Combined Gaussian process regression model and comprehensive learning particle swarm optimizer in reliability-based structural optimization

***Thu Huynh Van[1], Bach Do[2], Suchart Limkatanyu[3] and †Sawekchai Tangaramvong[1]**

[1] Applied Mechanics and Structures Research Unit, Department of Civil Engineering, Chulalongkorn University, 10330 Bangkok, Thailand
[2] Department of Architecture and Architectural Engineering, Graduate School of Engineering, Kyoto University, Kyoto-Daigaku Katsura, Nishikyo, Kyoto 615-8540, Japan
[3] Department of Civil Engineering, Faculty of Engineering, Prince of Songkla University, Songkla 90110, Thailand

*Presenting author: thuxd11@gmail.com
†Corresponding author: sawekchai.t@chula.ac.th

Abstract

Reliability-based design optimization (RBDO) addresses the cost-effective integrity design of structures in the presence of inherent uncertain parameters. Processing this class of problem is challenging from the computational burden to determine the failure probability of structures violating the limit-state function. This paper proposes an efficient decoupling RBDO method that advantageously couples a comprehensive learning particle swarm optimization (CLPSO) algorithm with a Gaussian process regression (GPR) model, termed as GPR-CLPSO approach. In essence, the proposed method iteratively performs the CLPSO with deterministic parameters based on the most probable point underpinning the limit-state function iterative updated by the reliability evaluation process. The GPR model approximates from the design data given by CLPSO the spectrum of the limit-state function under uncertain parameters, and hence enables a significant reduction of Monte-Carlo simulations for estimation of failure probability. What is more, is that a so-called expected feasibility function is maximized to systematically refine the GPR model by locating new sampling points in the region with high-reliability sensitivity leading to the more accurate prediction of failure probability. The RBDO terminates when the resulting failure probability reaches the allowable threshold. The CLPSO is primarily adopted in the optimization process for the GPR hyperparameters and the expected feasibility function. A numerical example is provided to illustrate the applications and robustness of the proposed schemes in solving the RBDO problems.

Keywords: Gaussian process regression; Comprehensive learning; Particle swarm optimization; Expected feasibility function; Reliability-based design optimization.

Introduction

Deterministic optimization has been extensively applied in engineering structures to improve the design performance with minimum resources. The design solution computed by the deterministic optimization becomes unreliable in some cases, especially when the influences of uncertainties inheriting structural dimensions, material properties, loading and operating conditions are significant and cannot be eliminated. By addressing the performance and reliability of the structure together, the structural reliability-based design optimization (RBDO) has been considered as the alternative approach in recent years. More explicitly, the RBDO problem minimizes the cost function, denoted as *C*, and satisfies the certain deterministic and probabilistic constraints, as state by the following generic mathematical formulations [1]:

$$\begin{aligned} &\min C(\mathbf{s}) \\ &\text{s.t.}\;\; \mathsf{P}\left[G(\mathbf{s},\mathbf{x}) - \bar{z} \le 0\right] - P_a \le 0, \\ &\quad\;\; \mathbf{s} \in \left[\mathbf{s}_\mathrm{L},\, \mathbf{s}_\mathrm{U}\right] \end{aligned} \tag{1}$$

where $\mathbf{s}$ and $\mathbf{x}$ are the vectors of deterministic design variables and random parameters, respectively. The vector $\mathbf{x}$ is characterized by the joint probability density function (PDF) $f(\mathbf{x})$ in the space Ω. The two $\mathbf{s}_\mathrm{L}$ and $\mathbf{s}_\mathrm{U}$ denote the lower and upper bound on the variables $\mathbf{s}$, respectively. The functions $G(\mathbf{s}, \mathbf{x})$ and $G(\mathbf{s},\mathbf{x}) - \bar{z}$ are respectively the performance and limit-state design expressions considered, where $\bar{z}$ is a constant threshold of $G(\mathbf{s},\mathbf{x})$. The failure domain specified for the design variables $\mathbf{s}$ (i.e., $\mathsf{F} = \left\{\mathbf{x} \in \Omega \,|\, G(\mathbf{s},\mathbf{x}) - \bar{z} \le 0\right\}$) reads the probability of failure, namely $P_f =$ $[G(\mathbf{s}, \mathbf{x}) - \bar{z} \le 0] = \int\ldots\int_{\mathsf{F}} f(\mathbf{x})\, \mathrm{d}\mathbf{x}$, where P_a is the allowable threshold of P_f.

