SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. 10, No. 1, pp. 294–316

© 2022 Society for Industrial and Applied Mathematics
and American Statistical Association

Cross-Validation–based Adaptive Sampling for Gaussian Process Models*

Hossein Mohammadi†, Peter Challenor†, Daniel Williamson†, and Marc Goodfellow†

Abstract. In many real-world applications, we are interested in approximating black-box, costly functions as accurately as possible with the smallest number of function evaluations. A complex computer code is an example of such a function. In this work, a Gaussian process (GP) emulator is used to approximate the output of complex computer code. We consider the problem of extending an initial experiment (set of model runs) sequentially to improve the emulator. A sequential sampling approach based on leave-one-out (LOO) cross-validation is proposed that can be easily extended to a batch mode. This is a desirable property since it saves the user time when parallel computing is available. After fitting a GP to training data points, the expected squared LOO (ES-LOO) error is calculated at each design point. ES-LOO is used as a measure to identify important data points. More precisely, when this quantity is large at a point it means that the quality of prediction depends a great deal on that point and adding more samples nearby could improve the accuracy of the GP. As a result, it is reasonable to select the next sample where ES-LOO is maximized. However, ES-LOO is only known at the experimental design and needs to be estimated at unobserved points. To do this, a second GP is fitted to the ES-LOO errors, and where the maximum of the modified expected improvement (EI) criterion occurs is chosen as the next sample. EI is a popular acquisition function in Bayesian optimization and is used to trade off between local and global search. However, it has a tendency towards exploitation, meaning that its maximum is close to the (current) "best" sample. To avoid clustering, a modified version of EI, called pseudoexpected improvement, is employed which is more explorative than EI yet allows us to discover unexplored regions. Our results show that the proposed sampling method is promising.

Key words. adaptive sampling, computer experiment, leave-one-out cross-validation, Gaussian processes

AMS subject classifications. 60G15, 62K99, 62L05, 62M20

DOI. 10.1137/21M1404260

1. Introduction.

In many real-world applications, we are interested in predicting the output of complex computer models (or simulators) such as high fidelity numerical solvers. The reason is that such models are computationally intensive and we cannot use them to perform analysis that requires very many runs. One way to predict the model output is to use surrogate models also known as emulators which are constructed based on a limited number of simulation runs. Surrogates are fast to run, and the analysis can be carried out on them; see, e.g., [34, 59, 4]. Among different classes of surrogate models, Gaussian process (GP) emulators [48] have gained increasing attention due to their statistical properties such as computational tractability and flexibility. GPs provide a flexible paradigm to approximate

*Received by the editors March 10, 2021; accepted for publication (in revised form) October 4, 2021; published electronically February 28, 2022.
https://doi.org/10.1137/21M1404260
Funding: This work was funded in part by the Engineering and Physical Sciences Research Council via grant EP/N014391/1 and in part by the Alan Turing Institute.
†College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, UK (h.mohammadi@exeter.ac.uk, P.G.Challenor@exeter.ac.uk, d.williamson@exeter.ac.uk, m.goodfellow@exeter.ac.uk).

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

any smooth, continuous function [43] thanks to the variety of covariance kernels available. Most importantly, the GP prediction is equipped with an estimation of uncertainty which reflects the accuracy of the prediction.

One factor that heavily affects the accuracy of emulators is the location of the training data, also called the design [56, 27]. In this context, the design of computer experiments has become an integral part of the analysis of computer experiments [51, 52]. Generally speaking, such design can be performed in a *one-shot* or *adaptive* manner [35]. In the former all samples are chosen at once, while in the latter the points are selected sequentially using information from the emulator and the existing data. Examples of one-shot design of experiment (DoE) methods are the Latin hypercube [40], full factorial [8], orthogonal array [44], and minimax and maximin-distance designs [23]. A potential drawback of one-shot DoEs is that they may result in under/oversampling and can waste computational resources [54, 15]. However, this is not the case for adaptive approaches where we can stop the computationally expensive sampling process as soon as the emulator reaches an acceptable level of accuracy. Moreover, with adaptive sampling it is possible to take more samples in "interesting" regions where, e.g., the underlying function is highly nonlinear or exhibits abrupt changes. This paper focuses on GP-based adaptive sampling where an initial design is extended sequentially to improve the emulator. The initial DoE is often *space-filling* meaning that the points are scattered uniformly over the input space. We refer the reader to [47, 26] and references therein for more information on space-filling designs.

There are various GP-based adaptive sampling methods which can be categorized according to their selection criteria, i.e., the strategies to find future designs. The readers are referred to [15, 35] for a comprehensive review of the existing methods. An intuitive criterion is the built-in predictive variance of GPs, also known as the prediction uncertainty or mean squared error (MSE). The idea is that the predictive variance is regarded as an estimation of the "real" prediction error and a point with the maximum uncertainty is taken as the next experimental design [38, 22]. The predictive variance increases away from the data points. It is highly probable that sampling based on the MSE criterion gives some sort of space-filling design which can be achieved using one-shot techniques. Note that the MSE criterion (see (2.4)) depends only on the location of samples and not the output values. Thus, the MSE-based sampling strategy can be regarded as nonadaptive in the sense that it does not consider output information. Moreover, MSE is large on the boundaries of the input space, and that can lead to taking a lot of samples on the boundaries. However, this is not desirable in many situations especially when the main characteristics of the true function appear inside the interior region and when the dimension of the input space is high. The integrated mean square error (IMSE) is a variant of the MSE criterion and selects a new point if adding that point to the existing design minimizes the integral of the MSE [51, 45]. However, computing IMSE can be cumbersome, especially in high dimensions.

Maximum entropy is another common selection criterion in the adaptive sampling paradigm [55, 28]. It is equivalent to the maximum MSE criteria under certain circumstances [22, 30]. As a result, an adaptive sampling strategy based on maximum entropy tends to place many points at the borders of the input space [28]. This issue can be mitigated using *mutual information* (MI) as proposed by Krause, Singh, and Guestrin [29] in the sensor placement problem. The MI of two random variables is a measure of reduction in the uncertainty of one

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

random variable through observing the other one. A sequential design approach is developed in [5] where the MI criterion is modified by introducing an extra parameter, called a *nugget*, to the correlation matrix of the GP (in the denominator); see Appendix A. The inclusion of the nugget parameter prevents selecting a new sample close to the current design. The algorithm called MICE (mutual information for computer experiments) is then used to emulate a simulator.

The leave-one-out (LOO) cross-validation (CV) error defines another class of adaptive sampling criterion [32, 31, 2, 36]. To obtain the LOO error at an experimental design we remove that point from the training data set. Then, a GP is fitted to the remaining samples, and the response at the left out point is predicted. The difference between the predicted and actual response serves as the LOO error. A relatively small error indicates that the prediction accuracy in a vicinity of the removed point is high and there is good information about the true function there. On the other hand, a comparably large error means that the removed point has a huge impact on the accuracy of the emulator and, hence, we need more samples in the nearby region to reduce the errors. In [61] scores obtained by CV are used to identify regions of distinct model behavior and specify mixture of covariance functions for GP emulators. Using the LOO errors as a sampling criterion has several advantages. First, it provides actual prediction error at the design points. Second, it is model-independent and can be achieved by any surrogate model. For example, in [7] a methodology based on the LOO CV is proposed to estimate the prediction uncertainty of any surrogate model, either deterministic or probabilistic. Third, computing the LOO errors is not expensive in terms of computational cost [12]. However, the LOO errors are not determined everywhere in the input space and only defined at locations where we have preexisting model runs.

It is worth mentioning that CV is used for other purposes (such as model selection/fitting and diagnostic and parameter inference) than adaptive sampling. A survey of CV strategies on the model selection can be found in [1, 65]. [17] suggested an efficient (multiple-fold) CV expression for GP model fitting and diagnostics. The CV technique is employed by [37] to estimate the covariance parameters of a GP with inequality constraints. Bachoc [3] studied the capability of CV (and maximum likelihood) in estimating the parameters of a GP with a misspecified covariance structure. A probabilistic version of CV is proposed in [39] and is reported robust against mismatch between the data and chosen model. Viana, Haftka, and Steffen [60] proposed to predict computer codes by an ensemble of surrogate models such that CV serves as a performance measure of surrogates to select the best one. [33] suggested to use CV for the optimal basis functions selection in designing the GP (prior) mean. A methodology for the Bayesian time series analysis is presented in [9] where the forecast of future observations is performed via CV. In [63] the human gait pattern kinematics is predicted with a GP regression whose validation is done by CV.

This paper proposes an adaptive sampling DoE relying on the LOO CV method to build GP emulators as accurately as possible for deterministic computer codes over the entire domain. The proposed method has a few parameters to be tuned and can be extended to a batch mode where at each iteration a set of inputs is selected for evaluation. This is an important property as it saves the user time when parallel computing is available [62]. The remainder of the paper is organized as follows. In the next section, the statistical methodology of GP emulators is briefly reviewed. Section 3 introduces the proposed adaptive sampling approach and

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

its extension to batch mode. Section 4 presents numerical experiments where the predictive performance of our algorithm is tested. Finally, the paper's conclusion is in section 5.

2. GP models. First we look at GP emulators and their statistical background. Let the underlying function of a deterministic complex computer code be given by $f : \mathcal{D} \mapsto \mathbb{R}$ in which $\mathcal{D}$ is a compact set in $\mathbb{R}^d$. Suppose $\mathcal{A} = \{\mathbf{X}_n, \mathbf{y}_n\}$ is a training data set, where $\mathbf{X}_n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ and $\mathbf{y}_n = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))^\top$ represent n locations in the input space $\mathcal{D}$ and the corresponding outputs (responses/observations), respectively. Let $(Z_0(\mathbf{x}))_{\mathbf{x}\in\mathcal{D}}$ be the GP by which we want to model f. In this framework, it is assumed that $\mathbf{y}_n$ has a multivariate normal distribution given by

$$\mathbf{y}_n \sim \mathcal{N}\left(m_0(\mathbf{X}_n), k_0(\mathbf{X}_n, \mathbf{X}_n)\right), \tag{2.1}$$

where m_0 and k_0 are the (preselected) mean and covariance functions of $(Z_0(\mathbf{x}))_{\mathbf{x}\in\mathcal{D}}$. Without loss of generality, we assume that the mean function is a constant: $m_0(x) = \mu$. The positive semidefinite covariance function k_0 plays an important role in GP modeling; assumptions about the underlying function such as differentiability or periodicity are encoded through kernels. The Matérn family of kernels are commonplace in computer experiments and (in the univariate case) are defined as

$$k_0(x, x') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\theta}|x - x'|\right)^\nu B_\nu \left(\frac{\sqrt{2\nu}}{\theta}|x - x'|\right), \tag{2.2}$$

where $\Gamma(\cdot)$ is the gamma function and $B_\nu(\cdot)$ denotes the modified Bessel function of the second kind of order ν. The parameter ν regulates the degree of smoothness of the GP sample paths/functions such that a process with the Matérn kernel of order ν is $\lceil \nu - 1 \rceil$ times differentiable [48]. The positive parameters σ^2 and θ are referred to as the *process variance* and *correlation length scale*, respectively. These parameters are usually unknown and need to be estimated from data. We refer the reader to [48, 16] for a more detailed explanation on parameter estimation techniques.

