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

A parallel Bayesian interval optimization approach for static analysis of structures with interval uncertainties

Chen Ding
PhD Student, Institute for Risk and Reliability, Leibniz Universität Hannover, Hannover, Germany

Chao Dang
PhD Student, Institute for Risk and Reliability, Leibniz Universität Hannover, Hannover, Germany

Matteo Broggi
Deputy Head, Institute for Risk and Reliability, Leibniz Universität Hannover, Hannover, Germany

Michael Beer
Professor, Institute for Risk and Reliability, Leibniz Universität Hannover, Hannover, Germany

ABSTRACT: This paper aims at approximating the bounds of the static response of structures with interval uncertainties. Such task is often challenging due to the large number of computationally intensive response evaluations required. To address this challenge, we propose an efficient non-intrusive method, namely, parallel Bayesian interval optimization (PBIO). The PBIO first assumes a Gaussian process (GP) prior over the response function. Such a prior can be updated to a posterior GP given observations arising from evaluating the response function at some locations. The main contribution lies in proposing a two-stage infill sampling strategy to guide the selection of multiple update points at each iteration. Briefly, the first stage is to search for promising points that bring significant improvements to the current minimum and maximum responses, using a new acquisition function called Average expected improvement (AvEI) and an improved multi-modal optimization (MMO) algorithm. The second stage is to find additional points that still have significant improvements to the current minimum or maximum response, based on traditional EI and improved MMO. By using PBIO, multiple response values can be evaluated in parallel, and both lower and upper response bounds can be obtained simultaneously in a single run. A static finite element example is studied to show the effectiveness of PBIO.

1. INTRODUCTION

Structural static analysis is essential in the engineering field, as it enables engineers to analyze structures under different loading conditions. Traditionally, a deterministic structural static analysis is performed, where a well-defined computational model with prescribed parameters is involved. Nevertheless, such deterministic analysis is not suitable for practical engineering situations since uncertainty is ubiquitous. Alternatively, non-deterministic structural static analysis can be adopted. In the state of lack of knowledge, the description of uncertainty is imprecise. When only bounds of uncertain parameters are available, uncertainty can be described by an interval. Under this concept, a nondeterministic parameter is treated as an interval variable whose value is taken within given lower and upper values. The main task to quantify the output uncertainty is to find the lower and upper bounds on the static structural response, providing information about the range of possible outcomes, namely the best and worst cases.

This best/worst case search can be transformed into an optimization problem where the objective is

to find the global minimum and maximum of the response. In this regard, traditional global optimization methods, such as genetic algorithm, can be adopted. However, these methods require a large number of response function calls, which is quite computationally expensive because the evaluation of response function usually involves time-consuming structural static finite element analysis. To alleviate the computational burden, the response function can be treated during the optimization process with a Bayesian model, e.g., Gaussian process (GP) model. Along this line, Bayesian global optimization (BGO) (Jones et al. (1998)) can be applied. The basic idea of BGO is to assume a GP-prior over the response function based on some observations, and then update the GP-prior by sequentially selecting update points in terms of an infill sampling strategy. Existing BGO methods for interval uncertainty propagation (De Munck et al. (2009); Liu et al. (2019)) mainly focus on developing efficient infill sampling strategies to reduce the number of response function calls. Nevertheless, most strategies only allow to choose one update point in each iteration and obtain the global minimum and maximum responses separately. Some parallel BGO methods (Dang et al. (2022)) enable to address this issue, however, they may still encounter problems related to the selection of update points. On the one hand, the location of selected update points may be too close to their neighbors, causing unnecessary computational wastes. On the other hand, too many update points can be added at one iteration, some of which may only have little contribution to the GP model refinement.

In this work, a novel parallel BGO method, named parallel Bayesian interval optimization (PBIO), is proposed for estimating the response bounds of structures with interval uncertainties. To reduce the simulation time of existing BGO, a two-stage infill sampling strategy that allows parallelism is developed. The proposed strategy enables to select a batch of diverse and informative points in each iteration, and obtain the lower and upper response bounds simultaneously in a single run. The effectiveness of PBIO is verified by a static finite element example.