Various optimization approaches have been developed to solve the problem stated in Eq. (1), mainly categorized by the double-loop, single-loop and decoupling approaches [2, 3]. Despite of some gain in efficiency, solving the RBDO problem in large-scale applications is very challenging. For example, some of the widely-employed conventional methods, such as Monte-Carlo simulation (MCS) [4] and first-order reliability method (FORM) [5], are endowed by the drawbacks related to the expensive computational cost and/or the result error. The quest for the efficient optimization method is essential and ongoing in the research field. Recently, the surrogate models [6-8], such as artificial neural networks, radial basic function, support vector machine and Gaussian process regression (GPR), have been adopted to alleviate the computational burden and the solution inaccuracy that would be possibly experienced by standard techniques.

From the above comments, this paper proposes the novel combined GPR-CLPSO approach that efficiently process the accurate solution of the decoupling form of the problem in Eq. (1), where CLPSO is an acronym for comprehensive learning particle swarm optimization [9]. At the beginning, the CLPSO solves the counterpart to Eq. (1) formulated using the initial vector of random parameters, called the most probable point (MPP). From the optimal result determined by the CLPSO, the GPR is constructed as a surrogate model to approximate the spectrum of the limit state functions under uncertainty parameters, thereby enabling the low-cost MCSs to estimate the failure probability associated with the optimal design. To enhance the accuracy of the failure probability estimate, the GPR model is strategically refined by adding new sampling points to the region with high-reliability sensitivity, which is further transferred to maximize a so-called expected feasibility function (EFF). The CLPSO is then adopted to optimize both the GPR hyperparameters and the learning function EFF. For each decoupling iteration, the MPP is redefined in the CLPSO algorithm to search for the new optimal design solution. The optimization terminates as when the failure probability associated with the design by the deterministic procedures reaches the allowable thresholds. Finally, the robustness and accuracy of the proposed method is illustrated through the benchmark on the problem of a ten-bar truss.

CLPSO Algorithm

Whilst the general PSO algorithm provides the good convergence rate to the optimal solutions, the method is often trapped into the local optima. To enable the diversity of design particles and overcome the premature convergence, Liang and Huang [9] developed the CLPSO algorithm using the comprehensive learning (CL) strategy. The velocity and position functions of CLPSO in the next time step $t + 1$ are written as follows:

$$\mathbf{V}_i(t+1) = w\mathbf{V}_i(t) + c_1 r_1 \left(\mathbf{pbest}_{fi(D)} - \mathbf{X}_i(t)\right) + c_2 r_2 \left(\mathbf{gbest} - \mathbf{X}_i(t)\right) \tag{2}$$

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1) \tag{3}$$

where $fi(D) = [fi(1), fi(2),\ldots, fi(N_d)]$ indicates if the i-th particle follows its own or some other best (called $\mathbf{pbest}_i$) position for each dimension $D \in \{1,\ldots,N_d\}$; N_d defines the total number of searching dimensions; $\mathbf{X}_i(t) = [X_i^1(t),\ldots,X_i^D(t)]$ and $\mathbf{V}_i(t) = [V_i^1(t),\ldots,V_i^D(t)]$ denote the position and velocity of the i-th particle for $i \in \{1,\ldots,N_p\}$ (N_p is the total number of particle populations) at the t-th time, respectively; w is an inertial weight; c_1 and c_2 are the two accelerating coefficients; r_1 and r_2 are the two independent random numbers lying uniformly within the [0,1] interval; $\mathbf{pbest}_{fi(D)}^D$ is the exemplar of the i-th particle on the D-th dimension, as well as those with reference to the best position found by its own **pbest**; and **gbest** is the best position of the whole swarm

The CL strategy can be briefly described in the following three-step implementations [10].

Step i. For a generic i-th particle, the position $\mathbf{X}_i(t)$ is tested for its solution convergence and optimality. If the deterministic objective functions $C(\mathbf{s} = \mathbf{X}_i(t))$ over m consecutive time steps (m is defined as a refreshing gap) all are larger than the recorded best result $C(\mathbf{s} = \mathbf{pbest}_i)$, perform the CL searching on the i-th particle for the best new best position $\mathbf{pbest}_i$ in *Step ii*.