Given that the unknown parameters are estimated, the posterior predictive distribution relying on $Z_n(\mathbf{x}) = Z_0(\mathbf{x}) \mid \mathcal{A}$ can be calculated. The posterior mean and covariance at a generic location $\mathbf{x}$ have closed-form expressions as [48]

$$m_n(\mathbf{x}) = \hat{\mu} + \mathbf{k}(\mathbf{x})^\top \mathbf{K}^{-1}(\mathbf{y}_n - \hat{\mu}\mathbf{1}), \tag{2.3}$$

$$k_n(\mathbf{x}, \mathbf{x}') = k_0(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x})^\top \mathbf{K}^{-1}\mathbf{k}(\mathbf{x}') + \frac{\left(1 - \mathbf{1}^\top \mathbf{K}^{-1}\mathbf{k}(\mathbf{x})\right)\left(1 - \mathbf{1}^\top \mathbf{K}^{-1}\mathbf{k}(\mathbf{x}')\right)}{\mathbf{1}^\top \mathbf{K}^{-1}\mathbf{1}}. \tag{2.4}$$

Here, $\mathbf{k}(\mathbf{x}) = (k_0(\mathbf{x}, \mathbf{x}_1), \ldots, k_0(\mathbf{x}, \mathbf{x}_n))^\top$ is the vector of covariances between $Z_0(\mathbf{x})$ and $Z_0(\mathbf{x}_i)$s, $\mathbf{1}$ is a vector of ones, and $\mathbf{K}$ is an $n \times n$ covariance matrix with elements $\mathbf{K}_{ij} = k_0(\mathbf{x}_i, \mathbf{x}_j)$ for all $1 \le i, j \le n$. The last term in (2.4) reflects the additional uncertainty concerning the estimation of μ. We refer the reader to [49] where the "ordinary kriging" is described. The predictive variance $s_n^2(\mathbf{x}) = k_n(\mathbf{x}, \mathbf{x})$ determines the uncertainty associated with the prediction at $\mathbf{x} \in \mathcal{D}$ and is a measure of the prediction accuracy. In adaptive sampling based on the MSE criterion, the new sample is the point that maximizes $s_n^2(\mathbf{x})$.

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

3. Proposed adaptive sampling method. Before introducing our method, it is worth mentioning that an "efficient" adaptive sampling strategy should meet the conditions below [35].

(i) *Local exploitation* that allows us to add more points in interesting areas discovered so far.
(ii) *Global exploration* by which unexplored domain regions can be detected.
(iii) *Trade-off between local exploitation and global exploration* which balances the previous two objectives using a suitable measure.

The proposed adaptive sampling approach uses *expected squared LOO error* (see subsection 3.1) for local exploitation. The global exploration and the trade-off between the local and global search are driven by the *pseudoexpected improvement* criterion introduced in subsection 3.2.

3.1. Expected squared LOO CV error. We wish to improve the GP emulator $Z_n(\mathbf{x})$ constructed on the training data set $\mathcal{A}$ using the LOO CV method. To do so, the first step is to obtain the LOO errors. Let $e_L(\mathbf{x}_i)$ denote the LOO CV error at the design sites $\mathbf{x}_i, i = 1, \dots, n$. The computation of $e_L(\mathbf{x}_i)$ relies on the GP $Z_{n,-i}(\mathbf{x})$ which is obtained by conditioning $Z_0(\mathbf{x})$ on all observations except the ith one: $Z_{n,-i}(\mathbf{x}) = Z_0(\mathbf{x}) \mid \mathbf{y}_n \setminus \{f(\mathbf{x}_i)\}$. The predictive mean and variance of $Z_{n,-i}(\mathbf{x})$ are shown by $m_{n,-i}(\mathbf{x})$ and $s^2_{n,-i}(\mathbf{x})$, respectively. The LOO CV error is then calculated as

$$e_L(\mathbf{x}_i) = |m_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)|, \tag{3.1}$$

which can be regarded as the sensitivity of the emulator to the left out point $\mathbf{x}_i$. As a result, the idea of adding new samples near the points with large LOO errors is used in several adaptive sampling works; see, e.g., [32, 2]. In this work, the unknown parameters of $Z_n(\mathbf{x})$ are estimated at each iteration when a new point is added to the existing data. However, we do not estimate the parameters of $Z_{n,-i}(\mathbf{x})$ to alleviate the computational burden. To this end, the estimated parameters of $Z_n(\mathbf{x})$ are used in $Z_{n,-i}(\mathbf{x})$. Moreover, (3.1) can be calculated efficiently using the formula proposed in [12].

The LOO error $e_L(\mathbf{x}_i)$ only accounts for the difference between the predictive mean and the real value at $\mathbf{x}_i$ and can be a misleading selection criterion in some situations. Figure 1 (left panel) shows an example where the prediction at $x_5 = 0$ is equal to the true value there and therefore $e_L(x_5) = 0$. This means that the chance of adding a new sample near the fifth data point is low although it is in a crucial region. The above mentioned problem can be mitigated using expected squared LOO (ES-LOO) error, which provides more information than the LOO error about the sensitivity of the emulator to the design points. More precisely, ES-LOO accounts for both the prediction uncertainty and the difference between the prediction and the true value, as described below.

Let $\mathcal{E}_L(\mathbf{x}_i)$ represent the value of (normalized) ES-LOO error at the design site $\mathbf{x}_i$. It is defined as

$$\mathcal{E}_L(\mathbf{x}_i) = \frac{\mathbb{E}\left[(Z_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i))^2\right]}{\sqrt{\mathbb{V}\mathrm{ar}\left((Z_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i))^2\right)}}, \tag{3.2}$$

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

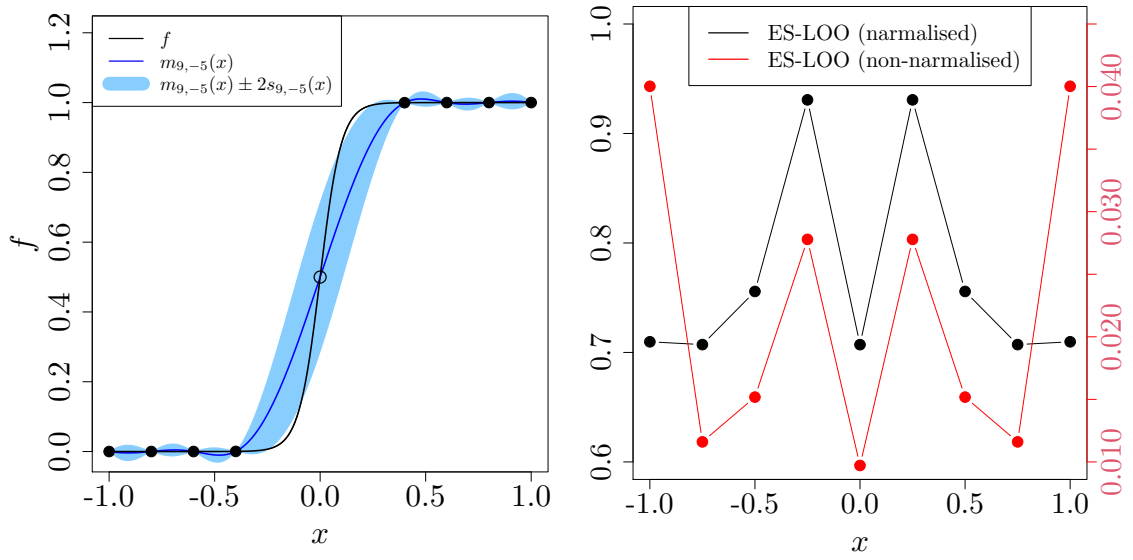

Figure 1. *Left: The LOO error (i.e., (3.1)) can be misleading as a selection criterion. Removing the fifth sample, i.e.,* $(0, 0.5)$*, does not change the prediction at* $x = 0$ *and the LOO error remains zero there:* $|m_{9,-5}(x_5) - f(x_5)| = 0$*. This means that the point* $(0, 0.5)$ *does not have any influence on the emulator, while* f *has a large gradient there. However, the ES-LOO error which accounts for the prediction uncertainty is not zero at* x_5*. Right: Normalized (black) vs. nonnormalized (red) ES-LOO errors for the sigmoid function visualized on the left panel. The value of nonnormalized ES-LOO is maximum at the endpoints where adding new samples does not improve the emulator. The true function is* $f(x) = \frac{1}{1+\exp(-20x)}$.

where

$$\mathbb{E}\left[\left(Z_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2\right] = s^2_{n,-i}(\mathbf{x}_i) + \left(m_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2, \tag{3.3}$$

$$\mathbb{V}\mathrm{ar}\left(\left(Z_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2\right) = 2s^4_{n,-i}(\mathbf{x}_i) + 4s^2_{n,-i}(\mathbf{x}_i)\left(m_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2. \tag{3.4}$$

The latter is used in (3.2) to normalize ES-LOO values. It is recommended in [31, 19] that standardizing the LOO errors yields a better measure for adaptive sampling. Our experiments also suggest that normalized ES-LOO should be preferred over nonnormalized one. For example, in Figure 1 (right panel) the normalized (black) and nonnormalized (red) ES-LOO are shown for the sigmoid function. As can be seen, the nonnormalized ES-LOO error is maximum at the endpoints where adding new samples does not improve the accuracy of the emulator significantly. In the standardized case, however, the value of ES-LOO is moderated at the endpoints since the term $s_{n,-i}(\mathbf{x}_i)$ plays an important role in (3.4).

