Parallelization of Adaptive Bayesian Cubature Using Multimodal Optimization Algorithms

Fangqi Hong, Pengfei Wei *

School of Power and Energy, Northwestern Polytechnical University, Xian, China

Michael Beer

Institute for Risk and Reliability, Leibniz Universitt Hannover, Hannover, Germany
Institute for Risk and Uncertainty, University of Liverpool, Liverpool, UK
International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai, China

Abstract

Purpose–Bayesian cubature (BC) has emerged to be one of most competitive approach for estimating the multi-dimensional integral especially when the integrand is expensive to evaluate, and alternative acquisition functions, such as the Posterior Variance Contribution (PVC) function, have been developed for adaptive experiment design of the integration points. However, those sequential design strategies also prevent BC from being implemented in a parallel scheme. Therefore, this paper aims at developing a parallelized adaptive BC method to further improve the computational efficiency.
Design/methodology/approach–By theoretically examining the multimodal behavior of the PVC function, it is concluded that the multiple local maxima all have important contribution to the integration accuracy as can be selected as design points, providing a practical way for parallelization of the adaptive BC. Inspired by the above finding, four multimodal optimization algorithms, including one newly developed in this work, are then introduced for finding multiple local maxima of the PVC function in one run, and further for parallel implementation of the adaptive BC.
Findings–The superiority of the parallel schemes and the performance of the four multimodal optimization algorithms are then demonstrated and compared with the k-means clustering method by using two numerical benchmarks and two engineering examples.
Originality–Multimodal behavior of acquisition function for BC is comprehensively investigated. All the local maxima of the acquisition function contribute to adaptive BC accuracy. Parallelization of adaptive BC is realized with four multimodal optimization methods.
Keywords Bayesian Cubature, Adaptive Experiment Design, Acquisition Function, Parallel Computation, Multimodal Optimization
Paper type Research paper

1. Introduction

Scientific computing plays a more and more important role in almost all aspects of science, technology, engineering, and mathematics, and the development of ac-

*Corresponding author: pengfeiwei@nwpu.edu.cn curate, robust, and efficient algorithms for solving different types of problems, e.g., multi-dimensional integrals, nonlinear equations, and partial differential equations (PDEs), is far from being sufficient due to the high demands for large-scale simulation. Unquestionably, the deterministic numerical methods, such as Gaussian-Hermite quadrature and finite element PDE solver, dominated scientific research for a long history and will continue to play a leading role in years or even decades to come (Trangenstein (2018)). However, the emergence of probabilistic methods (e.g., Bayesian cubature (BC) (Briol (2018)), Bayesian optimization (Huang *et al.,* (2006); Snoek *et al.,* (2012)), Bayesian ODE/PDE solver (Tronarp *et al.,* (2019)), and Bayesian reliability analysis (Song *et al.,* (2021))), which treat the numerical analysis tasks as statistical inference problems and then solve them with a Bayesian inference scheme, seems to begin to challenge the above mainstream (Hennig *et al.,* (2015); Ghosh *et al.,* (2020)). Their most appealing feature against the deterministic methods is that the numerical errors are regarded as a kind of epistemic uncertainty and then quantified by subjective probability distribution (Hennig *et al.,* (2015); Cockayne *et al.,* (2019)). A historic investigation of these developments was given by Oates and Sullivan (2019). Despite the appealing academic idea, the development of probabilistic numerical methods is still at the early stage, and many challenges need to be fixed. The concern of this work is on the computational efficiency of Bayesian cubature (BC), which aims at computing multi-dimensional integrals.

Although being new to the engineering computation community, BC can be dated back to 1988 when Diaconis first developed the connection between a Gaussian Process regression (GPR) approximation of the integrand and a classical deterministic cubature rule (Diaconis (1988)). This was later generalized by, e.g., O'Hagan (O'Hagan (1991)), who has shown that a Bayes-Hermite cubature rule can be derived following a similar scheme and performs better than the classical Gauss-Hermite rules in low dimension. A more recent important development is referred to Ghahramani and Rasmussen (2002), who have developed the closed-form formulas for the posterior mean and variance of the integrals under a Gaussian Process prior for the integrand, and named the method as Bayesian Monte Carlo. It is until recently that the rate of posterior contraction is theoretically and systematically studied by Briol et al. (Briol *et al.,* (2019)). Besides the above milestone developments, some other specific problems concerning BC have also been studies. For example, the approximate BC scheme is investigated in Ref. Osborne *et al.,* (2012) for problems with non-negative integrands. Besides, BC has also attracted more and more attention in the engineering communities, for example, for structural reliability analysis (Zhou and Peng (2020)), and for quantification of complex uncertainties (Wei *et al.,* (2021)). Overall, BC has shown to be competitive over the deterministic cubature rules for computing multi-dimensional integrals, and achieved great advancements around the past years.

The experiment design of integration points is another important task for BC. In real-world applications, the integrand is often computationally expensive, thus an optimal BC rule should be able to achieve the required accuracy with a minimal number of integrand calls. The traditional strategy for experiment design is based on Monte Carlo or quasi-Monte Carlo (Ghahramani and Rasmussen (2002)), which is definitely not optimal as the functional behavior of the integrand is not taken into consideration. A more practical strategy is to design the points sequentially to minimize the posterior variance of the integration as it measures the numerical error, and the core is the acquisition function. Following this scheme, Osborne et al. has proposed an active learning procedure for sequential experiment design by minimizing the expected posterior variance of the integration (Osborne *et al.,* (2012)); A closed-form acquisition function, called Posterior Variance Contribution (PVC), has been developed by some of the authors, and shown to be effective as it incorporates the information of both posterior variance and covariance of the GPR model (Wei *et al.,* (2020)); Another acquisition function, which combines the information of the current approximation of the posterior and the position information of the nodes, has been devised by Llorente et al. in Ref. Santiago *et al.,* (2020); The development of an acquisition function for adaptive BC from integrand information generated from multi-source models was conducted by Gessner et al. in Ref. Gessner *et al.,* (2020); An adaptive experiment design strategy is also developed by Sinsbeck et al. for Bayesian model selection (Sinsbeck *et al.,* (2021)). Besides the above development of acquisition functions, the convergence rate of the adaptive BC with alternatives acquisition functions has been studied by Kanagawa and Henning (Kanagawa and Hennig (2019)), which provides rigorous convergence guarantees for this class of methods. For the GPR model, a comprehensive review and comparison of the adaptive experiment design strategies can be found in Ref. Fuhg *et al.,* (2021); while for BC, although there is no study for comparing the performance of these adaptive design strategies, they are all proved to be more efficient than the random sampling scheme in terms of the number of integrand calls. Among the above acquisition functions, the PVC function is of special concern in this work, as it is in closed form, and computationally cheap.

Nowadays, parallel computation has been widely recognized as an effective way for improving the efficiency of scientific computation, and the equipment of numerical algorithms with parallel computation is then of vital importance. However, for adaptive BC, parallel computation is not straightforward as the design points are generated sequentially, and for each iteration, only one design point can be achieved by computing the global optima of an acquisition function. This work aims to fill this gap. The functional behavior of the PVC function is first investigated from both theoretical and practical aspects, and it is concluded that the PVC function is multi-modal when multiple training points are used, and all the local maxima are shown to have great contributions for improving the cubature accuracy. Motivated by this, four multi-modal optimization algorithms, with one developed in this work are introduced for devising parallel computation schemes for the adaptive BC algorithm. The performance of the four algorithms as well as their superiority over the algorithm without parallel computation and the clustering algorithm based parallel method are demonstrated by both numerical and real-world engineering benchmarks.

The rest of this work is organized as follows. Section 2 briefly reviews the adaptive BC, and presents the motivation and basic rationale of this work, followed by the introduction of four multi-modal algorithms, together with the k-means clustering method for comparison for parallel implementation of adaptive BC in section 3. With the above theoretical results and material ready, the parallelized adaptive BC algorithm is then presented in section 4. The benchmark studies are then carried out in section 5. Section 6 concludes this article.

2. Problem statement with insightful interpretation

2.1. *Brief review of Bayesian cubature*

The numerical analysis task of concern in this work is to estimate the n-dimensional integral:

$$d = \int_{\mathbb{R}^n} g\left(\boldsymbol{x}\right) \pi\left(\boldsymbol{x}\right) \mathrm{d}\boldsymbol{x}, \tag{1}$$

where $g\left(\boldsymbol{x}\right)$ indicates the integrand with n-dimensional arguments $\boldsymbol{x}$, and $\pi\left(\boldsymbol{x}\right)$ refers to the weight density. The BPI procedure for numerically estimating the above integral is based on assuming a stochastic process model $\hat{g}\left(\boldsymbol{x}\right)$ for the deterministic integrand $g\left(\boldsymbol{x}\right)$, and this way to impose a probability distribution on the unknown deterministic value d. In this work, the stochastic process model $\hat{g}\left(\boldsymbol{x}\right)$ is assumed to be Gaussian, and then the induced probability distribution of

$$\hat{d} = \int_{\mathbb{R}^n} \hat{g}\left(\boldsymbol{x}\right) \pi\left(\boldsymbol{x}\right) \mathrm{d}\boldsymbol{x} \tag{2}$$

is also Gaussian, resulting from the fact that the integral expressed by equation (2) is a linear projection of the GPR model $\hat{g}\left(\boldsymbol{x}\right)$ (Rasmussen and Ghahramani (2003)).