2. PROBLEM STATEMENT

In the context of finite element method, a general governing function for structural static analysis can be expressed as

$$\boldsymbol{K}(\boldsymbol{u})\,\boldsymbol{u} = \boldsymbol{F}, \tag{1}$$

where $\boldsymbol{K}(\boldsymbol{u})$ is an $n_u \times n_u$ stiffness matrix for nonlinear case, which can also be $\boldsymbol{K}(\boldsymbol{u}) = \boldsymbol{K}$ for linear case; $\boldsymbol{u}$ is a n_u-dimensional static displacement vector; $\boldsymbol{F}$ is a n_u-dimensional external load vector; and n_u is the degrees of freedom of the structure. Eq. (1) is usually solved with all involved parameters precisely given. However, such deterministic analysis is not suitable for practical engineering problems where uncertainties need to be considered. For the case of lack of knowledge, the input parameters of Eq. (1), such as material properties and external loads, are subject to pure epistemic uncertainty. In this paper, such epistemic uncertainty is modeled by intervals. Denote the input uncertain parameter vector as $\boldsymbol{x}^I = \left[x_1^I, x_2^I, ..., x_{n_s}^I\right]$, we have

$$\boldsymbol{x}^I = [\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}] = \left\{\boldsymbol{x} \in \mathbb{R}^{n_s} \,|\, \underline{\boldsymbol{x}} \leqslant \boldsymbol{x} \leqslant \overline{\boldsymbol{x}}\right\}, \tag{2}$$

where $\underline{\boldsymbol{x}} = \left[\underline{x}_1, \underline{x}_2, ..., \underline{x}_{n_s}\right]$ and $\overline{\boldsymbol{x}} = [\overline{x}_1, \overline{x}_2, ..., \overline{x}_{n_s}]$ are the lower and upper bounds of $\boldsymbol{x}^I$, respectively; and n_s is the number of variables in $\boldsymbol{x}^I$. Then the governing equation can be rewritten as

$$\boldsymbol{K}\left(\boldsymbol{x}^I\right)\boldsymbol{u}\left(\boldsymbol{x}^I\right) = \boldsymbol{F}\left(\boldsymbol{x}^I\right). \tag{3}$$

By solving Eq. (3), any static response of interest, such as stress and strain at a critical location, can be derived from its relationship with $\boldsymbol{u}$. Let us consider a response function written as

$$y^I = f\left(\boldsymbol{u}\left(\boldsymbol{x}^I\right)\right) = g\left(\boldsymbol{x}^I\right), \tag{4}$$

where $y^I = \left[\underline{y}, \overline{y}\right] = \left\{y \in \mathbb{R} \,\middle|\, \underline{y} \leqslant y \leqslant \overline{y}\right\}$ denotes the response of concern, which is one-dimensional and is also an interval valued with lower and upper bounds $\underline{y}$ and $\overline{y}$; $f(\cdot)$ is the function describing the relationship between y^I and $\boldsymbol{u}\left(\boldsymbol{x}^I\right)$; $g(\cdot)$ represents the mapping from $\boldsymbol{x}^I$ to y^I. To evaluate the effect of uncertainty over y^I, the main objective is to obtain $\underline{y}$ and $\overline{y}$, which can be regarded as the solutions of following optimization problems:

$$\underline{y} = \min\left\{y \,|\, y = g(\boldsymbol{x}), \underline{\boldsymbol{x}} \leqslant \boldsymbol{x} \leqslant \overline{\boldsymbol{x}}\right\}, \tag{5}$$

$$\bar{y} = \max\left\{y \,|y = g\left(\boldsymbol{x}\right), \underline{\boldsymbol{x}} \leqslant \boldsymbol{x} \leqslant \bar{\boldsymbol{x}}\right\}. \tag{6}$$

Although the definitions are simple, for most cases, analytical solutions for $\underline{y}$ and $\bar{y}$ are difficult to acquire since the response function (i.e., Eq. (4)) is usually implicit and expensive to evaluate. Alternatively, we can resort to numerical approximation methods that estimate $\underline{y}$ and $\bar{y}$. However, existing numerical approximation methods are usually unable to strike the balance between estimation accuracy and computational efficiency. Hence, a novel method for evaluating Eqs. (5) and (6) is proposed, which is given in detail in the following section. For notational simplicity, the superscripts of $\boldsymbol{x}^I$ and y^I are omitted when there is no confusion.

3. Proposed method

3.1. Gaussian process model

Following the Bayesian approach, a prior belief about the response function can be modeled by a stochastic process model. Commonly, the Gaussian process (GP) model can be applied for the purpose. In this manner, the prior belief about the response function can be described as $\hat{y}_0 = \hat{g}_0\left(\boldsymbol{x}\right) \sim GP\left(\mu_0\left(\boldsymbol{x}\right), \kappa_0\left(\boldsymbol{x},\boldsymbol{x}'\right)\right)$, where $\mu_0\left(\boldsymbol{x}\right)$ and $\kappa_0\left(\boldsymbol{x},\boldsymbol{x}'\right)$ are the prior mean and covariance function, respectively. According to Williams and Rasmussen (2006), there are various kinds of prior mean and covariance functions. Here, we consider the mean to be a constant such that $\mu_0\left(\boldsymbol{x}\right) = \tilde{\mu} \in \mathbb{R}$, and the covariance function takes the squared exponential form as

$$\kappa_0\left(\boldsymbol{x},\boldsymbol{x}'\right) = \sigma_0^2 \exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{x}'\right)\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{x}'\right)^{\mathrm{T}}\right), \tag{7}$$

in which $\sigma_0^2 > 0$ is the overall variance; $\boldsymbol{\Sigma} = \mathrm{diag}\left(l_1^2, l_2^2, ..., l_{n_s}^2\right)$ is a diagonal matrix and $l_i > 0, i = 1, ..., n_s$ is the characteristic length scale in the i-th dimension. Here, a total of $n_s + 2$ unknown hyperparameters $\boldsymbol{\psi} = \{\tilde{\mu}, \sigma_0, l_1, ..., l_{n_s}\}$ are involved.