Step ii. For each of the D-th dimensions of the i-th best particle $\mathbf{pbest}_i$, update based on the learning probability Pc_i in Eq. (4) an exemplar $\mathbf{pbest}_{fi(D)}^D$ from the two random particles $\mathbf{pbest}_{fi(D)}$ within the whole **pbest** populations that yields the lower objective function $C(\mathbf{s} = \mathbf{pbest}_{fi(D)})$.

$$Pc_i = 0.05 + 0.45\left(\exp\left(\frac{10(i-1)}{N_p - 1}\right) - 1\right) / \left(\exp(10) - 1\right) \tag{4}$$

Step iii. Repeat *Step ii* for all dimensions $D \in \{1,\ldots,N_d\}$ to generate the new best position $\mathbf{pbest}_i = \mathbf{pbest}_{fi(D)}$ of the i-th particle. In the case when some exemplar $\mathbf{pbest}_{fi(D)}^D$ returns the current $\mathbf{pbest}_i^D$, it randomly learns from another i-th best particle of the same D-th dimension.

GPR Model

Consider the training set $\Delta = (\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_i, y_i) \mid i = 1,\ldots, N\}$ of N observations, where $\mathbf{x}_i \in \mathbb{R}^D$ are the input variables and $y_i \in \mathbb{R}$ is the corresponding output variable (e.g., structural response in this case). The goal is to construct an input-output mapping $y = f(\mathbf{x}) + \epsilon$: $\mathbb{R}^D \rightarrow \mathbb{R}$, where $f(\mathbf{x})$ is the unknown regression function and the noise $\epsilon \sim \mathrm{N}(0, \sigma^2_N)$.

The GPR is defined as the distribution over the set of functions $f(\mathbf{x})$ [11, 12]. The distribution is characterized by the mean value function $m(\mathbf{x})$ to represent a prior knowledge about the regression function and the kernel function $\kappa(\mathbf{x}, \mathbf{x}')$ to control the smoothness of the function, such that

$$f(\mathbf{x}) \sim \mathrm{GP}\,(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}')) \qquad (\mathbf{x}, \mathbf{x}') \in \mathbb{R}^{D}. \tag{5}$$

Commonly, a zero-mean function (i.e., $m(\mathbf{x}) = 0$) and the following squared exponential kernel are used in Eq. (5) [11, 12].

$$\kappa(\mathbf{x}, \mathbf{x}') = \alpha_f^2 \exp\left(-\frac{1}{2} \sum_{d=1}^{D} \frac{\left(x_d - x_d'\right)^2}{l_d^2} \right), \tag{6}$$

where the hyperparameters $\theta = \left\{ \alpha_f, l_d \right\}$ consist of signal variance α_f and length-scale l_d.

Suppose one already defines Δ and aims to predict the responses $\mathbf{y}^* \in \mathbb{R}^{N^{*}\times 1}$ for the new set of input variables $\mathbf{X}^* \in \mathbb{R}^{N^{*}\times D}$. As the GPR nature, the jointed PDF $p(\mathbf{y},\mathbf{y}^*;\mathbf{X},\mathbf{X}^*)$ of $\mathbf{y}$ and $\mathbf{y}^*$ placing the condition on $\mathbf{X}^*$ and $\mathbf{X}$ is described by the Gaussian [13]

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_N^2 \mathbf{I} & \mathbf{K}(\mathbf{X},\mathbf{X}^*) \\ \mathbf{K}^{\mathrm{T}}(\mathbf{X},\mathbf{X}^*) & \mathbf{K}(\mathbf{X}^*,\mathbf{X}^*) \end{bmatrix} \right) \tag{7}$$

where $\mathbf{K}(\mathbf{X},\mathbf{X}) \in \mathsf{R}^{N\times N} = \{\kappa(\mathbf{x}_i,\mathbf{x}_j) \,|\, \forall i, \forall i \in (1,\ldots,N)\}$ is the covariance matrix of all input variables (viz., similarly for $\mathbf{K}(\mathbf{X},\mathbf{X}^*) \in \mathsf{R}^{N\times N^*}$ and $\mathbf{K}(\mathbf{X}^*,\mathbf{X}^*) \in \mathsf{R}^{N^*\times N^*}$), and $\mathbf{I} \in \mathsf{R}^{N\times N}$ is an identity matrix.

