflow.

1. **Initialization:**
   (a) Build initial experimental designs $\mathscr{D}_1^{(0)}, \ldots, \mathscr{D}_m^{(0)}$.
   (b) Construct surrogate models $\widehat{g}_1(\mathbf{x}), \ldots, \widehat{g}_m(\mathbf{x})$ using $\mathscr{D}_1^{(0)}, \ldots, \mathscr{D}_m^{(0)}$
2. **Calculate the U_{sys} learning function**
   (a) Estimate $\widehat{P}_f^{(k)}$ with subset simulation on $h(\widehat{g}_1(\mathbf{x}), \ldots, \widehat{g}_m(\mathbf{x}))$.
   (b) If convergence is reached go to step 5.
   (c) Obtain $\mathscr{X}_{SP}^{(k)}$ from samples of subset simulation.
   (d) Estimate U_{sys} for all samples of $\mathscr{X}_{SP}^{(k)}$ using Eqs. (7) to (9).
3. **Select the enrichment samples**
   (a) Filter the samples $\mathscr{X}_{SP}^{(k)}$ into $\widetilde{\mathscr{X}}_{SP}^{(k)}$ using Eq. (11).
   (b) Compute the total Sobol' indices $S_1^T, \ldots, S_m^T$ for all samples of $\widetilde{\mathscr{X}}_{SP}^{(k)}$.
   (c) Split $\widetilde{\mathscr{X}}_{SP}^{(k)}$ into m subsets $\mathscr{X}_{S_j^T}$ where each subset contains points whose largest Sobol' index is S_j^T.
   (d) Apply DBSCAN clustering to each set $\mathscr{X}_{S_j^T}$
   (e) For each cluster, add the point $\tilde{\mathbf{x}}$ with minimum U_{sys} to the candidate set $\mathscr{X}_C$.
   (f) Solve the optimization problem of Eq. (5) to determine the enrichment set $\mathscr{X}_{En}^{(k)}$.
4. **Enrich the surrogates**:
   (a) Add each sample of $\mathscr{X}_{En}^{(k)}$ with highest sensitivity index S_j^T to the experimental design $\mathscr{D}_j^{(k)}$ of the surrogate model $\widehat{g}_j(\mathbf{x})$.
   (b) Construct the surrogate models $\widehat{g}_1(\mathbf{x}), \ldots, \widehat{g}_m(\mathbf{x})$ with the updated experimental designs $\mathscr{D}_1^{(k)}, \ldots, \mathscr{D}_m^{(k)}$.
   (c) Go to step 2.
5. **Final estimate:** Estimate $\widehat{P}_f$ using subset simulation on $h(\widehat{g}_1(\mathbf{x}), \ldots, \widehat{g}_m(\mathbf{x}))$ with a larger number of samples.

3. EXAMPLE: FOUR-BRANCH FUNCTION

To illustrate the proposed methodology, we consider a popular reliability problem using the so-called four-branch function, which is defined as series system with:

$$\begin{aligned} &h\left(g_1(\mathbf{X}), g_2(\mathbf{X}), g_3(\mathbf{X}), g_4(\mathbf{X})\right) = \\ &\min\left(g_1(\mathbf{X}), g_2(\mathbf{X}), g_3(\mathbf{X}), g_4(\mathbf{X})\right), \end{aligned} \tag{14}$$

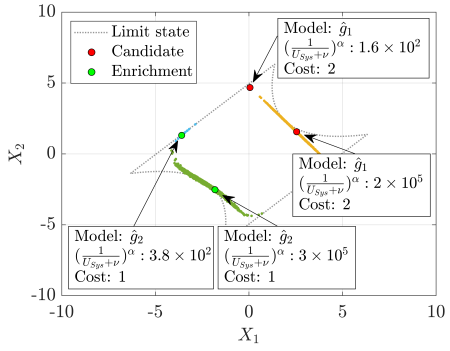
where

$$\begin{cases} g_1(\mathbf{X}) = 3 + 0.1(X_1 - X_2)^2 - \frac{1}{\sqrt{2}}(X_1 + X_2), \\ g_2(\mathbf{X}) = 3 + 0.1(X_1 - X_2)^2 + \frac{1}{\sqrt{2}}(X_1 + X_2), \\ g_3(\mathbf{X}) = (X_1 - X_2) + \frac{7}{\sqrt{2}}, \\ g_4(\mathbf{X}) = (X_2 - X_1) + \frac{7}{\sqrt{2}}, \end{cases} \tag{15}$$