To see how (3.3) and (3.4) are obtained we first note that

$$Z_{n,-i}(\mathbf{x}_i) \sim \mathcal{N}\left(m_{n,-i}(\mathbf{x}_i), s^2_{n,-i}(\mathbf{x}_i)\right), \tag{3.5}$$

as is shown with an example in Figure 2. By standardizing the above equation we reach

$$\frac{Z_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)}{s_{n,-i}(\mathbf{x}_i)} \sim \mathcal{N}\left(\frac{m_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)}{s_{n,-i}(\mathbf{x}_i)}, 1\right), \tag{3.6}$$

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

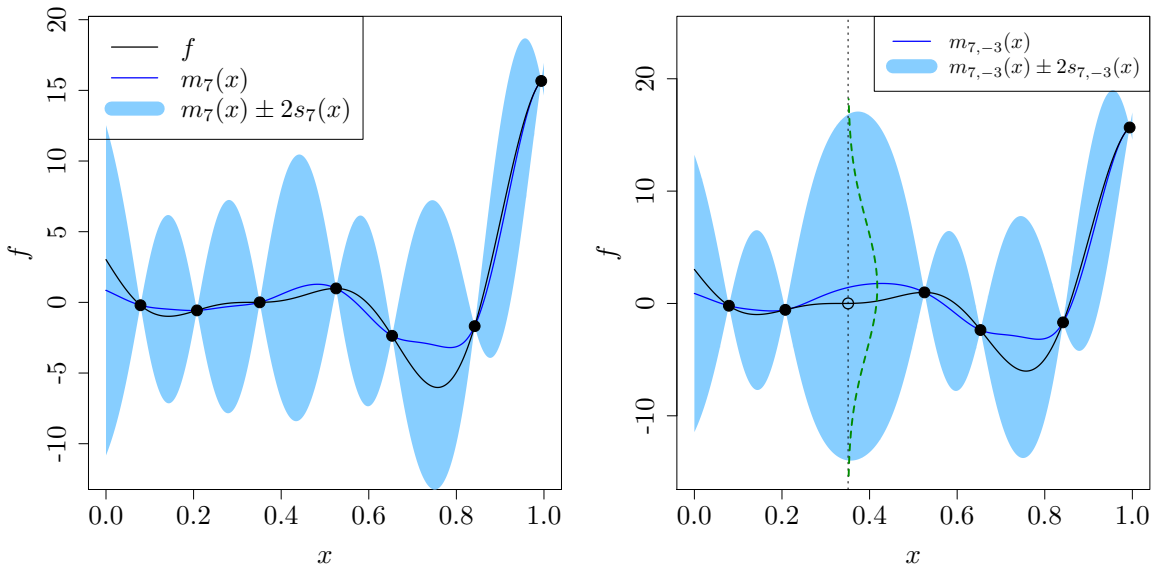

Figure 2. *Left: GP prediction (blue) with 95% credible intervals (shaded) based on 7 observations from the true function (black). Right: GP prediction based on the training data except the third one where the GP has a normal distribution specified by (3.5).*

in which the square of the left-hand side is a random variable with noncentral chi-square distribution characterized by

$$\left(\frac{Z_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)}{s_{n,-i}(\mathbf{x}_i)}\right)^2 \sim \chi'^2\left(\kappa = 1, \lambda = \left(\frac{m_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)}{s_{n,-i}(\mathbf{x}_i)}\right)^2\right). \tag{3.7}$$

Here, κ and λ are the degrees of freedom and noncentrality parameter, respectively.[1] As a result

$$\mathbb{E}\left[\left(\frac{Z_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)}{s_{n,-i}(\mathbf{x}_i)}\right)^2\right] = 1 + \left(\frac{m_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)}{s_{n,-i}(\mathbf{x}_i)}\right)^2, \tag{3.8}$$

$$\mathbb{V}\mathrm{ar}\left(\left(\frac{Z_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)}{s_{n,-i}(\mathbf{x}_i)}\right)^2\right) = 2\left(1 + 2\left(\frac{m_{n,-i}(\mathbf{x}_i) - f(\mathbf{x}_i)}{s_{n,-i}(\mathbf{x}_i)}\right)^2\right). \tag{3.9}$$

Finally, if the expectation and variance in the above equations are multiplied by $s^2_{n,-i}(\mathbf{x}_i)$ and $s^4_{n,-i}(\mathbf{x}_i)$, respectively, we reach (3.3) and (3.4).

The analytical expression of (3.3) is similar to the *expected improvement for global fit (EIGF)* infill sampling criterion proposed by Lam [30]. In section 4, we compare the predictive

[1]Suppose $X_1, \ldots, X_\kappa$ are κ independent random normal variables such that $X_i \sim \mathcal{N}(\mu_i, 1), 1 \le i \le \kappa$. Then, $\sum_{i=1}^{\kappa} X_i^2 \sim \chi'^2\left(\kappa, \lambda = \sum_{i=1}^{\kappa} \mu_i^2\right)$ has a noncentral chi-square distribution with mean $\kappa + \lambda$ and variance $2(\kappa + 2\lambda)$.

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

performance of our method with EIGF on several test functions. In EIGF, the improvement at an arbitrary point $\mathbf{x}$ is given by

$$IGF(\mathbf{x}) = \left(Z_n(\mathbf{x}) - f(\mathbf{x}_i^*)\right)^2, \tag{3.10}$$

where $f(\mathbf{x}_i^*)$ is the response at the location $\mathbf{x}_i^*$ which is closest (in Euclidean distance) to $\mathbf{x}$. The EIGF criterion is the expected value of $IGF(\mathbf{x})$ and takes the following form:

$$EIGF(\mathbf{x}) = \mathbb{E}[IGF(\mathbf{x})] = \left(m_n(\mathbf{x}) - f(\mathbf{x}_i^*)\right)^2 + s_n^2(\mathbf{x}). \tag{3.11}$$

Using EIGF as the selection criterion, the next sample is chosen where EIGF is maximum:

$$\mathbf{x}_{n+1} = \underset{\mathbf{x}\in\mathcal{D}}{\operatorname{argmax}}\ EIGF(\mathbf{x}).$$

3.2. Proposed selection criterion. Since the magnitude of $\mathcal{E}_L(\mathbf{x}_i)$ reflects the sensitivity of the emulator to the loss of information provided by the function evaluation at $\mathbf{x}_i$, it is reasonable to choose the next sample where ES-LOO is maximum. However, this quantity is only defined at the training data, while we need to look for the next design point out-of-sample. In this work, we extend the ES-LOO error to be a function defined over the whole domain, $\mathcal{E}_L(\mathbf{x}), \mathbf{x} \in \mathcal{D}$, that we have observed at the design points, and we model this function with a GP. The interpretation of $\mathcal{E}_L(\mathbf{x})$ is the value of the ES-LOO error we think we would see if $(\mathbf{x}, f(\mathbf{x}))$ were part of our data set. The GP model to estimate $\mathcal{E}_L(\mathbf{x})$ at unobserved locations is denoted by $Z_n^e(\mathbf{x})$ whose predictive mean and variance are indicated by $m_n^e(\mathbf{x})$ and $s_n^e(\mathbf{x})$, respectively. The training set for the second GP is $\{\mathbf{X}_n, \mathbf{y}_n^e\}$ where $\mathbf{y}_n^e = (\mathcal{E}_L(\mathbf{x}_1), \ldots, \mathcal{E}_L(\mathbf{x}_n))^\top$.

After estimating $\mathcal{E}_L(\mathbf{x})$, we can find its maximum applying techniques in surrogate-based optimization. In this framework, a naive approach is to maximize $m_n^e(\mathbf{x})$. However, this simple strategy does not define a valid optimization scheme due to *overexploitation* [24] meaning that the new samples are taken very close to the points with a large ES-LOO error. To overcome this problem, we need to take into account $s_n^e(\mathbf{x})$ as the exploration component in the course of the search. To this end, we employ *expected improvement (EI)* which is one of the most common acquisition functions in Bayesian optimization [10, 24], to trade-off between exploration and exploitation. It is expressed via

$$EI(\mathbf{x}) = \begin{cases} \left(m_n^e(\mathbf{x}) - \max(\mathbf{y}_n^e)\right)\Phi(u) + s_n^e(\mathbf{x})\phi(u) & \text{if} \quad s_n^e(\mathbf{x}) > 0, \\ 0 & \text{if} \quad s_n^e(\mathbf{x}) = 0, \end{cases} \tag{3.12}$$

where $u = \frac{m_n^e(\mathbf{x}) - \max(\mathbf{y}_n^e)}{s_n^e(\mathbf{x})}$ and $\phi(\cdot)$ and $\Phi(\cdot)$ represent the PDF and cumulative distribution function of the standard normal distribution, respectively. EI is a nonnegative, parameter-free function and is zero at the data points. However, it is shown that EI is biased towards exploitation especially at the beginning of the search [53, 25, 46]. As a result, if EI is used to find the maximum of $\mathcal{E}_L(\mathbf{x})$ at each iteration, the new samples are clustered. This is illustrated by an example in Figure 3 where the black circles are the initial design and the red circles

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

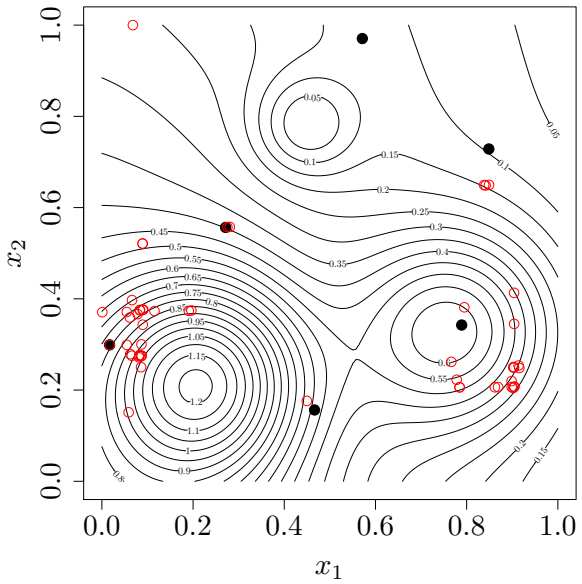

Figure 3. *Adaptive designs (red circles) are obtained by maximizing $\mathcal{E}_L(\mathbf{x})$ using the EI criterion. The filled circles are the initial design, and the true function is Franke's function. EI tends towards exploitation, and as a result clustering occurs.*

represent the new samples selected based on the EI criterion. Clusters of the new points can be detected on the contour plot of Franke's function (its analytic expression is given in Appendix B) due to the tendency of EI towards exploitation.

In this paper, *pseudoexpected improvement (PEI)* [64] is considered as the selection criterion which has a better exploration property than EI. PEI is obtained by multiplying EI by a repulsion function (RF): $PEI(\mathbf{x}) = EI(\mathbf{x})RF(\mathbf{x})$. The RF is defined as

$$RF(\mathbf{x}; \mathbf{X}_n) = \prod_{i=1}^{n} \left[1 - \mathbb{C}\text{orr}\left(Z_n^e(\mathbf{x}), Z_n^e(\mathbf{x}_i)\right)\right], \tag{3.13}$$

where $\mathbb{C}\text{orr}(\cdot,\cdot)$ is the correlation function of $Z_n^e(\cdot)$. $RF(\mathbf{x})$ is a measure of the (canonical) distance between $\mathbf{x}$ and the design points. It is always between zero and one at the data points because $\mathbb{C}\text{orr}\left(Z_n^e(\mathbf{x}_i), Z_n^e(\mathbf{x}_i)\right) = 1$ for all $\mathbf{x}_i \in \mathbf{X}_n$. Multiplying EI by the RF improves its exploration property as is shown in Figure 4. The picture on the left illustrates a GP fitted on five data points, and the corresponding RF (red), EI (blue), and PEI (black) are visualized on the right picture. It can be seen that the maximum of EI is biased towards the design point whose ES-LOO error is maximum.