By applying the conditional distribution to the PDF in Eq. (7), the predictive posterior PDF of the GPR model can be established. Both the posterior mean vector $\boldsymbol{\mu}^*$ and covariance matrix $\boldsymbol{\Sigma}^*$ are analytically expressed as

$$\begin{aligned} \boldsymbol{\mu}^* &= \mathbf{K}^{\mathrm{T}}(\mathbf{X},\mathbf{X}^*)\left[\mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_N^2 \mathbf{I}\right]^{-1} \mathbf{y} \\ \boldsymbol{\Sigma}^* &= \mathbf{K}(\mathbf{X}^*,\mathbf{X}^*) - \mathbf{K}^{\mathrm{T}}(\mathbf{X},\mathbf{X}^*)\left[\mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_N^2 \mathbf{I}\right]^{-1} \mathbf{K}(\mathbf{X},\mathbf{X}^*) \end{aligned}. \tag{8}$$

Determining the hyperparameters θ that best fits the data set Δ is essential for the construction of the GPR model. Based on the statistical regression, the optimal parameter $\hat{\theta}$ reads the maximizer of the marginal likelihood $\mathsf{L} = p(\mathbf{y};\mathbf{X},\theta)$, and hence the minimize of the (negative) log-marginal likelihood L of the training set Δ. As is clear,

$$\hat{\theta} = \underset{\theta}{\mathbf{arg\,min}}\left(-\mathbf{log}(\mathsf{L})\right) \tag{9}$$

where

$$-\mathbf{log}(\mathsf{L}) = \frac{N}{2}\mathbf{log}(2\pi) + \frac{1}{2}\mathbf{log}\left|\mathbf{K}(\mathbf{X},\mathbf{X};\theta) + \sigma_N^2 \mathbf{I}\right| + \frac{1}{2}\mathbf{y}^T \left|\mathbf{K}(\mathbf{X},\mathbf{X};\theta) + \sigma_N^2 \mathbf{I}\right|^{-1} \mathbf{y} \tag{10}$$

In this study, the problem in Eq. (9) is processed by the CLPSO schemes.

Failure Probability Estimation Using MCS and Learning Function EFF

Let $\hat{G}(\mathbf{x}) \sim N\left(\mu_{\hat{G}}(\mathbf{x}), \sigma^2_{\hat{G}}(\mathbf{x})\right)$ denote the GPR model of the performance function $G(\mathbf{s},\mathbf{x})$ evaluated at the particular design variables $\mathbf{s}$ (or $G(\mathbf{x})$ for simplification), where $\mu_{\hat{G}}(\mathbf{x})$ and $\sigma^2_{\hat{G}}(\mathbf{x})$ are provided in Eq. (8). The MCSs can be performed to approximate the mean function $\mu_{\hat{G}}(\mathbf{x})$ and thus the failure probability in Eq. (1) by

$$P_f = \int_{\Omega} f(\mathbf{x}) \quad [\mu_{\hat{G}}(\mathbf{x}) - \overline{z}]\,\mathrm{d}\mathbf{x} = \frac{1}{N_{MCS}} \sum_{k=1}^{N_{MCS}} \quad [\mu_{\hat{G}}(\mathbf{x}_k) - \overline{z}] \tag{11}$$

where N_{MCS} is the number of random samples generated within the space Ω, and [.] denotes the indicator function, namely

$$[\mu_{\hat{G}}(\mathbf{x}_k) - \overline{z}] = \begin{cases} 1 & \text{if} \quad \mu_{\hat{G}}(\mathbf{x}_k) - \overline{z} \le 0 \\ 0 & \quad\text{otherwise} \end{cases} \tag{12}$$

The prediction of P_f in Eq. (11) considers the uncertainty, but $\hat{G}(\mathbf{x})$ may not be able to capture the failure domain of the performance function, especially at the border between safe and unsafe ones. For this reason, the present approach adopts the learning function EFF [14] for the generation of the new training point added to the training set Δ giving the systematic refinement of the GPR model $\hat{G}(\mathbf{x})$. In essence, the learning function EFF describes the expectation of the true value of responses that can satisfy the equality constraint $G(\mathbf{x}) - \overline{z} = 0$ based on $\mu_{\hat{G}}(\mathbf{x})$ and $\sigma_{\hat{G}}(\mathbf{x})$ in the current GPR model $\hat{G}(\mathbf{x})$. This expectation enables the searches over the vicinity of the response threshold, where the estimation of P_f is the most sensitive. The analytical expressions of EFF can be found in [14]. Its determination processes the CLPSO schemes.