Let denoted by $\mathcal{GP}\left(m\left(\boldsymbol{x}\right), \kappa\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right)$ the prior GP model for $g\left(\boldsymbol{x}\right)$, where $m\left(\boldsymbol{x}\right)$ is the prior mean which can be assumed to be zero, constant, linear or higher order polynomials, and $\kappa\left(\boldsymbol{x}, \boldsymbol{x}'\right)$ indicates the prior covariance function between any two sites $\boldsymbol{x}$ and $\boldsymbol{x}'$ which can be assumed to be any kernels such as the squared exponential kernel and Matérn kernel (Rasmussen and Williams (2006)). Suppose now there is a set of training data $\mathcal{D} = \left(\mathcal{X}, \mathcal{Y}\right)$, where $\mathcal{X}$ is a sample matrix of dimension $\left(N, n\right)$, with each row indicating a realization of $\boldsymbol{x}$, and $\mathcal{Y}$ refers to a N-dimensional column vector of integrand values, i.e., $\mathcal{Y} = g\left(\mathcal{X}\right)$. The values of all the hyperparameters involved in $m\left(\boldsymbol{x}\right)$ and $\kappa\left(\boldsymbol{x}, \boldsymbol{x}'\right)$ can be easily computed by maximizing the likelihood function derived from $\mathcal{D}$ (see chapter 5 of Ref. Rasmussen and Williams (2006)). The posterior GP model for approximating $g\left(\boldsymbol{x}\right)$ can be obtained as $\hat{g}\left(\boldsymbol{x}\right) \sim \mathcal{GP}\left(\mu_g\left(\boldsymbol{x}\right), \mathrm{cov}_g\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right)$, where $\mu_g\left(\boldsymbol{x}\right)$ and $cov_g\left(\boldsymbol{x}, \boldsymbol{x}'\right)$ respectively indicate the posterior mean and posterior covariance, which are formulated as:

$$\mu_g\left(\boldsymbol{x}\right) = m\left(\boldsymbol{x}\right) + \kappa\left(\boldsymbol{x}, \mathcal{X}\right) \mathcal{K}^{-1}\left(\mathcal{Y} - m\left(\mathcal{X}\right)\right), \tag{3}$$

and

$$\mathrm{cov}_g\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \kappa\left(\boldsymbol{x}, \boldsymbol{x}'\right) - \kappa\left(\boldsymbol{x}, \mathcal{X}\right) \mathcal{K}^{-1} \kappa\left(\mathcal{X}, \boldsymbol{x}'\right), \tag{4}$$

where $\kappa\left(\boldsymbol{x}, \mathcal{X}\right)$ indicates the row vector with the i-th component being the prior covariance between $\boldsymbol{x}$ and the i-th row of $\mathcal{X}$, and $\mathcal{K}$ is the prior covariance of $\mathcal{X}$. The posterior variance $\sigma_g^2\left(\boldsymbol{x}\right)$ is computed by $\sigma_g^2\left(\boldsymbol{x}\right) = \mathrm{cov}_g(\boldsymbol{x}, \boldsymbol{x})$. The posterior mean μ_d and posterior variance σ_d^2 of $\hat{d}$ are then formulated as:

$$\mu_d = \Pi\left[m\left(\boldsymbol{x}\right)\right] + \Pi\left[\kappa\left(\boldsymbol{x}, \mathcal{X}\right)\right] \mathcal{K}^{-1} \left(\mathcal{Y} - m\left(\mathcal{X}\right)\right), \tag{5}$$

and

$$\sigma_d^2 = \Pi\Pi'\left[\kappa\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right] - \Pi\left[\kappa\left(\boldsymbol{x}, \mathcal{X}\right)\right] \mathcal{K}^{-1} \Pi\left[\kappa\left(\mathcal{X}, \boldsymbol{x}'\right)\right], \tag{6}$$

where $\Pi\left[\cdot\right]$ indicates the integral operator over $\pi\left(\boldsymbol{x}\right)$, and $\Pi\Pi'\left[\cdot\right]$ refers to the integral operator over $\pi\left(\boldsymbol{x}'\right)$ and $\pi\left(\boldsymbol{x}\right)$. For specific pairs of kernel κ and weight π, the integrals $\Pi\left[m\left(\boldsymbol{x}\right)\right]$, $\Pi\left[\kappa\left(\boldsymbol{x}, \mathcal{X}\right)\right]$ and $\Pi\Pi'\left[\kappa\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right]$ can be analytically derived in closed form. One can refer to Ref. Briol *et al.,* (2019) for a summary of such pairs. For the squared exponential kernel used in this work, the closed-form expressions can be found in our previous work (Wei *et al.,* (2020)).

Equation (5) reveals that the posterior mean equals the integral of the prior mean plus the term $\Pi\left[\kappa\left(\boldsymbol{x}, \mathcal{X}\right)\right] \mathcal{K}^{-1} \left(\mathcal{Y} - m\left(\mathcal{X}\right)\right)$ determined by the training data set $\mathcal{D}$. The posterior variance of the integration shown in equation (6) equals the integral of the prior variance minus a positive term $\Pi\left[\kappa\left(\boldsymbol{x}, \mathcal{X}\right)\right] \mathcal{K}^{-1} \Pi\left[\kappa\left(\mathcal{X}, \boldsymbol{x}'\right)\right]$ measuring the reduction of the posterior variance depended on the training data set $\mathcal{D}$. It is then clear that the performance of BC highly depends on the training data set $\mathcal{D}$. This training data set can be generated by simple random sampling, Latin-hypercube sampling, Sobol' low-discrepancy sequence, which allow to implement the BC in a parallel scheme as all the training points are designed at once. However, as has been shown by our previous work (Wei *et al.,* (2020)), with a sequential design strategy driven by suitable acquisition function, a great reduction of required integrand calls can be achieved. This will be introduced in the next subsection.

By the way, the variance of the n-dimensional integral in equation (1) can also be estimated by the trained GPR model as a by-product, and its posterior mean μ_{V_d} and variance $\sigma_{V_d}^2$ are formulated as (Song *et al.,* (2022)):

$$\mu_{V_d} = \Pi\left[\left(\mu_g\left(\boldsymbol{x}\right) - \mu_d\right)^2\right] + \Pi\left[\sigma_g^2\left(\boldsymbol{x}\right)\right], \tag{7}$$

and

$$\sigma_{V_d}^2 = 2\Pi\Pi'\left[\mathrm{cov}_g^2\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right] + 4\Pi\Pi'[\mu_g\left(\boldsymbol{x}\right)\mu_g\left(\boldsymbol{x}'\right)\mathrm{cov}_g\left(\boldsymbol{x}, \boldsymbol{x}'\right)], \tag{8}$$

As can be seen from the above equations, it is difficult to deduce their analytical expressions, therefore, the numerical algorithms are recommended to be applied here. In this work, the MCS method is used. Let $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{N_x}\}$ be the sample set of size N_x randomly generating from the density $\pi\left(\boldsymbol{x}\right)$, the MCS estimators of equation (7) and equation (8) can be formulated as:

$$\mu_{V_d}^{*} = \frac{1}{N_x}\sum_{i=1}^{N_x}\left[\mu_g\left(\boldsymbol{x}_i\right)^2\right] - \left[\frac{1}{N_x}\sum_{i=1}^{N_x}\mu_g\left(\boldsymbol{x}_i\right)\right]^2 + \frac{1}{N_x}\sum_{i=1}^{N_x}\left[\sigma_g^2\left(\boldsymbol{x}_i\right)\right], \tag{9}$$

and

$$\sigma_{V_d}^{2*} = \frac{2}{N_x}\sum_{i=1}^{N_x}\left[\mathrm{cov}_g^2\left(\boldsymbol{x}_i, \boldsymbol{x}_i\right)\right] + \frac{4}{N_x}\sum_{i=1}^{N_x}\left[\mu_g\left(\boldsymbol{x}_i\right)\mu_g\left(\boldsymbol{x}_i\right)\mathrm{cov}_g\left(\boldsymbol{x}_i, \boldsymbol{x}_i\right)\right], \tag{10}$$

respectively.

2.2. *Adaptive experiment design and rationale for parallelization*

The core of an adaptive design strategy for BC is the acquisition function. In our previous work, an acquisition function, called PVC function, has been developed, and shown to be much more effective than the random sampling design. The definition of the PVC function is formulated as (Wei *et al.,* (2020)):

$$\begin{aligned}\mathcal{L}^{\mathrm{PVC}} &= \pi\left(\boldsymbol{x}\right)\Pi'\left[\mathrm{cov}_g\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right] \\ &= \left(\Pi'\left[k\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right] - k\left(\boldsymbol{x}, \mathcal{X}\right)K^{-1}\Pi'\left[k\left(\mathcal{X}, \boldsymbol{x}'\right)\right]\right)\pi\left(\boldsymbol{x}\right).\end{aligned} \tag{11}$$

It was further shown that (Wei *et al.,* (2020)):

$$\sigma_d^2 = \int \mathcal{L}^{\mathrm{PVC}}\left(\boldsymbol{x}\right)\mathrm{d}\boldsymbol{x}. \tag{12}$$

Equation (12) reveals that the integration of the PVC function equals the posterior variance of the integration, indicating that the value of the PVC function at the point $\boldsymbol{x}$ measures the contribution of the GPR prediction error at $\boldsymbol{x}$ to the posterior variance of $\hat{d}$. It is also noted from equation (11) that the value of the PVC function at any non-training point $\boldsymbol{x}$ integrates the spatial correlation information of $\boldsymbol{x}$ with all the other sites. Thus, as a summary, the PVC function measures the contribution of the GPR prediction error at $\boldsymbol{x}$ to the posterior variance of the integral estimate, with the consideration of its correlation information with all the spatial sites in the integral support, and this is also why the acquisition function is named as PVC function. As stated in Refs. Rasmussen and Ghahramani (2003) and Wei *et al.,* (2020), the integration of the spatial correlation information into the cubature rule makes the Bayesian cubature superior to the traditional deterministic cubature rule in terms of accuracy. Indeed, the integration of the spatial correlation into the acquisition function also makes the adaptive experiment design more informative. The point with the largest GPR prediction error is not necessarily the one making the most contribution to the posterior variance of the integral estimate, but the one with the highest PVC value is. Inspired by this, an adaptive experiment design strategy is developed in Wei *et al.,* (2020).

The key of the adaptive experiment design is then to estimate the global maximum of the PVC function by using an optimization algorithms such as particle swarm optimization (PSO), genetic algorithm (GA) and Bayesian Interactive Search Algorithm (BISA) (Mortazavi (2021)). However, this design strategy will only produce one design point for each iteration, making it impossible to implement a parallel computation scheme. In practical applications, the computation of the integrand usually involve the call of one or more physic models, which are usually computationally expensive, and a parallel implementation of the adaptive design is anticipated. Fortunately, it is found that the PVC function is usually multimodal, which means there are more than one local maximum points, all of which makes important contributions to the posterior variance of the integral estimate. Inspired by this finding, it is then nature to implement the adaptive experiment design procedure in a parallel scheme by estimating the model response values of multiple local maximum points at once using multiple workers.

Before the development of the parallel computation, it is necessary to investigate the multimnodal property of the PVC function from both theoretical and benchmark studies. Let's first take a theoretical investigation for one-dimensional case. Substituting the training points into the PVC function yields:

$$\begin{aligned}\mathcal{L}^{\mathrm{PVC}}\left(\mathcal{X}\right) &= \left[\Pi'\left[k\left(\mathcal{X},\boldsymbol{x}'\right)\right] - k\left(\mathcal{X},\mathcal{X}\right)K^{-1}\Pi'\left[k\left(\mathcal{X},\boldsymbol{x}'\right)\right]\right]\pi\left(\mathcal{X}\right)\\ &= \left[\Pi'\left[k\left(\mathcal{X},\boldsymbol{x}'\right)\right] - \Pi'\left[k\left(\mathcal{X},\boldsymbol{x}'\right)\right]\right]\pi\left(\mathcal{X}\right)\\ &= \mathbf{0},\end{aligned} \tag{13}$$

indicating that, at each training point, the PVC function equals zero. As the PVC function is continuous, based on the Rolle's theorem in calculus, there must exist a point where the first derivative is zero, i.e., a extreme point. Thus, with more training points being added, more local extreme points may appear. The above fact demonstrates the multimodal property of the PVC function. Two examples of PVC functions for one- and two-dimensional integrals are schematically shown in Figure 1 for illustrating the multi-modal behaviors. For a one-dimensional integral, by constraining the variable $\boldsymbol{x}$ within adjacent training points, the estimation of all the extreme points can be transformed into a set of unimodal optimization problems. However, for multi-dimensional integral, the computation of all the local extreme points is a general multi-modal optimization problems.