The correlation function in (3.13) depends on length scales $\boldsymbol{\theta}^e = [\theta_1^e, \ldots, \theta_d^e]^\top$ that are estimated from $\{\mathbf{X}_n, \mathbf{y}_n^e\}$. It is important that θ_i^e's do not take very "small" values to circumvent clustering. The reason is that when they come near zero, $\mathbb{C}\text{orr}\left(Z_n^e(\mathbf{x}), Z_n^e(\mathbf{x}_i)\right)$ tends to zero and $RF(\mathbf{x})$ is (almost) one everywhere meaning that it has no influence on EI. Besides, the maximum of EI is located in a shrinking neighborhood of the current best point when $\boldsymbol{\theta}^e$ is

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

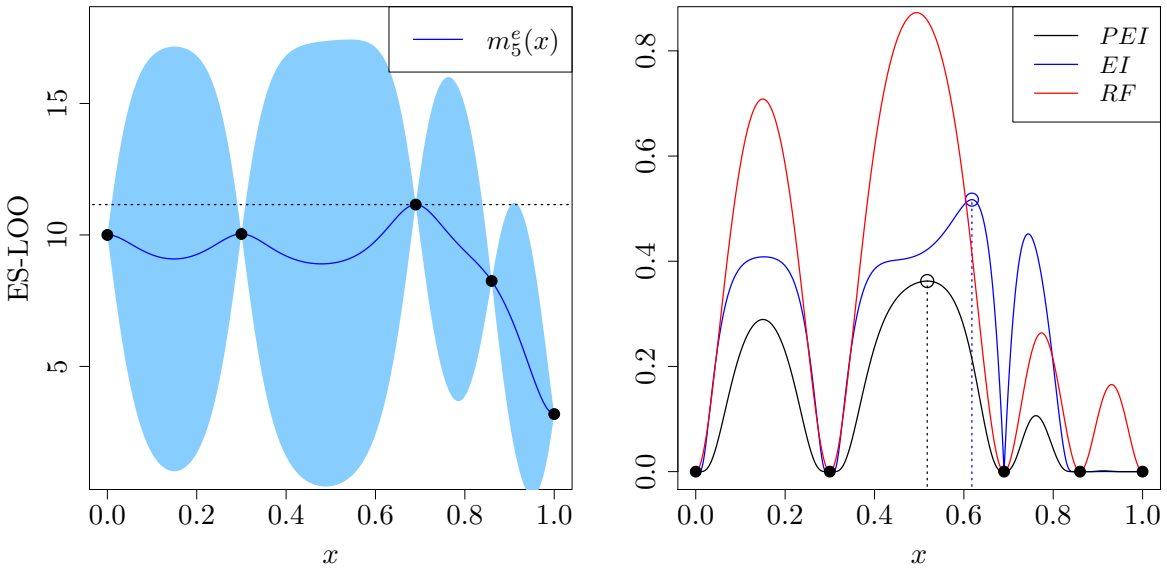

Figure 4. *Left: A GP is fitted to five ES-LOO errors. Right: PEI (black) is obtained by multiplying EI (blue) by the influence functions (red). The next sample point is where the PEI is maximum (black circle). The maximum of EI is shown by the blue circle. The PEI criterion is more explorative than EI.*

small [41]. Therefore, a lower bound has to be considered for $\boldsymbol{\theta}^e$. In a normalized input space, i.e., $\mathcal{D} = [0,1]^d$, we define this lower bound to be

$$\theta_{lb}^e = \sqrt{-0.5/\ln(10^{-8})} \tag{3.14}$$

for each dimension. It is obtained by setting the minimum correlation equal to 10^{-8} for the squared exponential correlation function defined as

$$k_0(x, x') = \exp\left(-\frac{|x - x'|^2}{2\theta^2}\right). \tag{3.15}$$

In the above equation, the minimum correlation between x and x' happens when $|x - x'| = 1$ which is the maximum distance between the two points in the normalized input space.

A common issue in adaptive sampling strategies is that many design points lie along the boundaries of the input space where the predictive variance is large. Such samples might be nonoptimal if the true model is not feasible on the boundaries [18]. However, this problem can be alleviated in our approach by introducing "pseudopoints" at appropriate locations on the boundaries. Pseudopoints, denoted by $\mathbf{X}_p$, are used to update the RF, i.e., $RF(\mathbf{x}; \mathbf{X}_n \cup \mathbf{X}_p)$, but the true function is not evaluated there due to computational cost. The following locations are considered as the pseudopoints in our algorithm:

1. corners of the input space,
2. closest point on each face of the (rectangular) bounding input region to the initial design.

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

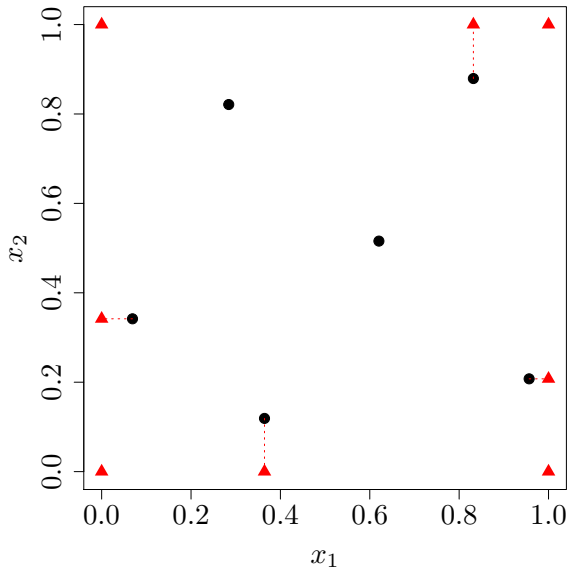

Figure 5. *Initial design (black points) and pseudopoints (red triangles). They are the closest points on each face of the input space to the initial design and also at the corners of the region. The RF is updated by pseudopoints at almost no cost.*

A 2-dimensional example is illustrated in Figure 5 that shows the location of pseudopoints (red triangles) and six initial design (black points). Finally, Algorithm 3.1 summarizes the steps of the proposed adaptive sampling method.

3.3. Extension to batch mode. When parallel computing is available, it is often better to evaluate the expensive function f at a set of inputs rather than a single point since it saves the user time. In batch sampling, $q > 1$ locations are chosen for evaluation at each iteration. Note that the computation time of running the simulator on q parallel cores is the same as a single run. The PEI criterion can be employed in a batch mode thanks to the RF. We now show how to choose q points $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q}$ in a single iteration. The first point $\mathbf{x}_{n+1}$ is obtained by maximizing the PEI criterion. Then, the RF is updated by $\mathbf{x}_{n+1}$:

$$RF(\mathbf{x}; \mathbf{X}_n \cup \mathbf{x}_{n+1}) = \prod_{i=1}^{n+1} \left[1 - \mathbb{C}\mathrm{orr}\left(Z_n^e(\mathbf{x}), Z_n^e(\mathbf{x}_i)\right)\right], \tag{3.16}$$

which updates PEI without evaluating f at $\mathbf{x}_{n+1}$. The second location $\mathbf{x}_{n+2}$ is selected where the updated PEI is maximum. We repeat this procedure until the last point $\mathbf{x}_{n+q}$ is chosen:

$$RF\left(\mathbf{x}; \mathbf{X}_n \cup \mathbf{x}_{n+1} \cup \cdots \cup \mathbf{x}_{n+q-1}\right) = \prod_{i=1}^{n+q-1} \left[1 - \mathbb{C}\mathrm{orr}\left(Z_n^e(\mathbf{x}), Z_n^e(\mathbf{x}_i)\right)\right],$$

$$\mathbf{x}_{n+q} = \arg\max_{\mathbf{x} \in \mathcal{D}} EI(\mathbf{x})\, RF\left(\mathbf{x}; \mathbf{X}_n \cup \mathbf{x}_{n+1} \cup \cdots \cup \mathbf{x}_{n+q-1}\right).$$

Figure 6 shows our adaptive sampling method in batch mode where $q = 3$ locations (x_6, x_7 and x_8) are picked in one iteration. The first new sample x_6 is chosen where the PEI criterion

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Algorithm 3.1 Proposed sequential sampling approach.

Create an initial design: $\mathbf{X}_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
Evaluate f at $\mathbf{X}_n$: $\mathbf{y}_n = f(\mathbf{X}_n)$
Fit the GP $Z_n(\mathbf{x})$ to $\{\mathbf{X}_n, \mathbf{y}_n\}$
while not stop **do**
  for $i = 1$ to n **do**
    Calculate $\mathcal{E}_L(\mathbf{x}_i)$ (equation (3.2))
  end for
  Set $\mathbf{y}_n^e = (\mathcal{E}_L(\mathbf{x}_1), \ldots, \mathcal{E}_L(\mathbf{x}_n))^\top$
  Set the lower bound for $\boldsymbol{\theta}^e$ (equation (3.14))
  Fit the GP $Z_n^e(\mathbf{x})$ to $\{\mathbf{X}_n, \mathbf{y}_n^e\}$
  Create pseudopoints $\mathbf{X}_p$ (see Figure 5)
  $\mathbf{x}_{n+1} \leftarrow \operatorname{argmax}_{\mathbf{x}\in\mathcal{D}} PEI(\mathbf{x}) = EI(\mathbf{x})RF(\mathbf{x}; \mathbf{X}_n \cup \mathbf{X}_p)$
  Set $\mathbf{X}_n = \mathbf{X}_n \cup \{\mathbf{x}_{n+1}\}$
  Evaluate f at $\mathbf{x}_{n+1}$ and $y_{n+1} \leftarrow f(\mathbf{x}_{n+1})$
  Set $\mathbf{y}_n = \mathbf{y}_n \cup \{y_{n+1}\}$
  Update $Z_n(\mathbf{x})$ using $(\mathbf{x}_{n+1}, y_{n+1})$
  $n \leftarrow n + 1$
end while

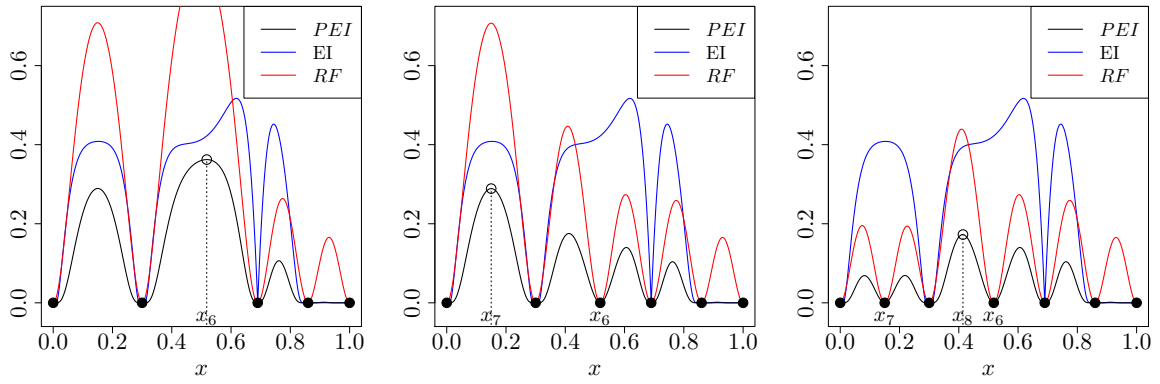

Figure 6. *A batch of $q = 3$ points (x_6, x_7, and x_8) are selected thanks to the RF. Left: there are five initial samples, and the first query point, x_6, is chosen where PEI is maximum. Middle: the RF is updated as $RF(x; x_{1:5} \cup x_6)$ which updates PEI accordingly without evaluating f at x_6. The second query point, x_7, is the location of maximum of updated PEI. Right: PEI is updated using x_7, and the third sample, x_8, is selected as explained.*

is maximum (left). Then, the RF is updated by x_6. This will update PEI, and its maximum allows us to find x_7 without evaluating f at x_6 (middle). Again, we update the RF using x_7 and maximize the updated PEI to obtain the sample site x_8 (right).