Combined GPR-CLPSO Algorithm

This section summarizes the proposed GPR-CLPSO algorithm as the following step-by-step procedures:

Step 1: Initialize the random variables in the original RBDO Eq. (1).

Step 2: Perform the deterministic optimization using the CLPSO algorithm to obtain the optimal design solutions.

Step 3: Employ the Latin hypercube sampling (LHS) method to generate random samples for the design obtained in *Step 2*. Calculate the actual responses associated with the samples using the limit state functions to create the training set.

Step 4: Construct the GPR model for the limit state functions based on the training set obtained in *Step 3*. The optimal hyperparameters for the GPR model are determined by the CLPSO algorithm.

Step 5: Compute the maximizer of the learning function EFF, and evaluate the associated actual response functions using the CLPSO schemes.

Step 6: Add the point obtained in *Step 5* and the associated response to the training data set. Reconstruct the GPR model, and re-iterate *Step 5* until the number of added points reaches the specified value *n*. The recommended number of added points is $n \in [5, 10]$.

Step 7: Based on the GPR model in *Step 6*, perform the MCSs with N_{MCS} random samples to estimate the failure probability P_f. Then, update the new MPP for the next deterministic optimization.

Step 8: Check the convergent criteria on the failure probability. If the estimated failure probability P_f converges, terminate. Otherwise, repeat *Steps 2* to *8*.

Illustrative Example

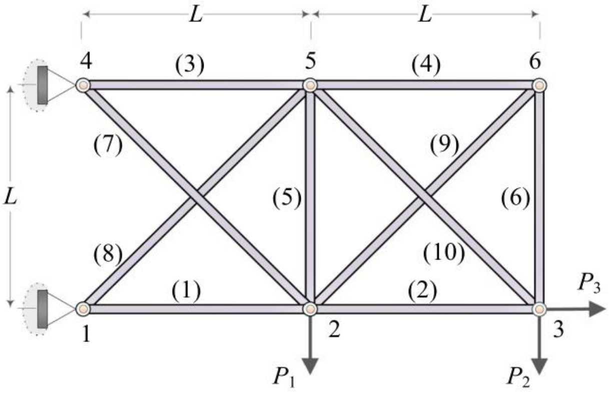

Figure 2. Ten-bar truss.

A 10-bar truss in Fig. 2 was considered, and its RBDO problem was formulated in Eq. 1 [15]. The cross-sectional areas were employed the design variables, namely $\mathbf{s} = [s_1, \ldots, s_{10}]^T$, where each variable is independently bounded within the intervals, i.e., $s_i \in [1, 20] \times 10^{-4}$ m2. The intrinsic (indeterministic) uncertainties inherited the external loads (P_1, P_2 and P_3), Young's modulus (E) and dimension length (L), simultaneously. The random variables were $\mathbf{x} = [P_1, P_2, P_3, E, L]^T$ with the probabilistic properties listed in Table 1.

The vertical displacement at node 3, denoted as Δ_3, was considered as the response performance of interest ($y = -\Delta_3$), whose probability of exceeding the allowable value of $\bar{z} = 4 \times 10^{-3}$ m was less than or equal to $P_a = 6.21 \times 10^{-3}$. Thus, the specific RBDO Eq. (1) was written as follows:

$$\begin{aligned} &\min C(\mathbf{s}) = \sum_{i=1}^{10} s_i \\ &\text{s.t.}\ \ \mathsf{P}\left[G(\mathbf{s},\mathbf{x}) + 4\times10^{-4} \le 0\right] - 6.21\times10^{-3} \le 0\,. \\ &\qquad s_i \in [1,\ 20]\times10^{-4},\ i = 1,\ldots,10 \\ &\qquad G(\mathbf{s},\mathbf{x}) = -\Delta_3 \end{aligned} \tag{13}$$

In the first instance, the initial random variables were assigned to take the mean values given in Table 1. The deterministic counterpart of the problem in Eq. (13) was solved using the CLPSO algorithm. The parameters adopted were: the total number of particle populations of N_p = 20; $c_1 = c_2 = 2$; the inertial weight of w linearly declining from 0.9 to 0.2; and the maximum number of iterations of 500 as per each particle set.