One may think that, in terms of PVC values, the points near to the global maximum may contribute more to the posterior variance of the integral estimate than the other local maximum points, and it is more valuable to use these adjacent points. However, those points share also much similarity with the global maximum, and once the global maxima being added to the training data set, those adjacent points will then have little or even negative contribution to the integration accuracy. Thus, the key issue left for parallel implementation of the adaptive design scheme is to compute all the local maximal points, or at least several local extreme points with the largest PVC values. It should be noted that, the target function, e.g., the PVC function, has closed-form expression, and is computationally much cheaper

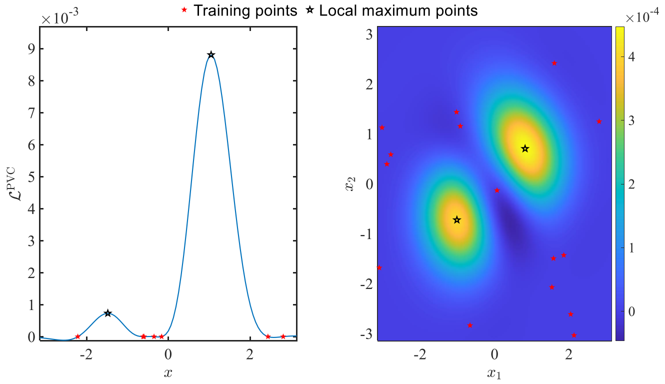

Figure 1. Examples of one-dimensional (left) and two-dimensional PVC functions.

than the integrand. In the next section, the multimodal optimization algorithms will be investigated.

3. Multimodal optimization algorithms for parallel computation

The multimodal optimization is a common problem in many research and engineering areas, and many approaches, especially those driven by metaheuristic algorithms such as GA and PSO, have been established (see e.g., Refs. Das *et al.,* (2011) and Ser *et al.,* (2019) for literature reviews). These methods aim at searching multiple global and local optima of a target function, and thus can served as engine for parallel implementation of the adaptive BC. Niching methods, such as speciation and crowding, provides flexibility to the metaheuristic algorithms for maintening population diversity. In this section, three multimodal optimization algorithms established based on the metaheuristic algorithms and niching methods, i.e., the Species Conserving Genetic Algorithm (SCGA) (Li *et al.,* (2002)), the MultiGrouped Particle Swarm Optimization (MPSO) (Seo *et al.,* (2006)), and the Whale Swarm Algorithm with Iterative Counter (WSA-IC) (Zeng *et al.,* (2017)), will first be examined for assisting the parallel implementation of adaptive BC, and, considering that the closed-form expression of the gradients of the PVC function are available, a gradient-based method, called Swarm Quasi-Newton Method (SQNM) will then be developed. The target is to find multiple maxima of the PVC function, but for ease of illustration, it is transformed to find the multiple local minima of $f(\boldsymbol{x}) = -\mathcal{L}^{\mathrm{PVC}}(\boldsymbol{x})$. The four algorithms, together with the k-means clustering method for comparison, are described as follows.

3.1. *Species Conserving Genetic Algorithm*

The classical GA algorithms was originally developed for finding the global optimum of a multimodal function, and it usually fails to identify multiple local optima. Alternative niching methods, such as preselection (Beasley *et al.,* (1993)), crowding (Mengshoel and Goldberg (2008)) and species conserving (Li *et al.,* (2002)), have been successfully injected into the GA with the aim of evolving parallel subpopulation, and this way to find multiple local optima. Here, the Species Conserving equipped GA will be utilized and key steps are briefly reviewed below.

Step 1.1: Initialization.

Let $k = 0$. Specify the maximum number of iteration. Initialize the k-th generation $G\left(k\right)$ by, e.g., simple random sampling, and evaluate the f-function values for all points in $G\left(k\right)$. Set species distance $\sigma_s = d/2$, where $d = 5\%L$, and L is the maximum distance of the search area.

Step 1.2: Judgment of stopping criteria.

Check whether the iteration is larger than the maximum iteration, if holds, go to **Step 1.6**; otherwise, go to **Step 1.3**.

Step 1.3: Determining the species seeds.

Let $X_s = \oslash$. Sort $G\left(k\right)$ in ascending order of the f-function value and consider each individual in turn. For an individual $\boldsymbol{x}^*$, one need to check whether there is one seed x in X_s satisfying $d\left(\boldsymbol{x}^*, x\right) \leqslant \sigma/2$, if not, add $\boldsymbol{x}^*$ to X_s. Then go to next individual until all individuals are considered. One can refer to Ref. Li *et al.,* (2002) for details.

Step 1.4: Implementation of Standard GA.

Implement the main procedures of the classical GA, i.e., selection $G\left(k+1\right)$, crossover $G\left(k+1\right)$ and mutate $G\left(k+1\right)$ (see Ref. Li *et al.,* (2002) for more details).

Step 1.5: Conserving species.

Set all individuals in $G\left(k+1\right)$ as unmarked. Consider each species seed $\boldsymbol{x}$ in X_s, find the worst unmarked individual $\boldsymbol{y}$ in $G\left(k+1\right)$ belonging to the species $S'\left(\boldsymbol{x}, \sigma_s\right)$ determined by $\boldsymbol{x}$. If $\boldsymbol{x}$ is better than $\boldsymbol{y}$, replace $\boldsymbol{y}$ with $\boldsymbol{x}$. If $\boldsymbol{y}$ does not exist, replace the worst unmarked individual in $G\left(k+1\right)$ with $\boldsymbol{x}$. Mark the replaced $\boldsymbol{y}$. Repeat the above procedure until all species seeds in X_s are considered. Let $k = k+1$, and go to **Step 1.3**.

Step 1.6: Identifying the global optima.

Determine the species seeds X_s of the final population $G\left(k\right)$, the seed $\boldsymbol{x}$ in X_s satisfying the following inequality will be identified as a global optima,

$$\mathrm{abs}\left(f\left(\boldsymbol{x}\right) - f_{\min}\right) \leqslant \mathrm{abs}\left(f_{\min}\right) \times 99\%, \tag{14}$$

where $f_{\min}$ is the minimum value of the f function.

■

The above procedures provide a rough description of the SCGA method, and one can refer to Ref. Li *et al.,* (2002) for more details, especially on the algorithm parameters. With this algorithm, multiple local peaks of the PVC function can be identified, but the number of peaks cannot be pre-specified. In the next subsection, the MPSO algorithm, which allows pre-specifying the number of local maxima to be identified, will be introduced.

3.2. *Multigrouped Particle Swarm Optimization*

Particle Swarm Optimization (PSO), originally developed by Kennedy and Eberhart in 1995 (Kennedy and Eberhart (1995)), is also a gradient-free and heuristic algorithm for searching the global optimum of a continuous function. The original version is not applicable for solving multiple peaks of the PVC function if it is multimodal. The MPSO algorithm, developed by Seo et al. in 2006 (Seo *et al.,* (2006)), overcomes this limitation, and it even allows to pre-specify the number of peaks to be found. With the assumption that the readers are familiar with the classical PSO algorithm, the main procedures of the MPSO algorithm are briefly summarized as follows.

Step 2.1: Initialization of algorithm parameters

Initialize the algorithm parameters including the number of groups (number of nadirs to be specified), size of population for each group, and the radius of each *gbest*, where *gbest* refers to the best position of a group.

Step 2.1: Initialization of swarm location and velocities

Initialize the locations and velocities of all particles by using, e.g., simple random sampling. The initial *pbest* of a particle is set to be its current location, and the *gbest* of a group is selected from the *pbests* with in the corresponding group (see **Step 6** in Ref. Seo *et al.,* (2006) for details).

Step 2.3: Update particle velocity

Update the velocities of all particles by:

$$\boldsymbol{v}_{ij}^{k+1} = w\boldsymbol{v}_{ij}^{k} + C_1\varphi_1\left(\mathrm{pbest}_{ij}^{k} - \boldsymbol{x}_{ij}^{k}\right) + C_2\varphi_2\left(\mathrm{gbest}_{i}^{k} - \boldsymbol{x}_{ij}^{k}\right) + C_3\varphi_3\left(\boldsymbol{x}_{ij}^{k} - \mathrm{gbest}_{m}^{k}\right), \tag{15}$$

where $\boldsymbol{v}_{ij}^{k}$, $\boldsymbol{x}_{ij}^{k}$ and pbest_{ij}^{k} refer to the velocity, the position and the *pbest* of the jth particle in the ith group at the k-th iteration respectively; gbest_{i}^{k} indicates the *gbest* of the i-th group; w is the inertia weight for balancing the global exploration and local exploitation; C_1 and C_2 are the acceleration parameters used for defining the relative pull of each particle toward *pbest* and *gbest* respectively; C_3 refers to the repulsive coefficient that determines the pull of a particle for pushing it out from territory of the group intruded by this particle; gbest_{m}^{k} implies the *gbest* of the m-th group intruded by the jth particle. For the setting details of the parameters in this step, one refers to Ref. Seo *et al.,* (2006).

Step 2.4: Update particle position

Update the position of each particle following:

$$\boldsymbol{x}_{ij}^{k+1} = \boldsymbol{x}_{ij}^{k} + \boldsymbol{v}_{ij}^{k} \tag{16}$$

Step 2.5: Update *pbests*

Compute the f-function value for each particle if its current position does not intrude the territory of the other *gbests*, and update the *pbest* of this particle as $\boldsymbol{x}_{ij}^{k+1}$ if $f\left(\boldsymbol{x}_{ij}^{k+1}\right) < f\left(\mathrm{pbest}_{ij}^{k}\right)$; otherwise, keep its *pbest* unchanged.

Step 2.6: Update *gbests*

A complex updating strategy was developed in Ref. Seo *et al.,* (2006) for avoiding overlap of some *gbests*, and one can refer to it for details.

Step 2.7: Judgment of convergence

If all particles are gathered around the the *gbest* of each group, or a maximum iteration is reached, finish the algorithm; otherwise repeat **Steps 2.3-2.6**.

■

In the above procedure, the initial radius of territory of a *gbest* is set to be 5% of the search space range. One can refer to Ref. Seo *et al.,* (2006) for details.

3.3. *Whale Swarm Algorithm with Iterative Counter*

The Whale Swarm Algorithm (WSA), inspired by the whale's communication mechanism, is also a gradient-free method for searching the global optima of a target function, and one can refer to Ref. Zeng *et al.,* (2017) for details. This algorithm was later equipped with iterative counter by the same authors in Ref. Zeng *et al.,* (2020) for searching multiple local optima. The main procedures of this algorithm are briefly summarized as follows.