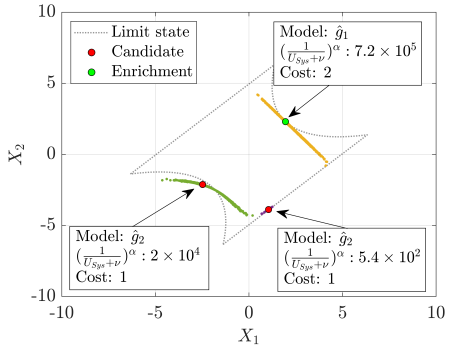
and $X_1, X_2 \sim \mathscr{N}(0, 1)$. The reference failure probability is $P_{f,\text{ref}} = 2.22 \cdot 10^{-3}$, which corresponds to a reliability index $\beta_{\text{ref}} = 2.84$.

To illustrate the algorithm, let us first consider the case when the component costs are defined as

$c_1 = 2, c_2 = 1, c_3 = 0.1$ and $c_4 = 0.1$. Figure 1 illustrates the first two enrichment iterations for a budget set to $B^{(k)} = 2$, where the initial experimental design is $N^{(0)} = 5$. The colored clouds of lines (green and orange) correspond to the filter and clustered sample sets $\mathscr{X}_{S_j^T}$. In this example the two linear limit-states g_3 and g_4, are already well defined and all samples associated to them are removed during the filtering step as their corresponding U_{sys} values are large (see Eq. (11)). The final candidate set $\mathscr{X}_C$ after clustering only consists of four and three points in iteration 1 and 2 respectively. They are shown by the large colored circles. The green ones correspond to the ones that will eventually form the enrichment set $\mathscr{X}_{En}$, that is the solution of the optimization problem. It can be seen by the information given in the figures that their improvement function are the largest, while their combined cost is equal to the iteration budget.



(a) Iteration 1



(b) Iteration 2

Figure 1: Illustration of the enrichment for two iterations in the four-branch function example.

Generally, the enrichment scheme is greatly influenced by the difference in the computational costs between the limit-states $c_1, \ldots, c_m$ (with regard to the computational budget $B^{(k)}$) and by the coefficient α. To study their effect on the convergence of the algorithm, we consider three different cases corresponding to different limit-state costs as illustrated in Table 1. On top of that, we consider three different values for the coefficient α, namely $\{0, 0.5, 1\}$. This gives a total of 9 problems that are solved using the proposed methodology. For these case studies, we consider an iteration budget $B^{(k)} = 3$ and a total budget $B^T = 60$. The initial experimental design size is still $N^{(0)} = 5$ for all limit-states.

Table 1: Different cases for the four-branch function.

Case	c_1	c_2	c_3	c_4
A	3	1	0.1	0.1
B	2	1	0.1	0.1
C	1	1	0.1	0.1

Figure 2 shows the boxplots of the resulting errors considering 15 repetitions of the analysis for each of the 9 cases. The relative error $\varepsilon_{P_f}^{(i)}$ of the i-th repetition is obtained with respect to the reference solution $P_{f,\text{ref}}$, i.e., $\varepsilon_{P_f}^{(i)} = |\widehat{P}_f^{(i)} - P_{f,\text{ref}}|/P_{f,\text{ref}}$. It can first be noted that the error in the estimation decreases from cases A to C. This is due to the fact that the cost of g_1 decreases, thus offering more computational budget. We can also observe that as the value of α increases, the probability of failure estimate is getting more accurate. This is because larger values of α put more emphasis on the contribution of the learning function U_{sys} in the improvement function, hence leading to more accurate estimates.