Suppose we have $N \in \mathbb{Z}^+$ observations that are collected in a training dataset $\mathscr{D} = \{\boldsymbol{X}, \boldsymbol{y}\} = \left\{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(N)}, y^{(1)}, ..., y^{(N)}\right\}$, where $y^{(i)} = g\left(\boldsymbol{x}^{(i)}\right)$. Based on $\mathscr{D}$, the hyperparameters can be evaluated by maximizing the log marginal likelihood function as:

$$\boldsymbol{\psi}^{\bigstar} = \arg\max_{\boldsymbol{\psi}} \left(\log\left(p\left(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\psi}\right)\right)\right), \tag{8}$$

with

$$\log\left(p\left(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\psi}\right)\right) = -\tfrac{1}{2}\left(\boldsymbol{y}-\tilde{\mu}\right)^{\mathrm{T}}\boldsymbol{K}_0^{-1}\left(\boldsymbol{y}-\tilde{\mu}\right) - \tfrac{1}{2}\log\left(|\boldsymbol{K}_0|\right) - \tfrac{N}{2}\log\left(2\pi\right), \tag{9}$$

where $\boldsymbol{K}_0$ is an $N \times N$ covariance matrix with its (i,j)-th element as $\kappa_0\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right)$.

Once $\boldsymbol{\psi}$ is determined, a posterior distribution of response function, denoted as $y_n = g_n\left(\boldsymbol{x}\right)$, can be acquired by conditioning the GP-prior on $\mathscr{D}$. This distribution still follows a GP, i.e., $\hat{y}_n = \hat{g}_n\left(\boldsymbol{x}\right) \sim \mathrm{GP}\left(\mu_n\left(\boldsymbol{x}\right), \kappa_n\left(\boldsymbol{x},\boldsymbol{x}'\right)\right)$. The posterior mean $\mu_n\left(\boldsymbol{x}\right)$ and covariance function $\kappa_n\left(\boldsymbol{x},\boldsymbol{x}'\right)$ are derived in a closed form:

$$\mu_n\left(\boldsymbol{x}\right) = \mu_0\left(\boldsymbol{x}\right) + \boldsymbol{\kappa}_0\left(\boldsymbol{x},\boldsymbol{X}\right)\boldsymbol{K}_0{}^{-1}\left(\boldsymbol{y} - \mu_0\left(\boldsymbol{X}\right)\right), \tag{10}$$

$$\kappa_n\left(\boldsymbol{x},\boldsymbol{x}'\right) = \kappa_0\left(\boldsymbol{x},\boldsymbol{x}'\right) - \boldsymbol{\kappa}_0\left(\boldsymbol{x},\boldsymbol{X}\right)\boldsymbol{K}_0{}^{-1}\boldsymbol{\kappa}_0\left(\boldsymbol{x}',\boldsymbol{X}\right)^{\mathrm{T}}, \tag{11}$$

in which $\boldsymbol{\kappa}_0\left(\boldsymbol{x},\boldsymbol{X}\right)$ is an $1 \times N$ covariance vector describing the dependency between $\boldsymbol{x}$ and $\boldsymbol{X}$, whose i-th component is $\boldsymbol{\kappa}_0\left(\boldsymbol{x},\boldsymbol{x}^{(i)}\right)$; $\boldsymbol{\kappa}_0\left(\boldsymbol{x}',\boldsymbol{X}\right)$ is similarly defined; $\mu_0\left(\boldsymbol{X}\right)$ is an $N \times 1$ mean vector with i-th element as $\mu_0\left(\boldsymbol{x}^i\right)$. In this manner, a full predictive distribution at a new observation $\boldsymbol{x}$ follows the Gaussian distribution such that $\hat{y}_n = \hat{g}_n\left(\boldsymbol{x}\right) \sim \mathrm{N}\left(\mu_n\left(\boldsymbol{x}\right), \sigma_n^2\left(\boldsymbol{x}\right)\right)$, where $\mu_n\left(\boldsymbol{x}\right)$ can be regarded as the predictor and $\sigma_n^2\left(\boldsymbol{x}\right) = \kappa_n\left(\boldsymbol{x},\boldsymbol{x}\right)$ measures the prediction uncertainty.