Step 3.1: Initialization

Initialize the whale swarm $\mathcal{S}$ (by e.g., simple random sampling) and the maximum iteration N. Evaluate the f-function values for all whales. Let $k = 0$, and creat a null global optima set $\mathcal{S}_{opt}$.

Step 3.2: Judgment of termination criterion

Check if $k > N$ holds, if no, turn to **Step 3.3**, and let $i = 1$; otherwise turn to **Step 3.7**.

Step 3.3: Find the "better and nearest" whale Y for $\mathcal{S}_i$

Let $X_{\mathcal{S}} = \oslash$. Check each whale $\mathcal{S}_u$ in $\mathcal{S}$ if it satisfies $f\left(\mathcal{S}_u\right) < f\left(\mathcal{S}_i\right)$, if yes, add it to $X_{\mathcal{S}}$. Find the whale in $X_{\mathcal{S}}$ that is nearest to the i-th whale $\mathcal{S}_i$. If Y exists, go to **Step 3.4**; otherwise, go to **Step 3.5**.

Step 3.4: Movement of $\mathcal{S}_i$

Compute the candidate position $\mathcal{S}_i$ of the i -th whale for the $(t+1)$-th iteration by:

$$x_j^{k+1} = x_j^k + \mathrm{rand}\left(0, \rho_0\right) \times \left(y_j^k - x_j^k\right), \tag{17}$$

where ρ_0 is the intensity of ultrasound and usually set to be 2; y_j^k refers to the j-th component of Y. Compute the f-function value at the candidate position, and if it is better than the f-function value of $\mathcal{S}_i$, update the position of the i-th whale with this candidate position and set $\mathcal{S}_{i,c} = 0$; otherwise, keep $\mathcal{S}_i$ unchanged.

Step 3.5: Check the iterative counter of $\mathcal{S}_i$

Denote T_s as the stability threshold which can be reasonably set as $100n$. Check if $\mathcal{S}_{i,c} = T_s$ holds or not, if yes, further judge whether $\mathcal{S}_i$ is a current global optimum by the methods in **Step 3.6**, then reinitialize $\mathcal{S}_i$ and evaluate $\mathcal{S}_i$. If not, $\mathcal{S}_{i,c} = \mathcal{S}_{i,c}+1$. If $i = |\mathcal{S}|$, let $t = t + 1$, go to **Step 3.2**; otherwise, repeat **Steps 3.3-3.5**.

Step 3.6: Judge whether each individual in $\mathcal{S}$ is a current global optimum.

Let f_{gbest} be the function value of the best whale among $\mathcal{S}_{opt}$, and denote T_f as the function threshold having a value between 0 and $f\left(\mathcal{S}_{\text{best}}\right)$. The way to judge whether a whale $\mathcal{S}_i$ is a current global optimum is described as follows.

If $f\left(\mathcal{S}_i\right) < f_{gbest}$ and $f\left(\mathcal{S}_i\right) - f_{gbest} > T_f$, Clear $\mathcal{S}_{opt}$, let $f_{gbest} = f\left(\mathcal{S}_i\right)$ and add $\mathcal{S}_i$ to $\mathcal{S}_{opt}$.

If $f\left(\mathcal{S}_i\right) < f_{gbest}$ and $f\left(\mathcal{S}_i\right) - f_{gbest} \leqslant T_f$, let $f_{gbest} = f\left(\mathcal{S}_i\right)$ and add $\mathcal{S}_i$ to $\mathcal{S}_{opt}$.

If $f\left(\mathcal{S}_i\right) \geqslant f_{gbest}$ and $f\left(\mathcal{S}_i\right) - f_{gbest} \leqslant T_f$, add $\mathcal{S}_i$ to $\mathcal{S}_{opt}$.

Repeat the above process until all whales in $\mathcal{S}$ are considered, then output $\mathcal{S}_{opt}$.

■

One need to note that the f-function value of some individuals in $\mathcal{S}_{opt}$ maybe are too small to apply, therefore, the final solution set will be consisited of the individuals in $\mathcal{S}_{opt}$ whose f-function values are better than the given threshold e.g., $1\% f_{\min}$. Please refer to (Zeng *et al.,* (2020)) for details.

3.4. *Swarm Quasi-Newton Method*

All the three above-reviewed multimodal optimization algorithms are gradient-free, and thus are applicable to non-smooth target functions. However, the gradient of the PVC function expressed by equation (11) is available and can be formulated in closed form. This provides possibility of using more efficient gradient-based methods, such as the Quasi-Newton method for solving the optimization problem. However, the classical Quasi-Newton can only find the global minimum of the f-function, but not multiple local minima. Here, we improve it for searching multiple nadirs, and the detailed procedures are given below.

Step 4.1: Initialization

Let $k = 0$. Generate a sample set $\mathcal{S} = \left\{\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \cdots, \boldsymbol{x}_N^k\right\}$ of size N by e.g., simple random sampling. Evaluate the values of the gradient vector $\nabla f\left(\boldsymbol{x_i^k}\right)$ for each sample. One can refer to the Appendix of our previous work (Wei *et al.,* (2020)) for the closed-form expression of the gradients of the PVC function. Create an identity matrix $\mathcal{H}_i^k$ of order n for each sample point $\boldsymbol{x}_i^k$. Create a null solution set $\mathcal{S}_{opt}$.

Step 4.2: Iteration

Update the location of each individual sample $\boldsymbol{x}_i^k$ by:

$$\begin{cases} \boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k - \lambda_i^k \mathcal{H}_i^k \nabla f\left(\boldsymbol{x}_i^k\right) \\ \mathcal{H}_i^{k+1} = \mathcal{H}_i^k + \frac{\left(\Delta \boldsymbol{x}_i^k - \mathcal{H}_i^k \Delta f_i^k\right)\left(\Delta \boldsymbol{x}_i^k - \mathcal{H}_i^k \Delta f_i^k\right)^{\mathrm{T}}}{\left(\Delta \boldsymbol{x}_i^k - \mathcal{H}_i^k \Delta f_i^k\right)^{\mathrm{T}} \Delta f_i^k} \\ , \end{cases} \tag{18}$$

where $\Delta \boldsymbol{x}_i^k = \boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i^k$, $\Delta f_i^k = \nabla f\left(\boldsymbol{x}_i^{k+1}\right) - \nabla f\left(\boldsymbol{x}_i^k\right)$, and λ_i^k is the step parameter with its values computed by solving $\mathrm{d} f\left(\boldsymbol{x}_i^k - \lambda_i^k \mathcal{H}_i^k \nabla f\left(\boldsymbol{x}_i^k\right)\right) / \mathrm{d}\lambda_i^k = 0$. One notes that,for f function, the values of the gradient $\nabla f\left(\boldsymbol{x}_i^k\right)$ may sometimes be too small, causing large numerical bias for iteration. Thus, it is suggested to replace the gradient with its normalized version $\nabla f\left(\boldsymbol{x}^k\right) / \left|\nabla f\left(\boldsymbol{x}^k\right)\right|$.

Step 4.3: Judgment of stopping criteria for individuals

For each individual, judge if $\left|\Delta \boldsymbol{x}_i^k\right| < \epsilon$ hold. If it holds, set $\boldsymbol{x}_i^{k+1}$ as a candidate solution, and remove it from $\mathcal{S}$. Compute the posterior correlation coefficient of this candidate individual with all the points in $\mathcal{S}_{opt}$ (see equation (4)). If this value is smaller than, e.g., 0.9, add this point to $\mathcal{S}_{opt}$.

Step 4.4: Judgment of stopping criteria for swarm

Check if $\mathcal{S}$ is empty. If yes, stop the implementation and produce the solution set $\mathcal{S}_{opt}$; otherwise, go to step **Step 4.2**.

■

Similar with the first three algorithms, the final solutions for SQNM method are set to be the same as equation (14).

3.5. *K-means clustering*

The existing parallel computing methods are mostly based on the the clustering algorithms, including k-means clustering (Dang *et al.,* (2022)), k-medoids clustering (Chen *et al.,* (2022)), spectral clustering (Li *et al.,* (2021)) and density clustering (Teixeira *et al.,* (2020)), etc. For demonstrating the efficiency of the proposed parallel strategies motivating by multimodal optimization method, the clustering algorithm is used to compare. Considering that k-means clustering is easy to be understood and implemented, therefore, k-means clustering is applied for comparing in this paper. The main steps of it are given as follows.

Step 5.1: Initialization

Initialize the number of clusters p, and the sample set $\mathcal{S} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$ of size N by random sampling. Randomly choose p points $\{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_p\}$ from $\mathcal{S}$ as the initial centroids.

Step 5.2: Clustering

For each sample of $\mathcal{S}$, find a centroid minimizing the square of the Euclidean distance between the sample and the centroid. Denote the i-th cluster as $\mathcal{Z}^{(i)} = \left\{\boldsymbol{z}_1^{(i)}, \cdots, \boldsymbol{z}_{N_i}^{(i)}\right\}$, where N_i represents the size of $\mathcal{Z}^{(i)}$.

Step 5.3: Update

Update the centroids by the PVC-weighted mean of a cluster, i.e.,

$$\boldsymbol{c}_i = \frac{\sum_{j=1}^{N_i} \left[\mathrm{PVC}\left(\boldsymbol{z}_j^{(i)}\right) \times \boldsymbol{z}_j^{(i)}\right]}{\sum_{j=1}^{N_i} \left[\mathrm{PVC}\left(\boldsymbol{z}_j^{(i)}\right)\right]} \tag{19}$$

Step 5.4: Iteration

Repeat **Step 5.2** and **Step 5.3** until the relative differences of the current centroids and previous centroids are accepTable

■

Output the final centroids, and the available solutions for k-means clustering are also set to be the same as equation (14).

4. Parallelized adaptive Bayesian cubature

With any one of the above four multimodal optimization algorithms or the k-means clustering method, a solution set $\mathcal{S}_{opt}$ with multiple design points for parallel calls of the integrand can be achieved based on the posterior features of the integrand and the integral. These algorithms can be then integarated for parallel implementation of the adaptive BC, and the flowchart of the algorithm is schematically shown in Figure 2, and procedures are described in details as follows.

