SIAM/ASA J. Uncertainty Quantification
Vol. 5, pp. 1086–1109

Universal Prediction Distribution for Surrogate Models*

Malek Ben Salem[†], Olivier Roustant[‡], Fabrice Gamboa[§], and Lionel Tomaso[¶]

Abstract. The use of surrogate models instead of computationally expensive simulation codes is very convenient in engineering. Roughly speaking, there are two kinds of surrogate models: the deterministic and the probabilistic ones. These last are generally based on Gaussian assumptions. The main advantage of the probabilistic approach is that it provides a measure of uncertainty associated with the surrogate model in the whole space. This uncertainty is an efficient tool to construct strategies for various problems such as prediction enhancement, optimization, or inversion. In this paper, we propose a universal method to define a measure of uncertainty suitable for any surrogate model either deterministic or probabilistic. It relies on cross-validation submodel predictions. This empirical distribution may be computed in much more general frames than the Gaussian one; thus it is called the universal prediction distribution (*UP distribution*). It allows the definition of many sampling criteria. We give and study adaptive sampling techniques for global refinement and an extension of the so-called efficient global optimization algorithm. We also discuss the use of the *UP distribution* for inversion problems. The performances of these new algorithms are studied both on toy models and on an engineering design problem.

Key words. surrogate modeling, design of experiments, Bayesian optimization

AMS subject classifications. 62L05, 62K05, 90C26

DOI. 10.1137/15M1053529

1. Introduction.

Surrogate modeling techniques are widely used and studied in engineering and research. Their main purpose is to replace an expensive-to-evaluate function s by a simple response surface $\hat{s}$ also called a surrogate model or metamodel. Notice that s can be a computation-intensive simulation code. These surrogate models are based on a given training set of n observations $z_j = (x_j, y_j)$, where $1 \leq j \leq n$ and $y_j = s(x_j)$. The accuracy of the surrogate model relies, inter alia, on the relevance of the training set. The aim of surrogate modeling is generally to estimate some features of the function s using $\hat{s}$. Of course one is looking for the best trade-off between good accuracy of the feature estimation and the number of calls of s. Consequently, the design of experiments, that is the sampling of $(x_j)_{1\leq j\leq n}$, is a crucial step and an active research field.

*Received by the editors December 18, 2015; accepted for publication (in revised form) August 28, 2017; published electronically November 14, 2017.

http://www.siam.org/journals/juq/5/M105352.html

Funding: The first author is funded by a CIFRE grant from the ANSYS company, subsidized by the French National Association for Research and Technology (ANRT, CIFRE grant 2014/1349).

[†] Mines de Saint-Étienne, UMR CNRS 6158, Limos, F-42023, Saint-Étienne, France and ANSYS, Inc, F-69100 Villeurbanne, France (malek.ben-salem@emse.fr).

[‡] Mines de Saint-Étienne, UMR CNRS 6158, Limos, F-42023, Saint-Étienne, France (roustant@emse.fr).

[§] IMT Institut de Mathématiques de Toulouse, F-31062 Toulouse, Cedex 9, France (fabrice.gamboa@math.univ-toulouse.fr).

[¶] ANSYS, Inc, F-69100 Villeurbanne, France (lionel.tomaso@ansys.com).

There are two ways to sample: either drawing the training set $(x_j)_{1\leq j\leq n}$ at once or building it sequentially. Among the sequential techniques, some are based on surrogate models. They rely on the feature of s that one wishes to estimate. Popular examples are efficient global optimization (EGO) [17] and stepwise uncertainty reduction [3]. These two methods use Gaussian process regression also called the kriging model. It is a widely used surrogate modeling technique. Its popularity is mainly due to its statistical nature and properties. Indeed, it is a Bayesian inference technique for functions. In this stochastic frame, it provides an estimate of the prediction error distribution. This distribution is the main tool in Gaussian surrogate sequential designs. For instance, it allows the introduction and the computation of different sampling criteria such as expected improvement (EI) [17] or expected feasibility [4]. Away from the Gaussian case, many surrogate models are also available and useful. Notice that none of them including the Gaussian process surrogate model are the best in all circumstances [14]. Classical surrogate models are, for instance, the support vector machine (SVM) [36], linear regression [5], moving least squares [22]. More recently a mixture of surrogates has been considered in [38, 13]. Nevertheless, these methods are generally not naturally embeddable in some stochastic frame. Hence, they do not provide any prediction error distribution. To overcome this drawback, several empirical design techniques have been discussed in the literature. These techniques are generally based on resampling methods such as bootstrap, jackknife, or cross validation. For instance, Gazut et al. [10] and Jin, Chen, and Sudjianto [15] consider a population of surrogate models constructed by resampling the available data using bootstrap or cross validation. Then, they compute the empirical variance of the predictions of these surrogate models. Finally, they sample iteratively the point that maximizes the empirical variance in order to improve the accuracy of the prediction. To perform optimization, Kleijnen, van Beers, and van Nieuwenhuyse [20] use a bootstrapped kriging variance instead of the kriging variance to compute the EI. Their algorithm consists in maximizing the EI computed through bootstrapped kriging variance. However, most of these resampling method-based design techniques lead to clustered designs [2, 15].