3.2. Proposed two-stage infill sampling strategy

In order to find the lower and upper response bounds with fewer response function calls, a novel two-stage infill sampling strategy is developed here for GP model refinement. First, multiple update points associated with the minimum and maximum responses are selected at each iteration based on a newly developed acquisition function and an improved multi-modal optimization algorithm. In this regard, better approximations of the bounds are obtained simultaneously. Subsequently, additional update points related to the minimum or maximum response are selected to avoid possible local convergence. The proposed strategy is described in detail below.

3.2.1. First stage: averaged EI

Following the traditional expected improvement (EI) (Jones et al. (1998)), a new acquisition function is defined to guide the selection of update points associated with improving both the current minimum and maximum responses.

Let $y_{\min} = \min\left\{y^{(1)}, y^{(2)}, ..., y^{(N)}\right\}$ be the current minimum value of response in $\mathscr{D}$. The improvement at a new point $\boldsymbol{x}$ over $y_{\min}$ can be measured as $I_{\min}(\boldsymbol{x}) = \max(y_{\min} - \hat{g}_n(\boldsymbol{x}), 0)$. The traditional EI is defined as the expected value of $I_{\min}(\boldsymbol{x})$ conditioning on the current training dataset $\mathscr{D}$, which has the analytical expression as

$$\begin{aligned}\mathscr{L}^{\mathrm{EI}}_{\min}(\boldsymbol{x}) &= E_{\mathscr{D}}\left[I_{\min}(\boldsymbol{x})\right] \\ &= (y_{\min} - \mu_n(\boldsymbol{x}))\,\Phi\left(\tfrac{y_{\min}-\mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}\right) \\ &\quad + \sigma_n(\boldsymbol{x})\,\phi\left(\tfrac{y_{\min}-\mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}\right),\end{aligned} \tag{12}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ represent the probability density function and cumulative distribution function of the standard Gaussian distribution, respectively. The promising point to be added in $\mathscr{D}$ is selected by maximizing the EI over $y_{\min}$:

$$\boldsymbol{x}^{+}_{\min} = \arg\max_{\boldsymbol{x}\in[\underline{\boldsymbol{x}},\overline{\boldsymbol{x}}]} \mathscr{L}^{\mathrm{EI}}_{\min}(\boldsymbol{x}). \tag{13}$$

Note that maximizing the first term in Eq. (12) reflects the preference for point $\boldsymbol{x}$ with response value smaller than $y_{\min}$, while maximizing the second term in Eq. (12) prefers the point $\boldsymbol{x}$ with a larger prediction uncertainty.

Similarly, let $y_{\max} = \max\left\{y^{(1)}, y^{(2)}, ..., y^{(N)}\right\}$ be the current maximum response among $\mathscr{D}$. The EI over current $y_{\max}$ can be defined as

$$\begin{aligned}\mathscr{L}^{\mathrm{EI}}_{\max}(\boldsymbol{x}) &= (\mu_n(\boldsymbol{x}) - y_{\max})\,\Phi\left(\tfrac{\mu_n(\boldsymbol{x})-y_{\max}}{\sigma_n(\boldsymbol{x})}\right) \\ &\quad + \sigma_n(\boldsymbol{x})\,\phi\left(\tfrac{\mu_n(\boldsymbol{x})-y_{\max}}{\sigma_n(\boldsymbol{x})}\right).\end{aligned} \tag{14}$$

The promising update point is selected by the following optimization:

$$\boldsymbol{x}^{+}_{\max} = \arg\max_{\boldsymbol{x}\in[\underline{\boldsymbol{x}},\overline{\boldsymbol{x}}]} \mathscr{L}^{\mathrm{EI}}_{\max}(\boldsymbol{x}). \tag{15}$$

Note that the traditional EI selects only one update point at each iteration, which does not allow to evaluate the response value in parallel. Moreover, the lower and upper response bounds can only be obtained in two separate optimization schemes, namely, Eqs. (13) and (15). In this regard, the traditional EI has limited computational efficiency in estimating the response bounds.

To overcome the limitations of traditional EI, a basic idea is to choose multiple update points that contribute to the improvement of both the lower and upper bounds of the response in a single iteration, so as to find better approximations of the minimum and maximum responses simultaneously. A simple way to do this is to form an acquisition function that considers the improvements to the minimum and maximum estimated responses in the current iteration. Since the amount of improvement over $y_{\min}$ or $y_{\max}$ in each iteration is not known in advance, the improvements to $y_{\min}$ and $y_{\max}$ are treated equally in the acquisition function. Hence, an averaged EI (AvEI) function is formulated, which is defined as

$$\mathscr{L}^{\mathrm{AvEI}}(\boldsymbol{x}) = \frac{1}{2}\mathscr{L}^{\mathrm{EI}}_{\min}(\boldsymbol{x}) + \frac{1}{2}\mathscr{L}^{\mathrm{EI}}_{\max}(\boldsymbol{x}). \tag{16}$$