These observations can be confirmed by the median values given in Table 2. Looking at the number of model evaluations for each case, we can note that the number of evaluations of g_1 increases when moving from case A to case C. This is consistent with the fact that the cost of evaluating g_1 decreases from A to C. Finally, for the same cost configurations we can note that the difference in number of evaluations between g_1 and g_2 decreases as α increases. This is due to the fact that g_1 and g_2 have

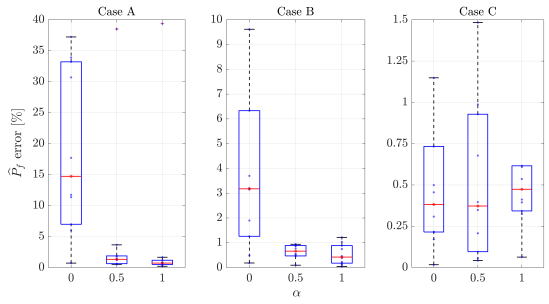


Figure 2: Relative error for the different cases considered for the four-branch function.

the same contribution to system failure, while small values of α, especially when $\alpha = 0$, puts more emphasis on the effect of the evaluation cost rather than the actual value of the learning function.

Table 2: Four-branch function median results for cost-aware simulations.

Case	α	ε_{P_f} [%]	# Eval. g_1	# Eval. g_2
A	0	14.7	7.5	48
A	0.5	1.33	13.5	33.5
A	1	0.720	14	37.5
B	0	3.18	12	50
B	0.5	0.662	23	29
B	1	0.423	23	29
C	0	0.382	25	40
C	0.5	0.373	37	33
C	1	0.474	34	36

4. CONCLUSIONS

In this work, we have proposed an active learning reliability method for the solution of system reliability problems with budget constraints. The proposed method relies on an optimization problem that aims at finding a combination of points that yield the largest improvement in the system limit-state surface while having an evaluation cost that is below the computational budget allowed at each iteration. The search space for the optimization problem is obtained by combining a learning function, clustering and sensitivity analysis. The latter allows us to select for each enrichment candidate the limit-state function that actually contributes to system failure. The proposed method is flexible enough as it allows to control the explorative behavior of the enrichment scheme through a cost sensitivity coefficient. We have used a toy problem to investigate how the enrichment is affected by various evaluation costs as well. The results have shown that the algorithm is efficient and effectively aware of the computational cost, yet it prefers in general enriching the cheaper models whenever possible.

5. REFERENCES

Au, S. K. and Beck, J. L. (2001). "Estimation of small failure probabilities in high dimensions by subset simulation." *Probabilistic Engineering Mechanics*, 16(4), 263–277.

Echard, B., Gayton, N., and Lemaire, M. (2011). "AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation." *Structural Safety*, 33(2), 145–154.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise." *Proc. 2nd International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, USA, August 2-4, 1996*, E. Simoudis, J. Han, and U. Fayyad, eds., AAAI press.

Fauriat, W. and Gayton, N. (2014). "AK-SYS: An adaptation of the AK-MCS method for system reliability." *Reliability Engineering & System Safety*, 123, 137–144.

Melchers, R. and Beck, A. (2018). *Structural reliability analysis and prediction*. John Wiley & Sons.

Moustapha, M., Marelli, S., and Sudret, B. (2022). "Active learning for structural reliability: Survey, general framework and benchmark." *Structural Safety*, 96(102174).

Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer, New York.

Teixeira, R., Nogal, M., and O'Connor, A. (2021). "Adaptive approaches in metamodel-based reliability analysis: A review." *Structural Safety*, 89, 102019.

Yang, X., Liu, X., Mi, C., and Tang, C. (2018). "System reliability analysis through active learning Kriging model with truncated candidate region." *Reliability Engineering & System Safety*, 169, 235–241.