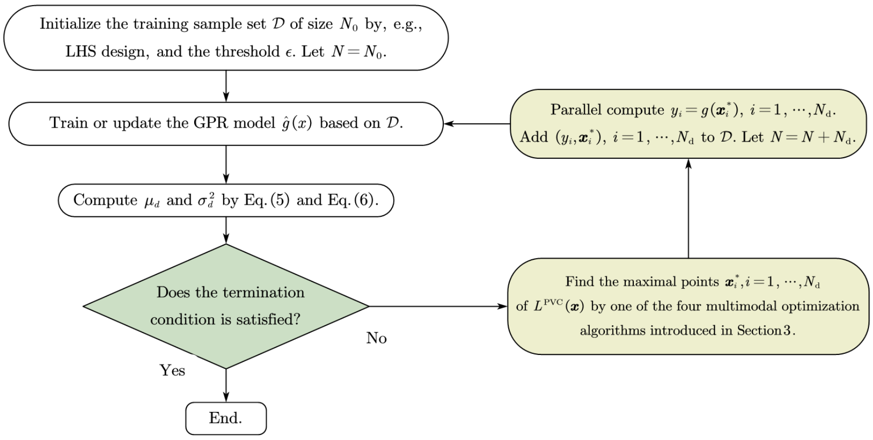
Initialize the training sample set $\mathcal{D}$ of size N_0 by, e.g., LHS design, and the threshold ϵ. Let $N=N_0$.
Train or update the GPR model $\hat{g}(x)$ based on $\mathcal{D}$.
Compute μ_d and σ_d^2 by Eq. (5) and Eq. (6).
Does the termination condition is satisfied?
Yes
End.
No
Find the maximal points $\boldsymbol{x}_i^*, i=1, \cdots, N_{\mathrm{d}}$ of $L^{\mathrm{PVC}}(\boldsymbol{x})$ by one of the four multimodal optimization algorithms introduced in Section 3.
Parallel compute $y_i=g(\boldsymbol{x}_i^*)$, $i=1,\cdots,N_{\mathrm{d}}$. Add $(y_i,\boldsymbol{x}_i^*)$, $i=1,\cdots,N_{\mathrm{d}}$ to $\mathcal{D}$. Let $N=N+N_{\mathrm{d}}$.

Figure 2. Flowchart of the parallelized adaptive experiment design algorithm.

Step 6.1: Initialization

Initialize the initial training sample size N_0 (depends on the dimension of $\boldsymbol{x}$), the stopping threshold ϵ. Generate N_0 samples of $\boldsymbol{x}$ by, e.g. simple random sampling or Latin hypercube sampling (LHS) following uniform probability distribution within the support of $\boldsymbol{x}$. Evaluate the integrand values for these N_0 design points, and initialize the design point set $\mathcal{D}$ with these N_0 points. One notes that the integrand values can be computed by parallel computation, which means to compute the integrand value for each design point by one worker. Initialize the number of integrand calls as $N = N_0$, and the number of steps as $N_s = 1$.

Step 6.2: Bayesian inference of the integrand

Train or update the GPR model $\hat{g}(\boldsymbol{x})$ for the integrand based on $\mathcal{D}$, and record the values of the hyper-parameters. In this work, the squared exponential kernel is used for training.

Step 6.3: Bayesian inference of the integral

Evaluate the posterior mean μ_d and the posterior variance σ_d^2 for the integral d by Eqs. (5) and (6).

Step 6.4: Judgment of stopping criteria

The posterior Coefficient of Variation (COV) of integral is set to be the stopping criteria, and is defined as COV $= \sigma_d/\mu_d$. Check whether COV is less than the specifed threshold ϵ, i.e., COV $< \epsilon$ is satisfied or not. If yes, finish the algorithm and produce the posterior features μ_d and σ_d^2; otherwise go to Step 5.5. One notes it is usually suggested to use a delay judgment strategy, which means to finish the algorithm if the stopping criteria is satisfied in succession for, e.g., twice.

Step 6.5: Parallel experiment design

Implement one of the four multimodal optimization algorithms or the k-means clustering method for achieve a design point set $\mathcal{S}_{opt}$, and evaluate the integrand values for each design point contained in $\mathcal{S}_{opt}$. Add these training points to $\mathcal{D}$. Let $N = N + |\mathcal{S}_{opt}|$, and $N_s = N_s + 1$, go to **Step 6.2**.

■

Based on the above description, the algorithm parameters of the proposed parallelized adaptive BC need to be pre-specified include the number N0 of the initial training sample set and the stopping criterion ϵ. Generally, N_0 can be set to be a small value, e.g., 3, 5 or $n+1$, if the prior mean is set to be zero or constant, for the highly non-linear g-function, N_0 is recommended to be a big value. The stopping threshold ϵ measures the normalized variation of posterior mean of expectation, and it is reasonable to choose it in the interval $[0.01, 0.05]$, and for the highly non-linear g-function, based on our experience, it is advised to choose a small value.

One notes that, in **Step 6.5**, as the evaluation of the PVC function does not require calling the integrand, the implementation of any multimodal optimization algorithms does not involve calling of the expensive integrand. It is also noted that, among the five parallelized algorithms, only the MPSO and k-means clustering algorithm allow pre-specifying the number of points to be designed. Once the above procedures stop, not only the posterior feature μ_d and σ_d^2 of the integral in equation (1) can be output, but also the variance of integral given in equation (7) and equation (8).

5. Benchmark study

In this section, we introduce several examples for demonstrating the effectiveness of the proposed parallel design schemes driven by multimodal optimization algorithms for adaptive BC, compared with the parallel strategy with k-means clustering algorithms. For ease of illustration, we add the two and three training points with the maximum PVC values to the training data set at each iteration for the MPSO and k-means clustering algorithms, indicating that two and three workers are required for the parallel computation respectively; for the other three multi-modal algorithms, all the produced design points with PVC values higher than one percent of the maximum values of the PVC function. The estimators of integral and variance, as well as the posterior COVs are used to measure the computational accuracy of parallel and non-parallel algorithms, and the number of iteration is used to measure their the computational efficiency.

5.1. *A one-dimensional numerical example*

Considering a one-dimensional integral with the integrand formulated as:

$$g\left(x\right) = x\cos\left(4x\right) + x^2\sin\left(4x\right) + 1, \tag{20}$$

where x follows standard normal distribution. Our target is to estimate and variance the expectation of the integrand, and the analytical results are 1 and 2 respectively.

We use this one-dimensional example to illustrate and compare the different methods for design. The adaptive BC is implemented in both parallel and non-parallel schemes, and for parallel implementation, all the five algorithms are applied. For all the implementation, we set $N_0 = 3$ and $\epsilon = 2\%$. Besides, the sparse grid cubature rule (for this one-dimensional problem, it degrades into the Gauss-Hermite rule) is also implemented for comparison.

For illustration of each design strategies, the design details of the first step of each implementation are schematically compared in Figure 3. It is shown that the PVC function inferred from the three initial design points has three peaks. It is also shown that the MPSO algorithm successfully finds the two most important peaks as the number of design points to be found for this implementation is set to be two. The other three multimodal optimization algorithms, i.e., SCGA , WSA-IC and SQNM, have all successfully found all the three peaks. It is also noted by the first panel that the classical PSO algorithm only finds the most important peak. This above facts demonstrate that all the four multimodal algorithms are effective for identifying multiple maxima, and thus are effective for parallel design of the integration points. While for the k-means clustering algorithm, though the PVC value of identified design point are large, these points are irregular. For illustrating the computational cost, the computer time for performing each parallel algorithm is also marked in the corresponding panel of Figure 3. It can be seen from that, although each of the five parallelized algorithms takes more time than the PSO for searching the global maximum, their costs are all less than 2.5 seconds, and thus are negligible compared to the cost for evaluating the expensive integrand in real-world application.

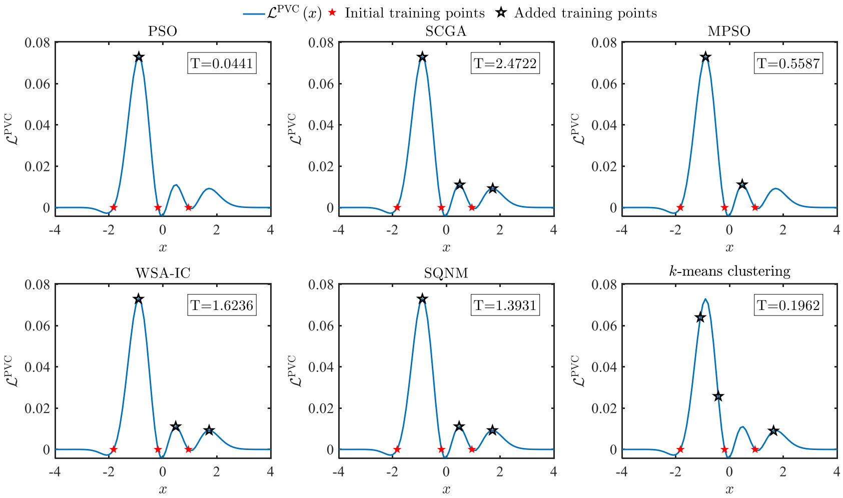

Figure 3. Comparison of the experiment design results for the one-dimensional example by different parallel approaches (PSO, SCGA, MPSO, WSA-IC, SQNM and k-means clustering).

Then we implement the parallelized adaptive BC to solve this one dimensional integral. The results of all the five parallel implementations are reported in Table I and Figure 4, where the results of non-parallelized adaptive BC with PVC function and sparse grid cubature are also listed for comparison. The evolution of the posterior COVs against the iteration step is compared in Figure 5. As can be seen, the convergence rates of SCGA, WSA-IC and SQNM are higher than MPSO and k-means clustering, meaning that the parallel algorithms with the prespecified number of design point are less inefficient compared with the adaptive scheme. Among the five parallel methods, the number of iterations required by k-means clustering is the largest, it is probably because the contribution of the identified design points to the posterior variance of integral is less than the points generated by the multimodal optimization algorithms. It is also shown that, for this example, all the parallel computation methods for adaptive BC are more efficient than the non-parallel one demonstrating the efficiency of parallel scheme. One can see from Table I, all methods produce accurate estimator for the integral, but the accuracy of variance for PVC is low than other methods. Based on the above results, it can be indicated that the developed parallel strategies for adaptive BC is effective for this example. It is also seen, for this one-dimensional problem, the adaptive BC procedures show no superiority to the sparse grid in terms of efficiency, but they provide estimations of numerical errors.

Table I. Probabilistic integration results for one-dimensional model by adding all generated points.