In this manner, the update point to be added in $\mathscr{D}$ can be found by maximizing the proposed AvEI function over the input range:

$$\boldsymbol{x}^{+} = \arg\max_{\boldsymbol{x}\in[\underline{\boldsymbol{x}},\overline{\boldsymbol{x}}]} \mathscr{L}^{\mathrm{AvEI}}(\boldsymbol{x}). \tag{17}$$

Note that to capture multiple update points associated with both the minimum and maximum responses based on the AvEI function, a multi-modal optimization (MMOP) algorithm is required. Such algorithm captures the local maxima of the AvEI function and therefore allows the selection of update points at each AvEI maximum. Recently, a series of MMOP algorithms have been developed. Among these algorithms, it has been found that the Evolutionary Multi-Modal Optimization based Multi-Objective Optimization (EMO-MMO) (Cheng et al. (2017)) is more advantageous due to its applicability to infinite optimal set and its robust performance in benchmark comparisons. Hence, the EMO-MMO is used here to select update points. However, since the EMO-MMO was developed for capturing multiple global optima rather than local optima, an improved EMO-MMO that finds local maxima is proposed in the following section.

3.2.2. Improved multi-modal optimization algorithm

A general procedure of the EMO-MMO (Cheng et al. (2017)) applied to find the maxima of the AvEI function is briefly summarized below.

To begin with, a multiobjective fitness landscape approximation (MOFLA) is developed to obtain the approximate fitness landscapes that reflects the shape characteristics of the AvEI function (Cheng et al. (2017)). In the MOFLA, the MMO is first transformed to a multi-objective optimization (MOP) with the first optimization objective as the AvEI function and the second optimization objective being a grid-based diversity indicator. Here, the widely used Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al. (2002)) is utilized to perform the transformed MOP. The candidate solutions created in each generation during MOP are stored in a set $\mathscr{D}_{\text{land}} = \{X_{\text{land}}, Y_{\text{land}}\}$ as the approximate fitness landscape, where $X_{\text{land}} = \left\{\boldsymbol{x}_{\text{land}}^{(1)}, \boldsymbol{x}_{\text{land}}^{(2)}, \ldots\right\}$ and $Y_{\text{land}} = \mathscr{L}^{\text{AvEI}}(X_{\text{land}})$.

Then, based on $\mathscr{D}_{\text{land}}$, the possible regions of all local AvEI maxima can be located by identifying the peaks of the fitness landscape. Such peak identification is performed by using a binary cutting-based adaptive peak detection (BC-APD) technique (Cheng et al. (2017)). The basic idea is to make a cut at the top of $\mathscr{D}_{\text{land}}$ to form a gap that disconnects the peaks from each other, so that the peaks can be identified based on the gaps. The BC-APD first makes an initial cut at the top of $\mathscr{D}_{\text{land}}$, using a user-defined cutting ratio $\gamma \in (0,1)$ to generate a cutting slice $\mathscr{D}_c = \{X_c, Y_c\}$, where $X_c = \left\{\boldsymbol{x}_c^{(1)}, \boldsymbol{x}_c^{(2)}, \ldots\right\}$ and $Y_c = \mathscr{L}^{\text{AvEI}}(X_c)$. Note that in order to identify the peak sets that include the local maxima of the AvEI function, the initial γ is set to be 0.9. After that, each peak set within $\mathscr{D}_{\text{c}}$ is identified successively by checking the neighboring distances between each point in X_{c}. Here, an adaptive threshold is required for each peak set to determine whether neighboring points belong to the same peak, which is defined as

$$\eta = \max_i \left\{ \min_{j \neq i} \left\| \boldsymbol{x}_c^{(i)} - \boldsymbol{x}_c^{(j)} \right\|_1 \right\} \tag{18}$$

The neighboring points having distances smaller than η will be regarded as belong to the same peak. Afterwards, binary cuttings with $\gamma = 0.5$ are successively operated based on $\mathscr{D}_c$ until all points in $\mathscr{D}_c$ have been assigned to a corresponding peak.

Subsequently, on the basis of the detected peak sets and $\mathscr{D}_{\text{land}}$, local search is implemented inside each peak set to obtain the final result of all local AvEI maxima. Here, a competitive swarm optimizer (CSO) (Cheng and Jin (2014)) is directly applied to independently search for the maximum within each peak.