4. Numerical experiments.

Experimental results are presented and discussed in this section. Four analytic test functions and two real-world problems are considered as the "true" function to assess the efficiency of our proposed sampling method. The results are compared

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

with one-shot Latin hypercube sampling (LHS) and three adaptive approaches, namely, MSE, EIGF, and MICE. A typical characteristic of these adaptive sampling approaches is exhibited in Figure 7 where the black dots are the initial design (identical in all pictures) and the red circles represent the adaptive samples. The true function has two spikes as it is a sum of two

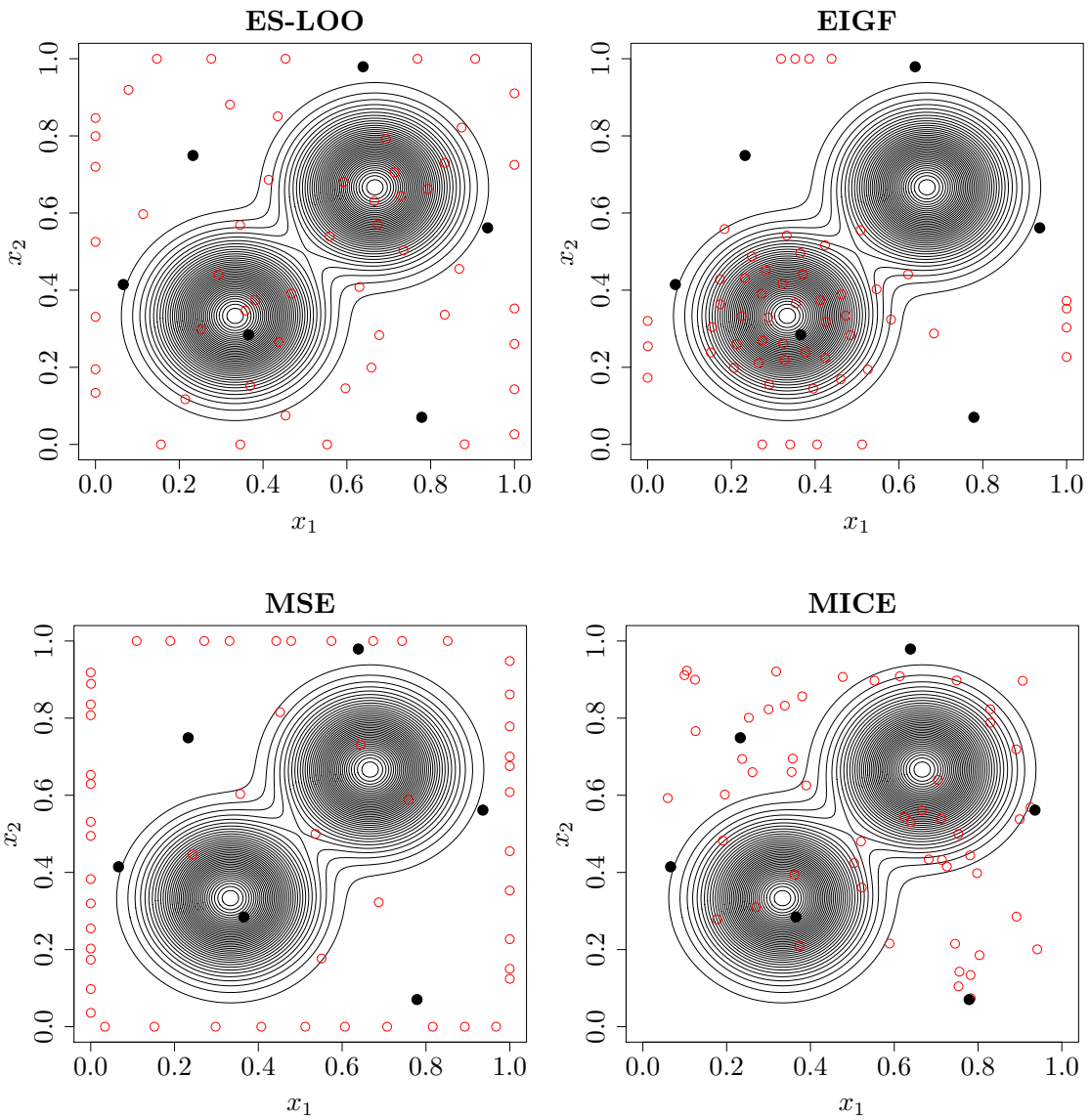

Figure 7. *A typical sampling behavior of four adaptive sampling methods: ES-LOO, EIGF, MSE, and MICE. The red circles represent adaptive samples that are added to the six initial design (black dots). ES-LOO tends to fill the space with focusing on regions where the function values change rapidly. EIGF mainly exploits the basin of attraction of an optimum while other areas are unexplored. MSE samples more points on the boundaries where the prediction uncertainty is large. There is not any special pattern in the sampling behavior of MICE; however, it avoids putting points on the boundaries.*

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Gaussian functions centered at $(1/3, 1/3)^\top$ and $(2/3, 2/3)^\top$. As can be seen, our proposed method fills space uniformly with a focus on areas where the true function has nonlinear behavior. The efficiency of our algorithm is due to two parts together. Firstly, the ES-LOO measure reflects the actual model error and indicates promising regions for future evaluations. Secondly, incorporating the RF in our EI-based selection criterion prevents points from clustering. Without the repulsion trick, as is the case in Figure 3, new experimental designs tend to pile up at points where the ES-LOO errors are large. However, further investigation is required to identify the exact contribution of each part on the algorithm. In our method the boundary issue, i.e., taking many points on the boundaries, is mitigated by introducing pseudopoints at locations shown in Figure 5.

EIGF tends toward local exploitation and does not explore the input space; it gets stuck in an optimum. This can be explained according to the EIGF formula in (3.11) where local exploitation is carried out by the first term, $(m_n(\mathbf{x}) - f(\mathbf{x}_i^*))^2$. It represents the difference between the GP prediction and response at the nearest sample location $\mathbf{x}_i^*$ and increases in regions where a drastic response change takes place. As a result, the first term can be interpreted as a measure of gradient. In Figure 7, there is a design point close to the center of the spike $(1/3, 1/3)^\top$ at which the function reaches its maximum. Since the function varies "significantly" there, the EIGF criterion is large (due to its first term) around $(1/3, 1/3)^\top$ where most samples are taken. Note that the points sampled on the boundaries are due to increase in the second term of the EIGF expression, i.e., $s_n^2(\mathbf{x})$. MSE samples most points on the boundaries where the prediction uncertainty is large. It is possible that MSE leads to a space-filling design as is the case of the Franke's function; see below. No special trend can be found on the performance of the MICE algorithm. However, it avoids sampling around the boundaries, and all new points are taken in the interior region. We describe our implementation of ES-LOO below.

4.1. Experimental setup. The prediction accuracy is assessed by the *root mean squared error (RMSE)* criterion. Given the test set $\{(\mathbf{x}_t, f(\mathbf{x}_t))\}_{t=1}^{t=N}$, RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N} \left(m_n(\mathbf{x}_t) - f(\mathbf{x}_t)\right)^2}{N}}, \tag{4.1}$$

which measures the distance between the emulator, $m_n(\mathbf{x}_t)$, and the true function, $f(\mathbf{x}_t)$. In our experiments, $N = 3000$, and the test points are selected uniformly across the input space. A total budget equal to $30d$ is considered for each experiment. The initial space-filling DoE is of size $3d$ and is obtained by the `maximinESE_LHS` function implemented in the R package *DiceDesign* [13]. There are ten different initial DoEs for every function, and we assess the prediction performance of each method using all ten sets. The R package *DiceKriging* [49] is employed to construct GP models. The covariance kernel for modeling the true function and $\mathcal{E}_L(\mathbf{x})$ is Matérn with $\nu = 3/2$. Since the ES-LOO error is a positive quantity, the GP emulator is fitted to its natural logarithm. The closest point on each face of the input space to the initial design and the corners of the input region are considered as the pseudopoints.

The optimization of the PEI function (and other selection criteria) is conducted by the differential evolution (DE) algorithm [57] implemented in the R package *DEoptim* [42]. This

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

algorithm is considered by [64] to optimize acquisition functions used in the Bayesian optimization paradigm. DE is a stochastic global search algorithm for continuous problems and belongs to the family of population-based evolutionary methods. It can tackle nondifferentiable, multimodal functions and has a small number of parameters to tune. The progress toward better solutions in the search space is made by applying mutation, crossover, and selection operators to the population of candidate solutions. In this paper, the population size is equal to $10d$. The other control parameters (such as the step size and crossover probability) are set to their default values; see the *DEoptim* package.

4.2. Test functions. The four test functions are

- $f_1(\mathbf{x})$: Franke's function [20], $d = 2$;
- $f_2(\mathbf{x})$: Hartmann function [21], $d = 3$;
- $f_3(\mathbf{x})$: Friedman function [14], $d = 5$;
- $f_4(\mathbf{x})$: Gramacy and Lee function [18], $d = 6$,

and their analytic expression are given in Appendix B. The four test functions are defined on $[0, 1]^d$. Figure 8 illustrates a comparison between the prediction performance of our proposed method ES-LOO: sequential (black) and batch with $q = 4$ (orange) and three other sequential sampling approaches: MSE (blue), EIGF (red) and MICE (green). Each curve represents the median of ten RMSEs using ten different initial DoEs. The results are also compared with the one-shot space-filling design (magenta) for all sample sizes $3d$, $4d, \ldots, 30d$; the graph represents the median of ten RMSEs obtained by LHS. The x-axis shows the number of function evaluations divided by the problem dimension, d. The y-axis is on logarithmic scale.

Generally, the prediction performance of our method is comparable to other adaptive approaches. In particular, it outperforms MSE and EIGF in approximating the Hartmann (f_2) and Gramacy and Lee (f_4) functions. As can be seen, the sequential (black) and batch (orange) ES-LOO have similar performances such that in both algorithms the RMSE is monotonically (almost linearly) decreasing in all test problems. However, this is not the case of other adaptive sampling approaches. The Hartmann function has four local minima, and EIGF can get stuck in one of them. The EIGF method has the best performance on the Friedman function (f_3) and has the lowest accuracy in approximating other functions, especially f_4. The reason is that the main response change of f_4 occurs near the right bound of the input space and EIGF puts more points there. The MICE algorithm shows a poor performance in predicting the test functions. However, it is the fastest algorithm as the criterion is based on a discrete representation. Other adaptive sampling methods could be implemented on similar discrete representation, and they would then gain the speed-up. It is observed that the points obtained by the MSE method fill the space almost uniformly in the Franke's function and thus has a similar performance to the LHS method. However, MSE favors sampling more points on the boundary of the input space of the (6-dimensional) f_4 function. LHS is the best sampling strategy to predict f_1, f_2, and f_4. However, the RMSE of ES-LOO approaches that of LHS when the number of function evaluations increases, say, after $25 \times d$ evaluations.