Methods	Means	Variance	COVs (%)	iterations	Number of samples
SCGA	0.9912	2.0521	0.60	5	17
MPSO	0.9930	2.0909	0.59	7	17
WSA-IC	0.9961	2.0372	0.34	5	21
SQNM	0.9975	2.0902	0.76	4	18
k-means clustering	1.0069	2.0060	0.51	9	18
PVC	1.0060	1.9662	0.79	12	15
Sparse grid	1.0000		–	–	5
True value	1	2			

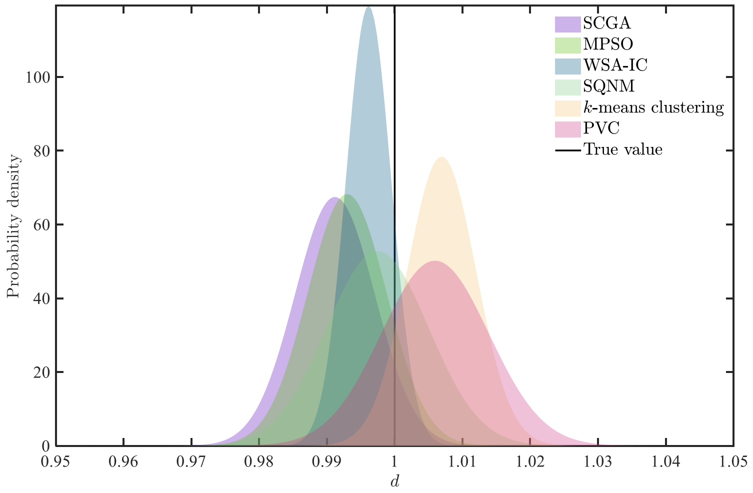

Figure 4. Comparison of the integration results for one-dimensional model by parallelized algorithm and non-parallel algorithm with all the identified PVC maxima being added to the training data set in each iteration.

5.2. *A ten-dimensional numerical example*

Considering a ten-dimensional integral with the integrand formulated as:

$$\begin{aligned} g(x) =& \sin(x_1) + 7\sin(2x_2) + x_3^4\sin(x_3) + \sin(x_4) + 6\sin(2x_5) + x_6^4\sin(x_6) \\ &+ \sin(x_7) + 5\sin(2x_8) + x_9^4\sin(x_9) + \sin(x_{10}) + 10, \end{aligned} \tag{21}$$

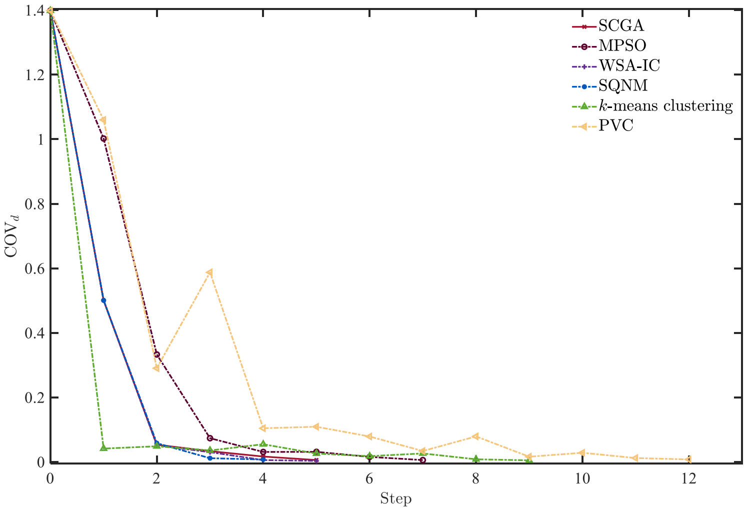

Figure 5. Evolution of posterior COVs of the probabilistic integration result for one-dimensional model by adding all generated points.

where $x_i, i = 1 \cdots, 10$ are independent standard normal random variables. The analytical result of the integral is 10, the reference value of variance is 143.8333 evaluated by MCS with 10000 samples, and we use this example to illustrate the effectiveness of parallel experiment design for multi-dimensional cases.

For this example, N_0 is set to be 70 for all the implementation, and we also set the total steps of iteration for each implementation as 70. For MPSO and k-means clustering, the numbers of design points to be added for training for each iteration are set to be 2 and 3, while for the other three multimodal optimization algorithms, all the identified PVC maxima are added to the training data set. The integration results generated by both parallelized and non-parallelized adaptive BC are then summarized in Table II, together with the results of sparse grid for comparison. From these results, we can come to the conclusion that, in terms of iteration steps and posterior COVs, the parallel implementation with any of the four multimodal optimization algorithms outperforms the one without parallelization, while the parallel design scheme with k-means clustering show lower efficiency than the original adaptive BC, it is probably because the identified design point of k-means clustering contribute less to the posterior variance of integral than the maximum point of PVC. The evolution of the posterior COVs of SCGA, MPSO, WSA-IC and SQNM is then reported in Figure 6. It is also clear that after certain steps of iterations, all the parallel strategies, except the k-means clustering, consistently produce smaller posterior COVs than the one without using parallelization, demonstrating the effectiveness of the proposed parallel algorithms equipped with multimodal optimization algorithms for this ten-dimensional problems. One can also seen that, using the sparse grid method with 221 integrand calls, the estimation is accurate enough. By the way, from the view of variance of integral, all method produce unacceptable results, it is probably because the linearity of integrand is too strong, or the uncertainties of inputs are to big, and one can add more training points to reduce the estimation error and improve the accuracy.

Table II. Integration results of the ten-dimensional numerical model.

Methods	Means	Variances	COVs (%)	Iterations	Samples
SCGA	10.2007	122.4331	1.79	70	278
MPSO	10.0079	115.1602	1.90	70	212
WSA-IC	9.9271	118.2989	1.29	70	298
SQNM	10.0558	104.0873	1.17	70	254
k-means clustering	9.8386	118.1808	2.72	70	220
PVC	9.8291	104.2588	2.41	70	141
Sparse Grid	10.0000	—	—	—	221
Reference value	10	143.8333	—	—	10000

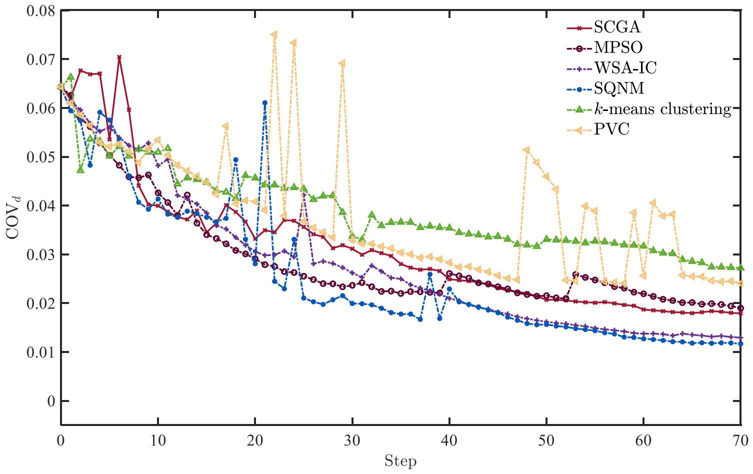

Figure 6. Evolution of the posterior COVs against the iteration step for ten-dimensional numerical model.

5.3. *Application to a wide flange steel column*

Considering a wide flange steel column depicted in Figure 7 selected from Ref. Papaioannou *et al.,* (2019), a compression force P, consisting of the permanent load P_p and the environmental load P_e, i.e., $P = P_\mathrm{p} + P_\mathrm{e}$, is applied to the column. Due to the construction imperfections, the column is assumed to have an initial deformation. The deformation has a parabolic shape with maximum amplitude δ_0 at the center of the column.

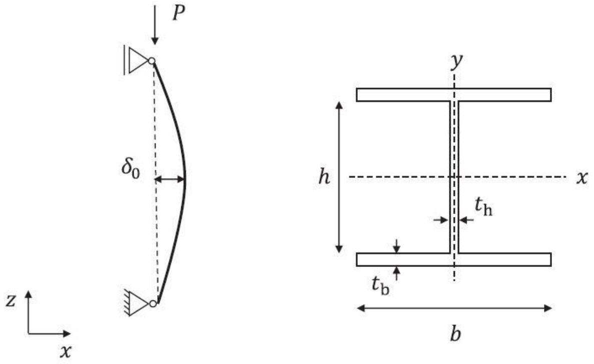

Figure 7. Wide flange steel column with initial deflection.

The column has a length of $L = 7.5$m. The critical limit state that governs the performance of the column is formulated as:

$$g(\boldsymbol{x}) = 1 - \left(\frac{P}{f_\mathrm{y} A_\mathrm{s}} + \frac{P\delta_0}{f_\mathrm{y} W_\mathrm{s}} \cdot \frac{P_\mathrm{b}}{P_\mathrm{b} - P} \right), \tag{22}$$

where $\boldsymbol{x}$ denotes possible outcomes of the uncertain parameters $\boldsymbol{x} = [P_\mathrm{p}, P_\mathrm{e}, \delta_0, f_\mathrm{y}, E]$. P_b is the Euler buckling load and is given by

$$P_\mathrm{b} = \frac{\pi^2 E I_\mathrm{s}}{L^2}, \tag{23}$$

A_s, W_s, and I_s denote the area, section modulus, and moment of inertia of the cross section around its weak axis, respectively, and are given by

$$\begin{cases} A_\mathrm{s} = 2bt_\mathrm{b} + ht_\mathrm{h} \\ W_\mathrm{s} = \frac{ht_\mathrm{h}^3}{6b} + \frac{t_\mathrm{b}b^2}{3} \\ I_\mathrm{s} = \frac{ht_\mathrm{h}^3}{12} + \frac{t_\mathrm{b}b^3}{6} \end{cases} \tag{24}$$

with f_y indicating the yield strength and E the Youngs modulus of the steel material. The target is to estimate the expectation of $g(\boldsymbol{x})$, which can be formulated as:

$$d = \mathbb{E}\left[g\left(P_\mathrm{p}, P_\mathrm{e}, \delta_0, f_\mathrm{y}, E\right)\right]. \tag{25}$$

The related parameters are shown in Table III, five of which do not follow Gaussian distribution, thus the Rosenblatt transformation is first applied to transform the integration space into standard normal space.

This problem is solved by the five parallel and the one non-parallelized adaptive BC algorithms by setting $N_0 = 5$ and $\epsilon = 1\%$. The results are reported in Table IV.

The evolution of the posterior COVs against the iteration steps is shown in Figure 8. The reference solution is generated by MC simulation with 50000 samples. It is shown that all the adaptive BC algorithms produce robust estimations of integral and its variance with high accuracy, but obviously, the four multimodal optimization based parallel algorithms need much less steps of iteration than k-means clustering and the original adaptive BC, demonstrating the efficiency of the parallel schemes driven by multimodal optimization algorithms.

Table III. Information of the related parameters of the wide flange steel column.