The optimal deterministic results **s** computed were adopted as the initial inputs for the construction of the GPR model that approximated the limit state functions in Eq. (13), where the LHS employed 75 training points and each decoupling iteration adopted n = 5 added points. The failure probability of the design problem was estimated using the CLPSO algorithm, and the MPP was updated. The MCSs were performed over the constructed GPR model without the need of any further finite element analysis (FEA) solves giving the computational advantages in approximating the failure probability P_f in Eq. (11). The coupling CLPSO and GPR procedures were iterated until the estimated probability of failure (viz., $P_f = 5.43\times10^{-3}$) was converged and well complied with the limit of $P_a = 6.21\times10^{-3}$. The proposed GPR-CLPSO method was encoded in Python, and run using the computer hardware with Intel Core i5-9400 CPU @ 2.9 GHz and 16 GB RAM.

The optimal solutions successfully designed by the present GPR-CLPSO method are reported in Table 2 that also provides the direct comparisons to some other benchmarks [1, 15, 16]. It presents the good agreement between the proposed approach and various design techniques in determining the optimal solutions to the RBDO problem in Eq. (13). More explicitly, the proposed GPR-CLPSO computed the optimal design of $C(\mathbf{s}) = 61.293 \times 10^{-4}$ m^2 that is not only close but also lies (most minimum) lower than all reported values, namely 62.367×10^{-4} m^2 in FEA-FORM [15], 63.649×10^{-4} m^2 in RSM2-FORM-MCS [15], 62.347×10^{-4} m^2 in PSA-ISAP [16] and 61.482×10^{-4} m^2 in MGP-SA [1].

The computional efforts (i.e., taking 2,213 s in CPU times) were modest, and the scheme only required 225 FEA iterates with 15 addded points in the learning function EFF. The corresponding failure probability of $P_f = 5.43 \times 10^{-3}$ complied with the limit of $P_a = 6.21 \times 10^{-3}$, and was close to the reference value of $P_f = 5.21 \times 10^{-3}$ given by standard MCSs exhaustively generating 5×10^5 random samples on the designed structure. Finally, the plot of the resulting optimal design layout is depicted in Fig. 3, where the sizes of the lines indicate the relative normalized member areas.

Table 1. Probabilistic properties of random parameters.

Variable	Distribution	Mean value	COV
P_1 [kN]	Normal	60	0.20
P_2 [kN]	Normal	40	0.20
P_3 [kN]	Normal	10	0.20
E [GPa]	Normal	200	0.10
L [m]	Normal	1	0.05

Table 2. Comparisons of optimization results for various analysis approaches.

Design variables	FEA - FORM [15]	RSM2- FORM- MCS [15]	PSA-ISAP [16]	MGP-SA [1]	GRP-CLPSO (This Study)
s_1	10.493	10.705	10.482	10.333	10.635
s_2	5.772	5.914	4.421	5.371	5.589
s_3	14.098	14.424	15.685	13.579	13.481
s_4	1.000	1.000	1.089	1.000	1.000
s_5	1.000	1.000	1.000	1.000	1.000
s_6	1.000	1.000	1.000	1.000	1.000
s_7	5.460	5.531	7.851	6.418	5.883
s_8	11.586	11.853	10.048	11.273	11.149
s_9	1.000	1.000	1.121	1.000	1.000
s_{10}	10.958	11.223	9.650	10.508	10.555
$C(\mathbf{s})$ [$\times 10^{-4}$ m^2]	**62.367**	**63.649**	**62.347**	**61.482**	**61.293**
No. of FEA iterates	**2,240**	**1,904**	**524**	**20,000**	**225 + 15**
Computational time [s]	**-**	**-**	**-**	**3,208**	**2,213**

$[G(\mathbf{s}, \mathbf{x}) \leq 0]$	**8.51×10^{-3}**	**6.19×10^{-3}**	**6.19×10^{-3}**	**4.34×10^{-3}**	**5.43×10^{-3}**
P_f(MCS) (with 5×10^5 samples)	**4.22×10^{-3}**	**2.95×10^{-3}**	**6.15×10^{-3}**	**5.64×10^{-3}**	**5.21×10^{-3}**

Note that: +15 is the total number of added points from the learning function EFF.