However, the EMO-MMO may have some problems when searching for local maxima. First, the EMO-MMO sometimes obtains points with lower AvEI function values around the local AvEI maxima. This is due to the fact that some of the peak sets detected by the BC-APD may be "fake" peak sets, i.e., not the actual peak sets of the fitness landscape. The fitness landscape points contained in these "fake" peak sets may be a set of fitness landscape "outliers" that have their neighboring distances slightly greater than the threshold defined in Eq. (18) but should still belong to the same peak set of neighboring points. In this case, these "outliers" will be identified as belonging to a new and "fake" peak set rather than to the actual peak set of fitness landscape. Second, the AvEI function in a single iteration may have many local peaks, resulting in many update points being added. Some of these update points may have very low AvEI function values, indicating negligible contributions to GP model refinement. If this is the case, adding such update points would result in unnecessary computational waste. Hence, to improve computational efficiency, it is desired to select update points that contribute significantly to GP model refinement.

In this regard, we propose two schemes to improve the EMO-MMO. First, a peak checking scheme is developed to remove the "fake" peak sets. The idea is to generate a set of equidistant additional points around the point $\boldsymbol{x}$ corresponding to the maximum AvEI function value of each peak set, and then remove the "fake" peak set whose maximum function value is lower than one of the function value of the additional points. For convenience, we propose to first project the input space

to a unit hypercube space. The additional points for each peak set can be then generated by a grid-based scheme based on the transformed space. Specifically, we pick 8 equidistant additional points along the boundary of an equidistant grid that is formed around the point $\boldsymbol{x}$ corresponding to the maximum value of AvEI function for each peak set. Take a two-dimensional input as an example. The equidistant additional points distributed around the point corresponding to the maximum function value of the detected peak set are shown in Figure 1. Here, the distance D_{dist} between the point related to maximum and each additional point takes 0.0002. By comparing the AvEI function values of these 9 points, the "fake" peak set can be identified and then removed.

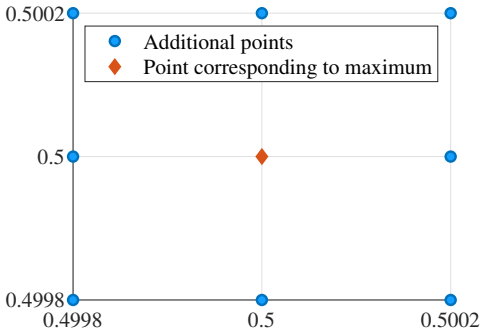

Figure 1: Additional points for "fake" peak set checking

Second, to reduce the number of possibly excessive update points, a cluster-based scheme is proposed to only select update points that contribute significantly to GP model refinement in one iteration. Here, the density-based spatial clustering of applications with noise (DBSCAN) (Ester et al. (1996)) is adopted to cluster the update points. Since the input space is transformed into a unit hypercube space and there is only one maximum value per peak set, the neighborhood search radius is set to $\zeta = 0.1$ and the minimum number of neighborhoods n_{minpts} required to identify the core point is set to 1. In each cluster, among all included update points, the update point with the largest AvEI function value is selected as the update point to be added in $\mathscr{D}$. In addition, the maximum number of update points n^+ is required to be no greater than a user-defined number $q_{\max}$. Therefore, only the first $q_{\max}$ update points with larger AvEI function values are finally added to the training dataset $\mathscr{D}$, i.e., $\boldsymbol{x}^+ = \left\{\boldsymbol{x}^{(1),+}, ..., \boldsymbol{x}^{(n^+),+}\right\}, n^+ \leqslant q_{\max}$.

3.2.3. *Stopping criteria and second stage*

Based on the proposed AvEI function and improved MMO algorithm, the selected update points will be successively added to $\mathscr{D}$ until a stopping criterion is satisfied. Considering that the magnitude of the response function may be quite large, a normalized stopping criterion is applied:

$$\frac{\max_{\boldsymbol{x}} \mathscr{L}^{\text{AvEI}}(\boldsymbol{x})}{\max\left\{y^{(1)}, ..., y^{(N)}\right\} - \min\left\{y^{(1)}, ..., y^{(N)}\right\}} < \varepsilon_1, \tag{19}$$

where ε_1 is a user-defined small convergence threshold, which can be specified as $10^{-3} - 10^{-2}$. To accommodate stochastic evaluations, a delayed judgment is adopted such that the above stopping criterion is required to be satisfied for several times (e.g., three times).

It is found that sometimes the first stage may converge too quickly to accurately obtain the global maximum and minimum of y. To avoid this, a second stage based on $\mathscr{L}^{\text{EI}}_{\min}(\boldsymbol{x})$ and $\mathscr{L}^{\text{EI}}_{\max}(\boldsymbol{x})$ is performed after Eq. (19) is satisfied. Two more stopping criteria are checked:

$$\frac{\max_{\boldsymbol{x}} \mathscr{L}^{\text{EI}}_{\min}(\boldsymbol{x})}{\max\left\{y^{(1)}, ..., y^{(N)}\right\} - \min\left\{y^{(1)}, ..., y^{(N)}\right\}} < \varepsilon_2, \tag{20}$$

and

$$\frac{\max_{\boldsymbol{x}} \mathscr{L}^{\text{EI}}_{\max}(\boldsymbol{x})}{\max\left\{y^{(1)}, ..., y^{(N)}\right\} - \min\left\{y^{(1)}, ..., y^{(N)}\right\}} < \varepsilon_3, \tag{21}$$

where ε_2 and ε_3 are two small thresholds specified by users. If either of Eqs. (20) and (21) is not satisfied, then it is necessary to add a set of update points to $\mathscr{D}$ using the improved MMO algorithm according to either Eq.(13) or Eq.(15).