Input variables	Distribution type	Mean	STD	Supports	Values
Permanent load P_p (kN)	Uniform	—	—	[0.1 1000]	—
Environmental load P_e (kN)	Uniform	—	—	[0.1 1000]	—
Maximum amplitude δ_0 (mm)	Uniform	—	—	[0 0.3]	—
Yield strength f_y (MPa)	lognormal	400	32	—	—
Youngs modulus E (MPa)	lognormal	210000	84000	—	—
Flange width b (mm)	—	—	—	—	300
Flange thickness t_b (mm)	—	—	—	—	15
Web height h (mm)	—	—	—	—	300
Web thickness t_h (mm)	—	—	—	—	15

Table IV. Integration results for the wide flange steel column, and the reference values are estimated by MCS with 10^5 samples

Algorithms	Means	Variances	COVs (%)	Iterations	Samples
SCGA	0.6329	0.0555	0.55	8	46
MPSO	0.6308	0.0570	0.64	8	21
WSA-IC	0.6302	0.0575	0.51	9	27
SQNM	0.6331	0.0547	0.84	8	20
k-means clustering	0.6212	0.0533	0.72	11	38
PVC	0.6239	0.0738	0.87	15	20
Sparse Grid	0.6322	—	—	—	781
Ref.value	0.6326	0.0543	0.16	–	50000

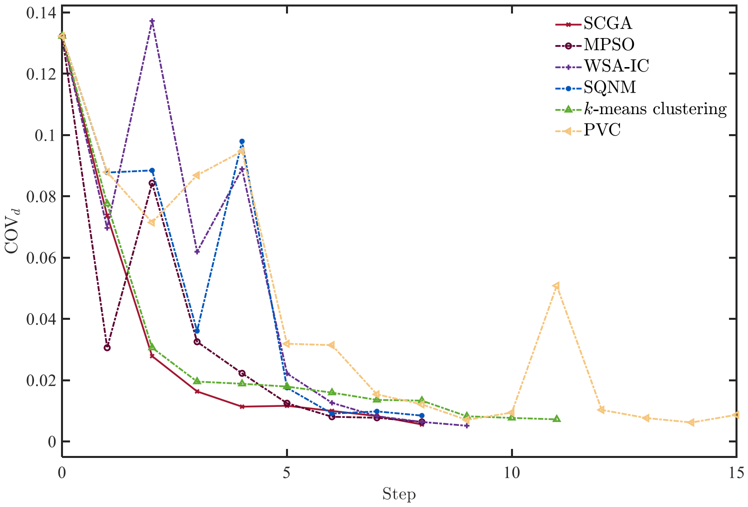

Figure 8. Evolution of the posterior COVs against the iteration step for the wide flange steel column.

5.4. *An elastoplastic analysis problem*

For further illustrating the effectiveness of the proposed parallel algorithms for real engineering problems, we then apply them to a nonlinear analysis of a one quarter annular plate in elastoplastic subjected to small strains, which is adapted from Ref. Du *et al.,* (2020). The Geometry of the structure is shown in the left panel of Figure 9, with the left end clamped and the right end being subjected to a uniform load P. Two dimension parameters, including the inner radius r and the outer radius R, three material parameters, i.e., the Young's modulus E, the Poission ratio v and the Yield stress Y_0, and the load parameter P are all random variables with distribution settings shown in Table V. The target is to estimate the expectation of the absolute displacement in the vertical direction the at node C.

Instead of using finite element analysis, a more advanced approach, called Nonlinear Isogeometric Analysis (NLIGA) is utilized for establishing the deterministic simulation model, and one can refer to Ref. Du *et al.,* (2020) for detailed description of this method. A deterministic simulation with all the six random parameters fixed at their mean values is implemented and the displacement nephogram in the vertical direction is shown by the left panel of Figure 9.

For non-parallel implementation of the adaptive BC, the PVC is utilized, and for parallel implementation, each of the four multimodal optimization algorithms and the k-means clustering are used. The initial size of training set is set to 6, and the stopping criterion is set to be 0.01. The results are listed in Table VI, together with the results of sparse grid integration, and the reference solutions generated by MCS, for comparison. We first compare the results of sparse grid with the reference solutions. It is seen that the COV of the MCS estimate is around 0.23%, indicating

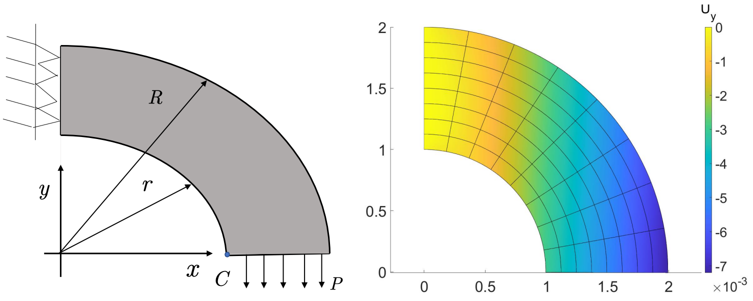

Figure 9. The structure of the quarter annular plate.

that its result is sufficiently accurate since the MCS estimator is unbiased. For the sparse grid, the six-dimensional integration points are specified using Gaussian-Hermite rule with the one-dimensional point size set to be 5, resulting in 1433 model calls respectively, as shown in VI. As can be seen, with 1433 model calls being consumed, the sparse grid method even provide poor estimation, indicating the inaccuracy of sparse grid for this example.

We then compare the results generated by adaptive BC with and without parallelization. It is seen that all the proposed multimodal optimization based parallel implementations, the k-means clustering based parallel implementation and the non-parallel implementation produce accurate and robust estimation to the expected vertical displacement its variance of point C. Specifically, the four multimodal optimization based parallel implementations are much more efficient than the k-means clustering based parallel implementation and the non-parallel implementations in terms of iteration steps. It is also shown that among the five parallel implementations, the one driven by SCGA are the most efficient one in terms of both iteration steps.

With the above analysis, it can be concluded that, for this example, the adaptive BC schemes are all more efficient than robust than the sparse grid cubature, the parallel implementations are all more efficient than the non-parallel implementation of the adaptive BC and the parallel implementations driven by multimodal optimization algorithms are all more efficient than the parallel implementation equipped with the k-means clustering.

5.5. *Final remarks*

The high efficiency of the proposed parallelized adaptive BC implemented with four multimodal optimization algorithms, together with its superiority to k-means clustering, is demonstrated by the above four benchmarks. The reason behind this results is that the added training points identified by multimodal optimization algorithms make a greater contribution to the posterior variance, than these produced by k-means clustering. Besides, with the well trained GPR model for estimating the integral, the corresponding variance of integral can also be deduced, and the

Table V. Setting of parameters for the elastoplastic analysis example.

Input variables	Distribution type	Mean	STD	Supports	Values
Inner radius r (m)	Uniform	—	—	[0.9 1.1]	—
Outer radius R (m)	Uniform	—	—	[1.9 2.1]	—
Load P (kN)	lognormal	45000	9000	—	—
Young's modulus E (GPa)	lognormal	200	40	—	—
Poission ratio v	Uniform	—	—	[0.15,0.35]	—
Yield stress Y_0 (kN)	lognormal	200000	40000	—	—

Table VI. Adaptive BC results for the elastoplastic analysis problem.

Methods	Means ($\times 10^{-3}$)	Variance ($\times 10^{-5}$)	COVs (%)	Iterations	Samples
SCGA	4.3082	1.1287	0.79	9	46
MPSO	4.3066	1.0757	0.65	15	36
WSA-IC	4.2668	1.0192	0.95	12	39
SQNM	4.2149	1.0139	0.53	14	35
k-means cluster	4.2361	0.8468	0.86	16	45
PVC	4.3505	1.1449	0.79	19	25
Sparse Grid	3.5946	–	—	—	1433
Ref.value	4.2547	0.9303	0.23		100000

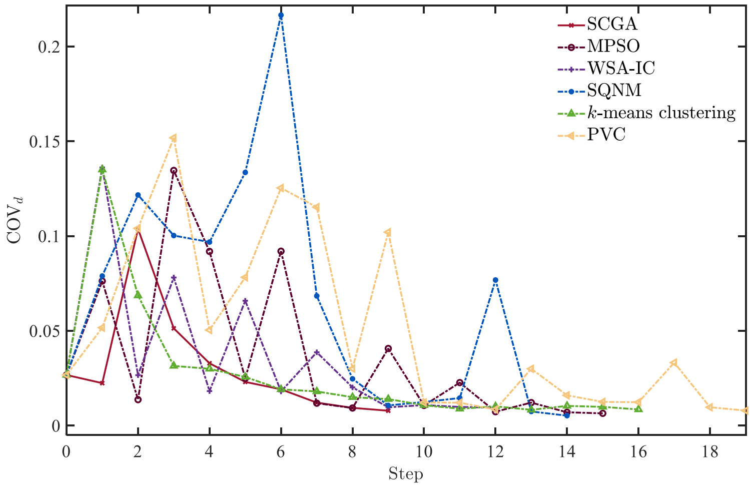

Figure 10. Comparison of C.O.V.s of the probabilistic integration result for the elastoplastic analysis problem.

proposed method still show superiority to the k-means clustering and the non-parallelized one.

The posterior COVs measure the normalized error of estimators, the smaller the COVs, the more precise the estimators. Therefore, we believe that the results with small COVs are more accurate than the one with big COVs. As can be seen from the above four benchmarks, the value of the posterior COVs for different parallelized adaptive BC are different, it is because, the four multimodal optimization algorithms and k-means clustering are developed by different mechanisms, their computational performance will show divergence. It also can be seen that, the COVs of the five parallel algorithms are different from the reference solution, where the COVs of the reference solutions are computed by MCS. According to our experiments, although the evaluated COVs of parallel algorithms are different from the reference solution, their accuracies are all acceptable if the value of their COVs are less than the termination thresholds.

In this paper, the relative low-dimensional problems are settled, and for high-dimensional problems, it is a pain spot of GPR model to perform statistical inference, as the kernel function used by GPR model is based on the Euclid distance, and it carries less information with the dimension increasing. Therefore, it is recommended to use the built-in dimension reduction techniques or replace the GPR model with a Bayesian neural network, and this will be investigated in future work. The proposed parallelized adaptive BC iteratively produces the design points based on a relative tiny initial training set, and it basically does not suffer from the big data problems. For the special cases with big data, the main difficulty of GPR model is the estimation of the inverse of covariance matrix, and, there are indeed some strategies to alleviate this limitation, for examples, the sparse covariance approximation method and the grid-based covariance approximation method (Rasmussen and Nickisch (2016)).

6. Conclusions

With the PVC function as acquisition function, this paper has promoted the adaptive BC to be implemented in a parallel computation scheme by using multiple multimodal optimization algorithms. The improvement is initialized by the multimodal behavior of the PVC acquisition function, which is first systematically investigated. Based on the multimodal behavior and the mathematical interpretation of the PVC function, it is then concluded that the multiple local maxima all have important contributions to the integration accuracy, and thus four multimodal optimization algorithms, including the newly developed SQNM in this work, have been introduced for searching the multiple local maxima. These algorithms are then embedded into the adaptive BC method for parallel implementation.

The performance of the four multimodal optimization methods for parallel implementation is then compared with the one equipped with k-means clustering algorithm, the one without parallel computation and also the sparse grid cubature by two numerical examples and two engineering examples. It is shown that the parallelized adaptive BC procedures with multimodal optimization algorithms out-perform the original adaptive BC and the one with k-means clustering algorithm in terms of both accuracy and efficiency for most multi-dimensional examples, resulting from the fact that the contribution of the local maxima of PVC function to the posterior variance of integral is bigger than other points. It is also seen that, with any one of the four multimodal optimization equipped, the parallel implementation of adaptive BC requires much less iteration steps for the same level of estimation accuracy, indicating that much less computer time is consumed if multiple works can be assigned for estimation.