According to Figure 8, it seems that LHS should be preferred over adaptive sampling methods as it has a superior performance in most cases. However, this is not always the case as LHS performs poorly on the real-world problems presented in the next section. Besides, the

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

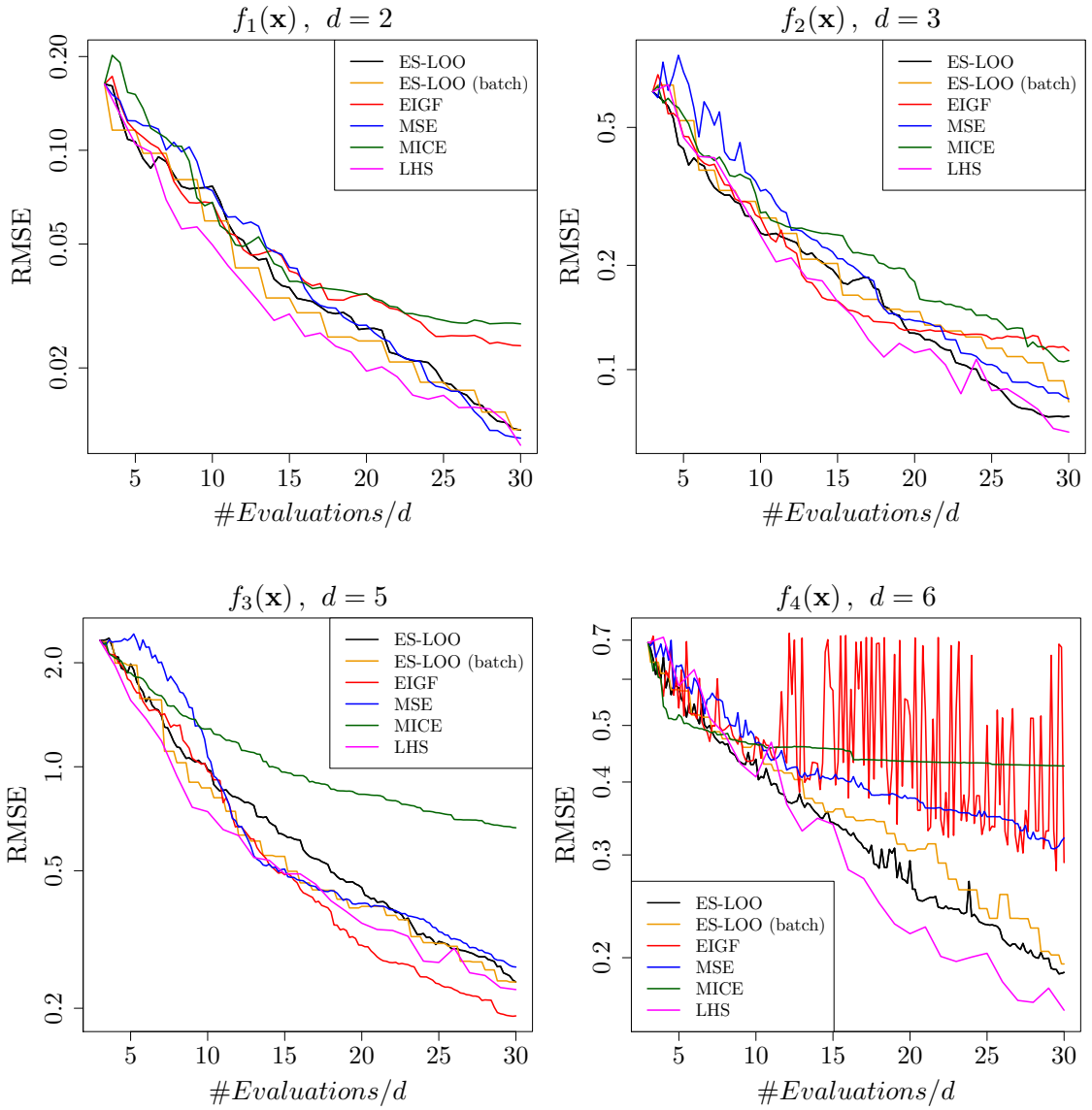

Figure 8. *The median of ten RMSEs of our proposed approach ES-LOO: sequential (black) and batch with $q = 4$ (orange), EIGF (red), MSE (blue), and MICE (green). Ten different initial DoEs of size $3d$ are considered for each function; every method produces ten predictions based on them. The median of RMSEs based on (one-shot) LHS design for sample sizes $3d$, $4d, \ldots, 30d$ is shown in magenta. The total budget is $30d$. The y-axis is on logarithmic scale.*

main disadvantage of LHS (one-shot methods in general) is that we cannot stop them as soon as the emulator reaches an acceptable level of accuracy. In adaptive approaches, however, the costly sampling procedure can be halted at any time. For example, suppose that f_2 is the underlying function of a complex code and the LHS sample size (as a rule of thumb) is $10 \times d = 30$. Let RMSE $= 0.5$ be the acceptable prediction accuracy. In this setting, the

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

ES-LOO algorithm can be stopped at about 15 runs which is half of the time required to execute the model based on the LHS design.

4.3. Real-world problems. We also tested our method on two real-world problems which are the 6-dimensional *output transformerless (OTL) circuit* and 7-dimensional *piston simulation* functions [6]. The former (f_{OTL}) returns the midpoint voltage of a transformerless circuit, and the latter (f_{piston}) measures the cycle time that a piston takes to complete one cycle within a cylinder. The analytical expressions of f_{OTL} and f_{piston} and their design spaces are given in Appendix C. Figure 9 illustrates the results of comparing our proposed method with MSE, EIGF, MICE, and LHS design in predicting the OTL circuit and piston simulation functions. The experimental setup is the same as explained in subsection 4.1. The sequential (black) and batch (orange) ES-LOO again have similar performances, and their RMSE criterion reduces steadily on both problems.

As can be seen, ES-LOO is the best sampling approach for emulating the piston simulation function and performs quite well on the OTL problem. The MSE approach does not work well in both problems, especially after almost $10 \times d$ function evaluations. Based on our experiments, MSE is not recommended when the dimensionality of the problem is larger than five. LHS has a similar behavior to MSE and is not a good sampling approach on the OTL and piston simulation problems, contrary to our previous results demonstrated in Figure 8. While EIGF has the best performance on the OTL circuit problem, there is no improvement in the prediction accuracy of the emulator after $10 \times d$ evaluations of the piston simulation function. MICE is the least successful algorithm in these two problems.

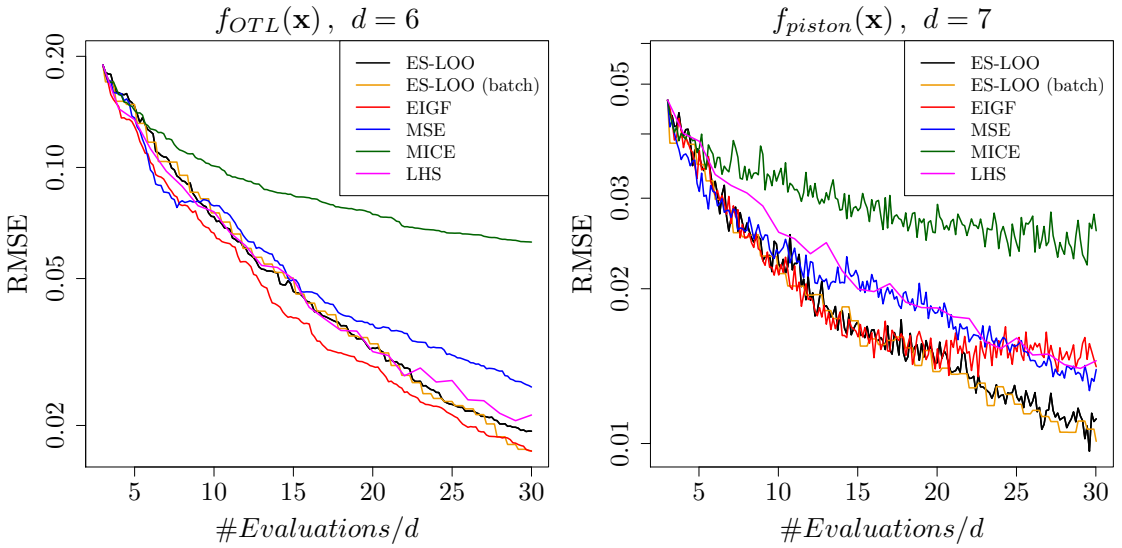

Figure 9. *The median of ten RMSEs of our proposed approach ES-LOO: sequential (black) and batch with $q = 4$ (orange), EIGF (red), MSE (blue), and MICE (green). Size of initial design $= 3d$ and total budget $= 30d$. The magenta line represents the median of RMSEs obtained by (one-shot) LHS design for sample sizes $3d, 4d, \ldots, 30d$. The y-axis is on logarithmic scale.*

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

5. Conclusions. This paper deals with the problem of extending an initial design sequentially for training GP models. This is an important issue in the context of emulating computationally expensive computer codes where the goal is to approximate the underlying function with a minimum number of evaluations. An adaptive sampling scheme is presented based on the ES-LOO error which is used to identify "good" locations for future evaluations. Since the value of ES-LOO is only known at the design points, another GP model is applied to approximate it at unobserved sites. Then, the PEI criterion is employed at each iteration to find the location of maximum of ES-LOO as the most promising point to improve the emulator. PEI is obtained by multiplying the EI criterion by an RF. Once the new sample is chosen, it is added to the existing designs, and the procedure is repeated until a stopping criterion is met. The proposed method can be easily promoted to a batch mode where at each iteration a set of input points is selected for evaluation. This can save the user time if parallel computing is available. Several test functions are used to test the capability of our method, and the results are compared with other commonly used sampling techniques. The results show that our proposed adaptive sampling approach is promising. Ideally there would be some proof of the asymptotic convergence of the ES-LOO to zero error. Unfortunately, we have not been able to derive such proofs, and they remain for further work. We think that following similar results from the convergence of EI in Bayesian optimization (see, e.g., [58, 11, 50]) can help to prove the asymptotic properties of our method.

One can extend the idea of the RF to other sequential sampling approaches. For example, Figure 10 illustrates the behavior of EIGF and MSE modified by the repulsion trick. As can be seen, such modification promotes the diversity of samples in EIGF and improves the RMSE criterion. However, in the modified MSE algorithm the boundary issue is intensified as the RF tends to push away the points. A possible future research direction is to investigate the inclusion of such repulsion-type functions in sampling strategies to improve their exploration property.

Appendix A. MICE algorithm. In the MICE algorithm, the continuous design space is discretized into a finite grid $\mathbf{X}_G \subseteq \mathcal{D}$ such that $\mathbf{X}_G = \mathbf{X}_n \cup \mathbf{X}_{cand}$. The latter is a set of candidate points on which the optimization of the MI criterion is performed. The elements of $\mathbf{X}_{cand}$ are regenerated at each iteration based on a maximin design scheme. The next sample location is obtained by the following optimization problem:

$$\mathbf{x}_{n+1} = \underset{\mathbf{x}\in\mathbf{X}_{cand}}{\operatorname{argmax}}\ s_n^2(\mathbf{x})/s^2_{G\setminus(n\cup\mathbf{x})}(\mathbf{x};\tau^2), \tag{A.1}$$

wherein $G\setminus(n\cup\mathbf{x})$ denotes $\mathbf{X}_G\setminus(\mathbf{X}_n\cup\mathbf{x})$ and τ^2 is the nugget effect added to the correlation matrix of the GP fitted to $\mathbf{X}_G\setminus(\mathbf{X}_n\cup\mathbf{x})$. The inclusion of the nugget prevents the denominator of (A.1) approaching zero. The recommended value for the nugget parameter is one, although it can take any positive value in theory.