3.3. *Proposed algorithm*

Based on the GP model and proposed two-stage infill sampling strategy, a novel parallel Bayesian interval optimization (PBIO) method is proposed here for estimating the lower and upper bounds of y. The implementation procedure of the PBIO is provided as follows:

Step 1: Initialization. According to the response function $y = g(\boldsymbol{x})$ and the inputs $\boldsymbol{x}^I$, define the minimization and maximization optimization problems as Eqs. (5) and (6). Specify the initial sample size N_{ini} (e.g., $N_{\text{ini}} = 10$), and then generate the corresponding training dataset $\mathscr{D} = \{\boldsymbol{X}, \boldsymbol{y}\} = \left\{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(N_{\text{ini}})}, y^{(1)}, ..., y^{(N_{\text{ini}})}\right\}$ of size N_{ini} using Latin hypercube sampling over $\boldsymbol{x}^I$. Set $N = N_{\text{ini}}$.

Step 2: Train a GP model. Construct a GP model $\hat{y}_n = \hat{g}_n(\boldsymbol{x}) \sim \text{GP}(\mu_n(\boldsymbol{x}), \kappa_n(\boldsymbol{x}, \boldsymbol{x}'))$ based on $\mathscr{D}$. The involved hyperparameters can be evaluated by Eq. (8).

Step 3: Check the stopping criterion given by Eq. (19)), where the threshold ε_1 can be any small value, such as 0.02. If Eq. (19) is satisfied for three times consecutively, then go to **Step 5**; otherwise, go to **Step 4**.

Step 4: Identify update points in the first stage. A set of update points $\boldsymbol{x}^+ = \left(\boldsymbol{x}^{(1),+}, ..., \boldsymbol{x}^{(n^+),+}\right)$ are identified by Eq. (17) using the improved MMO algorithm, where $n^+ \leqslant q_{\max}$ and $q_{\max}$ can be any positive integer. Then, go to **Step 7**.

Step 5: Check the two stopping criteria given by Eqs. (20) and (21), where the thresholds $\varepsilon_2 = \varepsilon_3 = \varepsilon_1$ for convenience. If both Eqs. (20) and (21) are satisfied, then go to **Step 8**; otherwise, go to **Step 6**.

Step 6: Identify update points in the second stage. A set of update points are identified by either Eq. (13) or Eq. (15) using the improved MMO algorithm according to the unsatisfied local stopping criteria. Then, go to **Step 7**.

Step 7: Enrich the training dataset. Evaluate the response over the identified update points in parallel. And then, add both the identified update points and corresponding responses into $\mathscr{D}$ such that $\mathscr{D} = \mathscr{D} \cup \mathscr{D}_{\text{add}}$, where $\mathscr{D}_{\text{add}} = \{\boldsymbol{x}^+, g(\boldsymbol{x}^+)\}$ and $\boldsymbol{x}^+$ with size n^+ include update points from both first and second stages. Set $N = N + n^+$ and then return to **Step 2**.

Step 8: Record results and end the algorithm. Record $\underline{y} = \min_{1 \leqslant i \leqslant N} g\left(\boldsymbol{x}^{(i)}\right)$ and $\overline{y} = \max_{1 \leqslant i \leqslant N} g\left(\boldsymbol{x}^{(i)}\right)$ as the estimates of lower and upper response bounds, and then end the algorithm.

4. CASE STUDY

To demonstrate the effectiveness of the proposed PBIO, a static finite element example involving a 3D elastic cantilever beam subjected to stress (van Alem (2022)) is studied here. The geometry of the cantilever beam is shown in Figure 2, where the F5 surface is supported and the F2 surface is under stress loading. The structural response of interest is the maximum vertical displacement of the beam, which is evaluated by using the Partial Differential Equation (PDE) toolbox in MATLAB software. Six mutually independent interval variables are considered here, whose description is listed in Table 1.