In this work, the PVC acquisition function is used for all the developments. However, there are also other acquisition functions being developed (see e.g., Refs. Osborne *et al.,* (2012); Santiago *et al.,* (2020); Gessner *et al.,* (2020)), and their performance for parallel computation may differ a lot. This will be investigated in the future work.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under grant number 72171194, and the Sino-German Mobility Programme under grant number M-0175 (2021-2024).

References

Beasley, D., Bull, D. R. and Martin, R. R. (1993), 'A sequential niche technique for multimodal function optimization', *Evolutionary Computation*, 1(2), pp. 101–125. doi:10.1162/evco.1993.1.2.101.

Briol, F.-X. (2018), Statistical Computation with Kernels, PhD thesis, University of Warwick.

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., Sejdinovic, D. *et al.,* (2019), 'Probabilistic integration: A role in statistical computation?', *Statistical Science*, 34(1), pp. 1–22. doi:10.48550/arXiv.1512.00933.

Chen, Z., Li, G., He, J., Yang, Z. and Wang, J. (2022), 'A new parallel adaptive structural reliability analysis method based on importance sampling and K-medoids clustering', *Reliability Engineering and System Safety*, 218, p. 108124. doi:10.1016/j.ress.2021.108124.

Cockayne, J., Oates, C. J., Sullivan, T. J. and Girolami, M. (2019), 'Bayesian probabilistic numerical methods', *SIAM Review*, 61(4), pp. 756–789. doi:10.1137/17M1139357.

Dang, C., Wei, P., Faes, M. G., Valdebenito, M. A. and Beer, M. (2022), 'Parallel adaptive Bayesian quadrature for rare event estimation', *Reliability Engineering and System Safety*, 225, p. 108621. doi:10.1016/j.ress.2022.108621.

Das, S., Maity, S., Qu, B.-Y. and Suganthan, P. N. (2011), 'Real-parameter evolutionary multimodal optimization A survey of the state-of-the-art', *Swarm and evolutionary computation*, 1(2), pp. 71–88. doi:10.1016/j.swevo.2011.05.005.

Diaconis, P. (1988), *Bayesian Numerical Analysis*, Statistical Decision Theory and Related Topics, IV, Vol. 1 (West Lafayette, Ind., 1986).

Du, X., Zhao, G., Wang, W., Guo, M., Zhang, R. and Yang, J. (2020), 'NLIGA: A MATLAB framework for nonlinear isogeometric analysis', *Computer Aided Geometric Design*, 80, p. 101869. doi:10.1016/j.cagd.2020.101869.

Fuhg, J. N., Fau, A. and Nackenhorst, U. (2021), 'State-of-the-art and comparative review of adaptive sampling methods for Kriging', *Archives of Computational Methods in Engineering*, 28(4), pp. 2689–2747. doi:10.1007/s11831-020-09474-6.

Gessner, A., Gonzalez, J. and Mahsereci, M. (2020), Active multi-information source Bayesian quadrature, *in* 'Uncertainty in Artificial Intelligence', PMLR, pp. 712–721.

Ghahramani, Z. and Rasmussen, C. E. (2002), 'Bayesian Monte Carlo', *International Journal for Numerical Methods in Engineering*, .

Ghosh, S., Pandita, P., Atkinson, S., Subber, W., Zhang, Y., Kumar, N. C., Chakrabarti, S. and Wang, L. (2020), 'Advances in Bayesian probabilistic modeling for industrial applications', *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 6(3), p. 030904. doi:10.1115/1.4046747.

Hennig, P., Osborne, M. A. and Girolami, M. (2015), 'Probabilistic numerics and uncertainty in computations', *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 471(2179), p. 20150142. doi:10.1098/rspa.2015.0142.

Huang, D., Allen, T. T., Notz, W. I. and Zeng, N. (2006), 'Global optimization of stochastic black-box systems via sequential Kriging meta-models', *Journal of Global Optimization*, 34(3), pp. 441–466. doi:10.1007/s10898-005-2454-3.

Kanagawa, M. and Hennig, P. (2019), 'Convergence guarantees for adaptive Bayesian quadrature methods', *Advances in Neural Information Processing Systems*, 32.

Kennedy, J. and Eberhart, R. (1995), Particle swarm optimization, *in* 'Proceedings of IEEE International Conference on Neural Networks', Vol. 4, pp. 1942–1948.

Li, J.-P., Balazs, M. E., Parks, G. T. and Clarkson, P. J. (2002), 'A species conserving genetic algorithm for multimodal function optimization', *Evolutionary computation*, 10(3), pp. 207–234. doi:10.1162/106365602760234081.

Li, R., Liang, X. and Peng, Q. (2021), 'A selection strategy for Kriging based design of experiments by spectral clustering and learning function', *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 7(2), p. 020902. doi:10.1115/1.4050160.

Mengshoel, O. J. and Goldberg, D. E. (2008), 'The crowding approach to niching in genetic algorithms', *Evolutionary Computation*, 16(3), pp. 315–354. doi:10.1162/evco.2008.16.3.315.

Mortazavi, A. (2021), 'Bayesian interactive search algorithm: A new probabilistic swarm intelligence tested on mathematical and structural optimization problems', *Advances in Engineering Software*, 155, p. 102994. doi:10.1016/j.advengsoft.2021.102994.

Oates, C. J. and Sullivan, T. J. (2019), 'A modern retrospective on probabilistic numerics', *Statistics and Computing*, 29(6), pp. 1335–1351. doi:10.1007/s11222-019-09902-z.

O'Hagan, A. (1991), 'Bayes-hermite quadrature', *Journal of Statistical Planning and Inference*, 29(3). doi:10.1016/0378-3758(91)90002-V.

Osborne, M., Garnett, R., Ghahramani, Z., Duvenaud, D. K., Roberts, S. J. and Rasmussen, C. E. (2012), Active learning of model evidence using Bayesian quadrature, *in* 'Advances in Neural Information Processing Systems 25', Vol. 25, pp. 46–54.

Papaioannou, I., Daub, M., Drieschner, M., Duddeck, F., Ehre, M., Eichner, L., Eigel, M., Götz, M., Graf, W., Grasedyck, L. *et al.,* (2019), 'Assessment and design of an engineering structure with polymorphic uncertainty quantification', *GAMM-Mitteilungen*, 42(2), p. e201900009. doi:10.1002/gamm.201900009.

Rasmussen, C. E. and Ghahramani, Z. (2003), Bayesian Monte carlo, *in* 'Advances in neural information processing systems: Vol. 15', MIT Press, Cambridge, pp. 505–512.

Rasmussen, C. E. and Nickisch, H. (2016), The gpml toolbox version 4.2, *in* 'Technical Documentation'.

Rasmussen, C. E. and Williams, C. (2006), 'Gaussian processes for machine learning, vol. 1', *MIT press*, 39, pp. 40–43. doi:10.7551/mitpress/3206.001.0001.

Santiago, J. L., Gmez, D. D., Arregui, V. E., Martino, L. and Fernandez, F. L. (2020), 'Adaptive quadrature schemes for Bayesian inference via active learning', *DES - Working Papers. Statistics and Econometrics. WS*, . doi:10.1109/ACCESS.2020.3038333.

Seo, J.-H., Im, C.-H., Heo, C.-G., Kim, J.-K., Jung, H.-K. and Lee, C.-G. (2006), 'Multimodal function optimization based on particle swarm optimization', *IEEE Transactions on Magnetics*, 42(4), pp. 1095–1098. doi:10.1109/TMAG.2006.871568.

Ser, J. D., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D., Das, S., Suganthan, P. N., Coello, C. A. C. and Herrera, F. (2019), 'Bio-inspired computation: Where we stand and what's next', *Swarm and evolutionary computation*, 48, pp. 220–250. doi:10.1016/j.swevo.2019.04.008.

Sinsbeck, M., Cooke, E. and Nowak, W. (2021), 'Sequential design of computer experiments for the computation of Bayesian model evidence', *SIAM/ASA Journal on Uncertainty Quantification*, 9(1), pp. 260–279. doi:10.1137/20M132043.

Snoek, J., Larochelle, H. and Adams, R. P. (2012), 'Practical Bayesian optimization of machine learning algorithms', *Advances in Neural Information Processing Systems*, 4. doi:10.48550/arXiv.1206.2944.

Song, J., Wei, P., Valdebenito, M. A., Faes, M. and Beer, M. (2022), 'Data-driven and active learning of variance-based sensitivity indices with Bayesian probabilistic integration', *Mechanical Systems and Signal Processing*, 163, p. 108106. doi:10.1016/j.ymssp.2021.108106.

Song, J., Wei, P., Valdebenito, M. and Beer, M. (2021), 'Active learning line sampling for rare event analysis', *Mechanical Systems and Signal Processing*, 147, p. 107113. doi:10.1016/j.ymssp.2020.107113.

Teixeira, R., Nogal, M., OConnor, A. and Martinez-Pastor, B. (2020), 'Reliability assessment with density scanned adaptive Kriging', *Reliability Engineering and System Safety*, 199, p. 106908. doi:10.1016/j.ress.2020.106908.

Trangenstein, J. A. (2018), *Scientific Computing: Vol. I-Linear and Nonlinear Equations*, Vol. 18, Springer.

Tronarp, F., Kersting, H., Srkk, S. and Hennig, P. (2019), 'Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: a new perspective', *Statistics and Computing*, 29(6), pp. 1297–1315. doi:10.1007/s11222-019-09900-1.

Wei, P., Hong, F., Phoon, K.-K. and Beer, M. (2021), 'Bounds optimization of model response moments: a twin-engine Bayesian active learning method', *Computational Mechanics*, 67(5), pp. 1273–1292. doi:10.1007/s00466-021-01977-8.

Wei, P., Zhang, X. and Beer, M. (2020), 'Adaptive experiment design for probabilistic integration', *Computer Methods in Applied Mechanics and Engineering*, 365, p. 113035. doi:10.1016/j.cma.2020.113035.

Zeng, B., Gao, L. and Li, X. (2017), Whale swarm algorithm for function optimization, *in* 'International Conference on Intelligent Computing', Springer, pp. 624–639. doi:10.1007/978-3-319-63309-1_55.

Zeng, B., Li, X., Gao, L., Zhang, Y. and Dong, H. (2020), 'Whale swarm algorithm with the mechanism of identifying and escaping from extreme points for multimodal function optimization', *Neural Computing and Applications*, 32(9), pp. 5071–5091. doi:10.1007/s00521-018-3949-4.

Zhou, T. and Peng, Y. (2020), 'Adaptive Bayesian quadrature based statistical moments estimation for structural reliability analysis', *Reliability Engineering and System Safety*, 198, p. 106902. doi:10.1016/j.ress.2020.106902.