Appendix B. Test function expressions. The analytic expressions of four test functions used in our experiments are given below.

1. $f_1(\mathbf{x}) = 0.75\exp(-\frac{(9x_1-2)^2}{4} - \frac{(9x_2-2)^2}{4}) + 0.75\exp(-\frac{(9x_1+1)^2}{49} - \frac{9x_2+1)}{10})$
$+ 0.5\exp(-\frac{(9x_1-7)^2}{4} - \frac{(9x_2-3)^2}{4}) - 0.2\exp(-(9x_1-4)^2 - (9x_2-7)^2).$

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

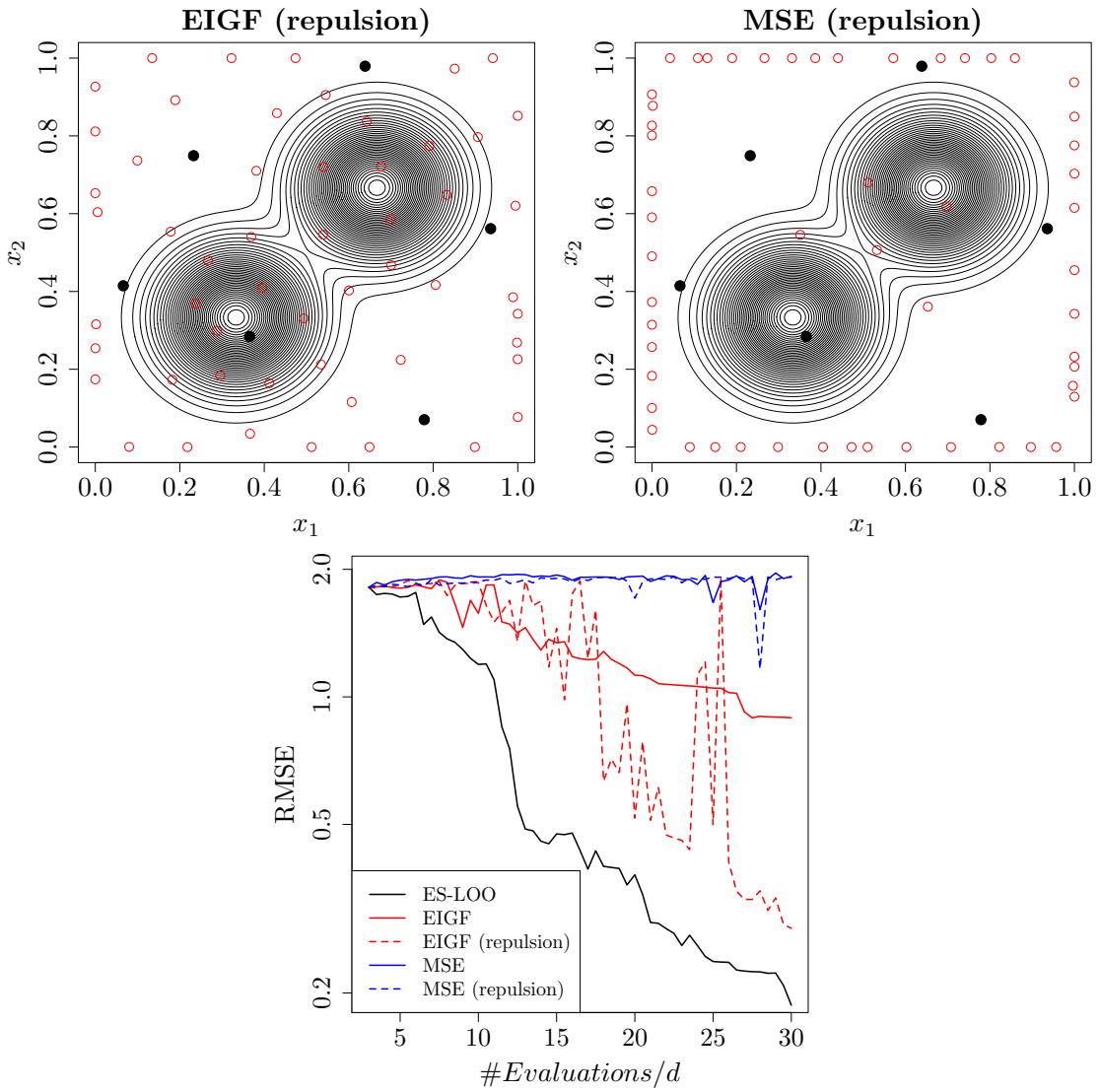

Figure 10. *Top: Incorporating the RF in the EIGF (left) and MSE (right) algorithms. While such modification promotes the diversity of points in EIGF, it intensifies the boundary issue in MSE. Bottom: The median of ten RMSEs associated with ES-LOO (black), EIGF (red), MSE (blue), modified EIGF (dashed red), and MSE (dashed blue) with the RF.*

2. $f_2(\mathbf{x}) = -\sum_{i=1}^{4} \boldsymbol{\alpha}_i \exp(\sum_{j=1}^{3} \mathbf{A}_{ij}(x_j - \mathbf{P}_{ij})^2)$, where $\boldsymbol{\alpha} = (1, 1.2, 3, 3.2)^\top$,

$$\mathbf{A} = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}, \quad \mathbf{P} = 10^{-4} \begin{bmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{bmatrix}.$$

3. $f_3(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$.

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

4. $f_4(\mathbf{x}) = \exp\left(\sin\left([0.9(x_1 + 0.48))]^{10}\right)\right) + x_2 x_3 + x_4$.

Appendix C. Real-world function expressions.

OTL circuit function. The function f_{OTL} is defined as

$$f_{OTL}(\mathbf{x}) = \frac{(V_{b1} + 0.74)\beta(R_{c2} + 9)}{\beta(R_{c2} + 9) + R_f} + \frac{11.35 R_f}{\beta(R_{c2} + 9) + R_f} + \frac{0.74 R_f \beta(R_{c2} + 9)}{(\beta(R_{c2} + 9) + R_f) R_{c1}},$$

where $V_{b1} = \frac{12 R_{b2}}{R_{b1} + R_{b2}}$. The input variables of f_{OTL} are the following:

- $R_{b1} \in [50, 150]$ is the resistance b_1 (K-Ohms),
- $R_{b2} \in [25, 70]$ is the resistance b_2 (K-Ohms),
- $R_f \in [0.5, 3]$ is the resistance f (K-Ohms),
- $R_{c1} \in [1.2, 2.5]$ is the resistance c_1 (K-Ohms),
- $R_{c2} \in [0.25, 1.2]$ is the resistance c_2 (K-Ohms),
- $\beta \in [50, 300]$ is the current gain c_1 (amperes).

Piston simulation function. The function f_{piston} is defined as

$$f_{piston}(\mathbf{x}) = 2\pi \sqrt{\frac{M}{k + S^2 \frac{P_0 V_0 T_a}{T_0 V^2}}}, \quad \text{where } V = \frac{S}{2k}\left(\sqrt{A^2 + 4k\frac{P_0 V_0}{T_0} T_a} - A\right),$$

$$A = P_0 S + 19.62 M - \frac{k V_0}{S}.$$

The input variables of f_{piston} are the following:

- $M \in [30, 60]$ is the piston weight (kg),
- $S \in [0.005, 0.020]$ is the piston surface area (m^2),
- $V_0 \in [0.002, 0.010]$ is the initial gas volume (m^3),
- $k \in [1000, 5000]$ is the spring coefficient (N/m),
- $P_0 \in [90000, 110000]$ is the atmospheric pressure (N/m^2),
- $T_a \in [290, 296]$ is the ambient temperature (K),
- $T_0 \in [340, 360]$ is the filling gas temperature (K).

Acknowledgments. We would like to thank the two anonymous reviewers for their suggestions and comments. The authors would like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the Uncertainty Quantification programme (UNQ) where work on this article was undertaken. We also warmly thank Joakim Beck for discussions on the MICE algorithm.

REFERENCES

[1] S. Arlot and A. Celisse, *A survey of cross-validation procedures for model selection*, Stat. Surv., 4 (2010), pp. 40–79, https://doi.org/10.1214/09-SS054.

[2] V. Aute, K. Saleh, O. Abdelaziz, S. Azarm, and R. Radermacher, *Cross-validation based single response adaptive design of experiments for Kriging metamodeling of deterministic computer simulations*, Struct. Multidiscip. Optim., 48 (2013), pp. 581–605, https://doi.org/10.1007/s00158-013-0918-5.

[3] F. Bachoc, *Cross validation and maximum likelihood estimation of hyper-parameters of Gaussian processes with model misspecification*, Comput. Statis. Data Anal., 66 (2013), pp. 55–69.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

[4] G. A. Banyay, M. D. Shields, and J. C. Brigham, *Efficient global sensitivity analysis for flow-induced vibration of a nuclear reactor assembly using Kriging surrogates*, Nuclear Engrg. Des., 341 (2019), pp. 1–15, https://doi.org/10.1016/j.nucengdes.2018.10.013.

[5] J. Beck and S. Guillas, *Sequential design with mutual information for computer experiments (MICE): Emulation of a tsunami model*, SIAM/ASA J. Uncertain. Quantif., 4 (2016), pp. 739–766, https://doi.org/10.1137/140989613.

[6] E. N. Ben-Ari and D. M. Steinberg, *Modeling data from computer experiments: An empirical comparison of Kriging with Mars and projection pursuit regression*, Qual. Engrg, 19 (2007), pp. 327–338, https://doi.org/10.1080/08982110701580930.

[7] M. Ben Salem, O. Roustant, F. Gamboa, and L. Tomaso, *Universal prediction distribution for surrogate models*, SIAM/ASA J. Uncertain. Quantif., 5 (2017), pp. 1086–1109, https://doi.org/10.1137/15M1053529.

[8] G. E. P. Box and J. S. Hunter, *The* 2^{k-p} *fractional factorial designs part* II, Technometrics, 3 (1961), pp. 449–458.

[9] P.-C. Bürkner, J. Gabry, and A. Vehtari, *Approximate leave-future-out cross-validation for Bayesian time series models*, J. Stat. Comput. Simul., 90 (2020), pp. 2499–2523, https://doi.org/10.1080/00949655.2020.1783262.

[10] E. Brochu, V. M. Cora, and N. de Freitas, *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*, preprint, arXiv:1012.2599 [cs.LG], 2010.

[11] A. D. Bull, *Convergence rates of efficient global optimization algorithms*, J. Mach. Learn. Res., 12 (2011), p. 2879–2904.

[12] O. Dubrule, *Cross validation of Kriging in a unique neighborhood*, J. Int. Assoc. Math. Geol., 15 (1983), pp. 687–699, https://doi.org/10.1007/BF01033232.

[13] D. Dupuy, C. Helbert, and J. Franco, *DiceDesign and DiceEval: Two R packages for design and analysis of computer experiments*, J. Statist. Software, 65 (2015), pp. 1–38, https://doi.org/10.18637/jss.v065.i11.