The lower and upper response bounds are estimated by the PBIO, particle swarm optimization (PSO) algorithm, vertex method (Dong and Shah (1987)), and non-parallel BGO (N-PBGO) (Jones et al. (1998); Dang et al. (2022)), where their obtained results are given in Table 2. Here, the parameters of PBIO are set as: $N_{\text{ini}} = 10$, $D_{\text{dist}} = 0.0002$, $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 0.02$, and $q_{\max} = 8$. The PSO result is served as the reference result, which is obtained by two independent minimization and maximization optimizations. In addition, the N-PBGO results are evaluated by two separate optimizations using the traditional EI (i.e., Eqs. (12) and (14)) with $N_{\text{ini}} = 10$, whose stopping conditions (i.e., Eqs. (20) and (21)) need to be satisfied twice successively and use the same threshold as ε_1. It is found that although PBIO provides a slightly narrower response interval than the PSO result, PBIO requires only 16 simulations in total and 4 iterations, which is much more efficient than PSO. Besides, the response bounds estimated by vertex method using 8 processors for parallel computation have acceptable accuracy but require more simulations than PBIO. Moreover, N-PBGO performs worse than PBIO, which uses more simulations and iterations to obtain a narrower response interval.

Table 1: Description of interval variables

Variable	Unit	Description	Interval
E	Pa	Young's modulus	$[1.8 \times 10^{11}, 2.2 \times 10^{11}]$
υ	-	Poisson's ratio	$[0.25, 0.35]$
P	Pa	Applied stress	$[1.5 \times 10^{7}, 2.5 \times 10^{7}]$
l	m	Length of beam	$[0.095, 0.105]$
b	m	Width of beam	$[0.0095, 0.0105]$
h	m	Height of beam	$[0.0095, 0.0105]$

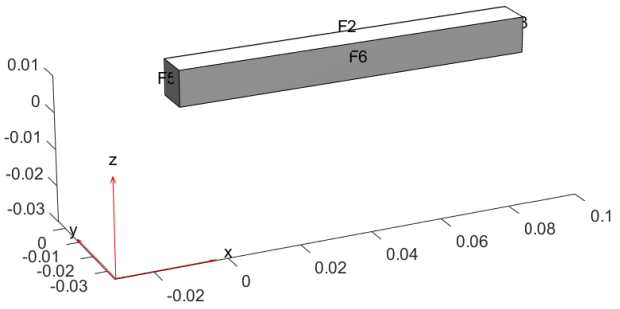

Figure 2: Geometry of the beam

Table 2: Comparison of results by different methods

Method	$\underline{y}$ (mm)	$\overline{y}$ (mm)	N	N_{iter}
PSO	1.9959	7.5448	6420	6420
Vertex	1.9967	7.5425	64	8
N-PBGO	1.9972	7.4408	26	26
PBIO	1.9968	7.5442	16	4

5. CONCLUSIONS

In this paper, a parallel Bayesian interval optimization (PBIO) method is proposed for static analysis of structures with interval uncertainties. A new two-stage infill sampling strategy, including a new acquisition function and an improved MMO algorithm, is developed to guide the optimization search towards the global minimum and maximum of response with improved efficiency. Such strategy enables to add multiple update points at each iteration and obtain both lower and upper response bounds simultaneously in a single run. The feasibility of the proposed PBIO is tested with a static finite element analysis example. The results show that PBIO is able to find the lower and upper response bounds with a reduced computational cost.

6. ACKNOWLEDGMENT

Chen Ding acknowledges the support of the European Union's Horizon 2020 research and innovation programme under Marie Sklodowska-Curie project GREYDIENT - Grant Agreement n°955393.

7. REFERENCES

Cheng, R. and Jin, Y. (2014). "A competitive swarm optimizer for large scale optimization." *IEEE Transactions on Cybernetics*, 45(2), 191–204.

Cheng, R., Li, M., Li, K., and Yao, X. (2017). "Evolutionary multiobjective optimization-based multimodal optimization: Fitness landscape approximation and peak detection." *IEEE Transactions on Evolutionary Computation*, 22(5), 692–706.

Dang, C., Wei, P., Faes, M. G., Valdebenito, M. A., and Beer, M. (2022). "Interval uncertainty propagation by a parallel Bayesian global optimization method." *Applied Mathematical Modelling*, 108, 220–235.

De Munck, M., Moens, D., Desmet, W., and Vandepitte, D. (2009). "An efficient response surface based optimisation method for non-deterministic harmonic and transient dynamic analysis." *CMES-Computer Modeling in Engineering & Sciences*, 47(2), 119–166.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.

Dong, W. and Shah, H. C. (1987). "Vertex method for computing functions of fuzzy variables." *Fuzzy Sets and Systems*, 24(1), 65–78.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise." *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, Vol. 96, 226–231.

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

Liu, Y., Wang, X., Wang, L., and Lv, Z. (2019). "A Bayesian collocation method for static analysis of structures with unknown-but-bounded uncertainties." *Computer Methods in Applied Mechanics and Engineering*, 346, 727–745.

van Alem, P. (2022). "Introduction to FEM". `https://www.mathworks.com/matlabcentral/fileexchange/100766-professional_plots` [Online; accessed November 22, 2022].

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian Processes for Machine Learning*, Vol. 2. MIT press Cambridge, MA.