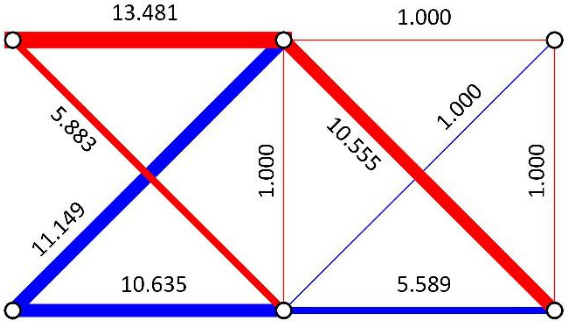

Figure 3. Normalized cross-sectional areas of optimally designed members, where blue and red colors indicate compression and tension, respectively.

Concluding remarks

The paper has presented the computationally efficient GPR-CLPSO method that couples the CLPSO algorithm with the GPR model in processing the decoupling RBDO problem. The CLPSO schemes in particular initialize the nominal design points (assuming the deterministic optimization), which are subsequently adopted to estimate the failure probability and the updated MPP underpinning the limit state functions. The GPR model is constructed such that the reliability responses of limit state functions can be mapped out using only few training points (enhanced by the use of a so-called learning function EFF). What is interesting is that the method advantageously by-passes the finite element analysis (time consuming) computations as when the GPR model has been estimated. The MCSs can be fully enumerated on the GPR model at modest computing efforts even for a large number of samples.

A number of RBRO problems and benchmarks (one of which has been provided herein) have been successfully performed by the GPR-CLPSO approach. These illustrate the accuracy and robustness of the proposed method.

Acknowledgements

This research has been supported under the Thailand Research Fund (Grant No. RTA6280012). The financial support provided by Chulalongkorn University under Ratchadaphiseksomphot Endowment Fund and The Second Century Fund is also acknowledged.

References

[1] Do B, Ohsaki M, Yamakawa M. (2021) Sequential mixture of Gaussian processes and saddlepoint approximation for reliability-based design optimization of structures. *Structural & Multidisciplinary Optimization*, 1-24.

[2] Aoues Y, Chateauneuf A. (2010) Benchmark study of numerical methods for reliability-based design optimization. *Structural & Multidisciplinary Optimization* **41**, 277-94.

[3] Moustapha M, Sudret B. (2019) Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework. *Structural & Multidisciplinary Optimization*, 1-20.

[4] Papadrakakis M, Lagaros ND. (2002) Reliability-based structural optimization using neural networks and Monte Carlo simulation. *Computer Methods in Applied Mechanics and Engineering* **191**, 3491-507.

[5] Zhao Y-G, Ono T. (1999) A general procedure for first/second-order reliability method (FORM/SORM). *Structural Safety* **21**, 95-112.
[6] Cheng K, Lu Z. (2020) Structural reliability analysis based on ensemble learning of surrogate models. *Structural Safety* **83**, 101905.
[7] Lehký D, Slowik O, Novák D. (2018) Reliability-based design: Artificial neural networks and double-loop reliability-based optimization approaches. *Advances in Engineering Software* **117**, 123-35.
[8] Jing Z, Chen J, Li X. RBF-GA: (2019) An adaptive radial basis function metamodeling with genetic algorithm for structural reliability analysis. *Reliability Engineering & System Safety* **189**, 42-57.
[9] Liang JJ, Qin AK, Suganthan PN, Baskar S. (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation* **10**, 281-95.
[10] Zhang K, Huang Q, Zhang Y. (2019) Enhancing comprehensive learning particle swarm optimization with local optima topology. *Information Sciences* **471**, 1-18.
[11] Murphy KP. (2012) *Machine learning: a probabilistic perspective*: MIT press.
[12] Rasmussen CE, Williams C. (2006) *Gaussian processes for machine learning*. MIT press Cambridge MA.
[13] Schulz E, Speekenbrink M, Krause A. (2018) A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology* **85**, 1-16.
[14] Bichon BJ, McFarland JM, Mahadevan S. (2011) Efficient surrogate models for reliability analysis of systems with multiple failure modes. *Reliability Engineering & System Safety* **96**, 1386-95.
[15] Zhao W, Qiu Z. (2013) An efficient response surface method and its application to structural reliability and reliability-based optimization. *Finite Elements in Analysis Design* **67**, 34-42.
[16] Ghasemi MR, Camp CV, Dizangian B. (2019) Novel decoupled framework for reliability-based design optimization of structures using a robust shifting technique. *Frontiers of Structural and Civil Engineering* **13**, 800-20.