[14] J. H. Friedman, *Multivariate adaptive regression splines*, Ann. Statist., 19 (1991), pp. 1–67.

[15] S. S. Garud, I. A. Karimi, and M. Kraft, *Design of computer experiments: A review*, Comput. Chem. Engrg., 106 (2017), pp. 71–95, https://doi.org/10.1016/j.compchemeng.2017.05.010.

[16] A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin, *Bayesian Data Analysis*, 3rd ed., Chapman & Hall/CRC Texts Stat. Sci. Ser., CRC Press, Boca Raton, FL, 2013.

[17] D. Ginsbourger and C. Schärer, *Fast Calculation of Gaussian Process Multiple-Fold Cross-Validation Residuals and Their Covariances*, preprint, arXiv:2101.03108 [stat.ME], 2021, https://arxiv.org/abs/2101.03108.

[18] R. B. Gramacy and H. K. H. Lee, *Adaptive design and analysis of supercomputer experiments*, Technometrics, 51 (2009), pp. 130–145, https://doi.org/10.1198/TECH.2009.0015.

[19] L. L. Gratiet and C. Cannamela, *Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes*, Technometrics, 57 (2015), pp. 418–427, https://doi.org/10.1080/00401706.2014.928233.

[20] B. Haaland and P. Z. G. Qian, *Accurate emulators for large-scale computer experiments*, Ann. Statist., 39 (2011), pp. 2974–3002, https://doi.org/10.1214/11-AOS929.

[21] M. Jamil and X.-S. Yang, *A literature survey of benchmark functions for global optimization problems*, Int. J. Math. Model. Numer. Optim., 4 (2013), pp. 150–194, https://doi.org/10.1504/IJMMNO.2013.055204.

[22] R. Jin, W. Chen, and A. Sudjianto, *On sequential sampling for global metamodeling in engineering design*, in Conference Proceedings: Design Engineering Technical Conference and Computers and Information in Engineering, vol. 2, ASME, 2002, pp. 539–548.

[23] M. Johnson, L. Moore, and D. Ylvisaker, *Minimax and maximin distance designs*, J. Statist. Plann. Inference, 26 (1990), pp. 131–148, https://doi.org/10.1016/0378-3758(90)90122-B.

[24] D. R. Jones, *A taxonomy of global optimization methods based on response surfaces*, J. Global Optim., 21 (2001), pp. 345–383, https://doi.org/10.1023/A:1012771025575.

[25] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient global optimization of expensive black-box functions*, J. Global Optim., 13 (1998), pp. 455–492, https://doi.org/10.1023/A:1008306431147.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

[26] V. R. Joseph, *Space-filling designs for computer experiments: A review*, Qual. Engrg., 28 (2016), pp. 28–35, https://doi.org/10.1080/08982112.2015.1100447.

[27] C. Karel, D. Gorissen, D. Deschrijver, and T. Dhaene, *A novel hybrid sequential design strategy for global surrogate modeling of computer experiments*, SIAM J. Sci. Comput., 33 (2011), pp. 1948–1974, https://doi.org/10.1137/090761811.

[28] J. Koehler and A. Owen, *Computer experiments*, in Design and Analysis of Experiments, Handbook of Statist. 13, Elsevier, Amsterdam, 1996, pp. 261–308, https://doi.org/10.1016/S0169-7161(96)13011-X.

[29] A. Krause, A. Singh, and C. Guestrin, *Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies*, J. Mach. Learn. Res., 9 (2008), pp. 235–284.

[30] C. Q. Lam, *Sequential Adaptive Designs in Computer Experiments for Response Surface Model Fit*, Ph.D. thesis, Ohio State University, Columbus, OH, 2008.

[31] L. Le Gratiet and C. Cannamela, *Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes*, Technometrics, 57 (2015), pp. 418–427.

[32] G. Li, V. Aute, and S. Azarm, *An accumulative error based adaptive design of experiments for offline metamodeling*, Struct. Multidiscip. Optim., 40 (2009), pp. 137–157, https://doi.org/10.1007/s00158-009-0395-z.

[33] H. Liang, M. Zhu, and Z. Wu, *Using cross-validation to design trend function in Kriging surrogate modeling*, AIAA J., 52 (2014), pp. 2313–2327, https://doi.org/10.2514/1.J052879.

[34] D. Liu, A. Litvinenko, C. Schillings, and V. Schulz, *Quantification of airfoil geometry-induced aerodynamic uncertainties—comparison of approaches*, SIAM/ASA J. Uncertain. Quantif., 5 (2017), pp. 334–352, https://doi.org/10.1137/15M1050239.

[35] H. Liu, Y.-S. Ong, and J. Cai, *A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design*, Struct. Multidiscip. Optim., 57 (2018), pp. 393–416, https://doi.org/10.1007/s00158-017-1739-8.

[36] H. Liu, S. Xu, Y. Ma, X. Chen, and X. Wang, *An adaptive Bayesian sequential sampling approach for global metamodeling*, J. Mech. Des., 138 (2015), https://doi.org/10.1115/1.4031905.

[37] H. Maatouk, O. Roustant, and Y. Richet, *Cross-validation estimations of hyper-parameters of Gaussian processes with inequality constraints*, Procedia Environ. Sci., 27 (2015), pp. 38–44, https://doi.org/10.1016/j.proenv.2015.07.105.

[38] J. Martin and T. Simpson, *Use of adaptive metamodeling for design optimization*, in Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2002, pp. 1–9, https://doi.org/10.2514/6.2002-5631.

[39] L. Martino, V. Laparra, and G. Camps-Valls, *Probabilistic cross-validation estimators for Gaussian process regression*, in Proceedings of the 25th European Signal Processing Conference (EUSIPCO), 2017, pp. 823–827, https://doi.org/10.23919/EUSIPCO.2017.8081322.

[40] M. D. McKay, R. J. Beckman, and W. J. Conover, *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code*, Technometrics, 21 (1979), pp. 239–245.

[41] H. Mohammadi, *Kriging-Based Black-Box Global Optimization: Analysis and New Algorithms*, Ph.D. thesis, École Nationale Supérieure des Mines de Saint-Étienne, France, 2016.

[42] K. M. Mullen, D. Ardia, D. L. Gil, D. Windover, and J. Cline, *DEoptim: An R package for global optimization by differential evolution*, J. Statist. Software, 40 (2011), pp. 1–26, https://doi.org/10.18637/jss.v040.i06.

[43] R. M. Neal, *Regression and classification using Gaussian process priors*, in Bayesian Statistics 6, Oxford University Press, Oxford, 1998, pp. 475–501.

[44] A. B. Owen, *Orthogonal arrays for computer experiments, integration and visualization*, Statist. Sinica, 2 (1992), pp. 439–452.

[45] V. Picheny, D. Ginsbourger, O. Roustant, R. T. Haftka, and N. H. Kim, *Adaptive designs of experiments for accurate approximation of a target region*, J. Mech. Des., 132 (2010), pp. 1–9.

[46] W. Ponweiser, T. Wagner, and M. Vincze, *Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models*, in Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pp. 3515–3522, https://doi.org/10.1109/CEC.2008.4631273.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

[47] L. Pronzato and W. G. Müller, *Design of computer experiments: Space filling and beyond*, Statist. Comput., 22 (2012), pp. 681–701, https://doi.org/10.1007/s11222-011-9242-3.
[48] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, MIT Press, Cambridge, MA, 2005.
[49] O. Roustant, D. Ginsbourger, and Y. Deville, *DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by Kriging-based metamodeling and optimization*, J. Statist. Software, 51 (2012), pp. 1–55, https://doi.org/10.18637/jss.v051.i01.
[50] I. Ryzhov, *On the convergence rates of expected improvement methods*, Oper. Res., 64 (2016), https://doi.org/10.1287/opre.2016.1494.
[51] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and analysis of computer experiments*, Statist. Sci., 4 (1989), pp. 409–423, https://doi.org/10.1214/ss/1177012420.
[52] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*, Springer-Verlag, Cham, 2003.
[53] M. Schonlau, *Computer Experiments and Global Optimization*, Ph.D. thesis, University of Waterloo, 1997.
[54] R. Sheikholeslami and S. Razavi, *Progressive Latin hypercube sampling: An efficient approach for robust sampling-based analysis of environmental models*, Environ. Model. Software, 93 (2017), pp. 109–126, https://doi.org/10.1016/j.envsoft.2017.03.010.
[55] M. C. Shewry and H. P. Wynn, *Maximum entropy sampling*, J. Appl. Stat., 14 (1987), pp. 165–170, https://doi.org/10.1080/02664768700000020.
[56] T. Simpson, J. Poplinski, P. N. Koch, and J. Allen, *Metamodels for computer-based engineering design: Survey and recommendations*, Engrg. Comput., 17 (2001), pp. 129–150, https://doi.org/10.1007/PL00007198.
[57] R. Storn and K. Price, *Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces*, J. Global Optim., 11 (1997), pp. 341–359, https://doi.org/10.1023/A:1008202821328.
[58] E. Vazquez and J. Bect, *Convergence properties of the expected improvement algorithm with fixed mean and covariance functions*, J. Statist. Plann. Inference, 140 (2010), pp. 3088–3095, https://doi.org/10.1016/j.jspi.2010.04.018.
[59] I. Vernon, J. Liu, M. Goldstein, J. Rowe, J. Topping, and K. Lindsey, *Bayesian uncertainty analysis for complex systems biology models: Emulation, global parameter searches and evaluation of gene functions*, BMC Syst. Biol., 12 (2018), p. 1, https://doi.org/10.1186/s12918-017-0484-3.
[60] F. A. C. Viana, R. T. Haftka, and V. Steffen, *Multiple surrogates: How cross-validation errors can help us to obtain the best predictor*, Struct. Multidiscip. Optim., 39 (2009), pp. 439–457, https://doi.org/10.1007/s00158-008-0338-0.
[61] V. Volodina and D. Williamson, *Diagnostics-driven nonstationary emulators using kernel mixtures*, SIAM/ASA J. Uncertain. Quantif., 8 (2020), pp. 1–26, https://doi.org/10.1137/19M124438X.
[62] D. Williamson, *Exploratory ensemble designs for environmental models using k-extended Latin hypercubes*, Environmetrics, 26 (2015), pp. 268–283, https://doi.org/10.1002/env.2335.
[63] Y. Yun, H.-C. Kim, S. Y. Shin, J. Lee, A. D. Deshpande, and C. Kim, *Statistical method for prediction of gait kinematics with Gaussian process regression*, J. Biomech., 47 (2014), pp. 186–192, https://doi.org/https://doi.org/10.1016/j.jbiomech.2013.09.032.
[64] D. Zhan, J. Qian, and Y. Cheng, *Pseudo expected improvement criterion for parallel EGO algorithm*, J. Global Optim., 68 (2017), pp. 641–662, https://doi.org/10.1007/s10898-016-0484-7.
[65] Y. Zhang and Y. Yang, *Cross-validation for selecting a model selection procedure*, J. Econometrics, 187 (2015), pp. 95–112, https://doi.org/10.1016/j.jeconom.2015.02.006.

Downloaded 03/25/22 to 144.173.177.198 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.