In this paper, we give a general way to build an empirical prediction distribution allowing sequential design strategies in a very broad frame. Its support is the set of all the predictions obtained by the cross-validation surrogate models. The novelty of our approach is that it provides a prediction uncertainty distribution. This allows a large set of sampling criteria.

The paper is organized as follows. We start by presenting in section 2 the background and notations. In section 3 we introduce the universal prediction (UP) empirical distribution. In sections 4 and 5, we use and study features estimation and the corresponding sampling schemes built on the UP empirical distribution. Section 4 is devoted to the enhancement of the overall model accuracy. Section 5 concerns optimization. In section 6, we study a real life industrial case implementing the methodology developed in section 4. Section 7 deals with the inversion problem. In section 8, we conclude and discuss the possible extensions of our work. All proofs are postponed to section 9.

2. Background and notations.

2.1. General notation.

To begin with, let s denote a real-valued function defined on $\mathbb{X}$, a nonempty compact subset of the Euclidean space $\mathbb{R}^p$ ($p \in \mathbb{N}^\star$). In order to estimate s, we have at hand a sample of size n ($n \geq 2$): $\mathbf{X_n} = \left(\mathbf{x_1}, \quad \ldots, \quad \mathbf{x_n}\right)^\top$ with $\mathbf{x_j} \in \mathbb{X}$, $j \in [\![1;n]\!]$, and $\mathbf{Y_n} = \left(y_1, \quad \ldots, \quad y_n\right)^\top$, where $y_j = s(\mathbf{x_j})$ for $j \in [\![1;n]\!]$. We note $\mathbf{Y_n} = s(\mathbf{X_n})$.

Let $\mathbf{Z_n}$ denote the observations: $\mathbf{Z_n} := \{(\mathbf{x_j}, y_j), j \in [\![1;n]\!]\}$. Using $\mathbf{Z_n}$, we build a surrogate model $\hat{s}_n$ that mimics the behavior of s. For example, $\hat{s}_n$ can be a second order polynomial regression model. For $i \in \{1 \ldots n\}$, we set $\mathbf{Z}_{n,-i} := \{(\mathbf{x_j}, y_j), j = 1, \ldots, n, j \neq i\}$ and so $\hat{s}_{n,-i}$ is the surrogate model obtained by using only the dataset $\mathbf{Z}_{n,-i}$. We will call $\hat{s}_n$ the master surrogate model and $(\hat{s}_{n,-i})_{i=1\ldots n}$ its submodels.

Further, let $d(.,.)$ denote a given distance on $\mathbb{R}^p$ (typically the Euclidean one). For $\mathbf{x} \in \mathbb{X}$ and $A \subset \mathbb{X}$, we set $\underline{d}_A(\mathbf{x}) = \inf\{d(\mathbf{x}, \mathbf{x}') : \mathbf{x}' \in A\}$ and if $A = \{\mathbf{x'_1}, \ldots, \mathbf{x'_m}\}$ is finite ($m \in \mathbb{N}^\star$), for $i \in 1, \ldots, m$ let A_{-i} denote $\{\mathbf{x'_j}, j = 1 \ldots m, j \neq i\}$. Finally, we set $\bar{d}(A) = \max\{\underline{d}_{A_{-i}}(\mathbf{x'_i}) : i = 1, \ldots, m\}$, the largest distance of an element of A to its nearest neighbor.

2.2. Cross validation. Training an algorithm and evaluating its statistical performances on the same data yields an optimistic result [1]. It is well known that it is easy to overfit the data by including too many degrees of freedom and so inflate the fit statistics. The idea behind cross validation (CV) is to estimate the risk of an algorithm splitting the dataset once or several times. One part of the data (the training set) is used for training and the remaining one (the validation set) is used for estimating the risk of the algorithm. Simple validation or holdout [8] is hence a CV technique. It relies on one splitting of the data. Then one set is used as training set and the second one is used as validation set. Some other CV techniques consist in a repetitive generation of holdout estimators with different data splitting [11]. One can cite, for instance, the leave-one-out (LOO) CV (LOO-CV) and the k-fold CV (KFCV). KFCV consists in dividing the data into k subsets. Each subset plays the role of validation set while the remaining $k - 1$ subsets are used together as the training set. The LOO-CV method is a particular case of KFCV with $k = n$.

For $i = 1, \ldots, n$, the LOO error is $\varepsilon_i = \hat{s}_{n,-i}(\mathbf{x_i}) - y_i$, where the submodels $\hat{s}_{n,-i}$ are introduced in section 2.1. In our study, we are interested in the distribution of the local predictor for all $\mathbf{x} \in \mathbb{X}$ ($\mathbf{x}$ is not necessarily a design point). As explained in the next section, the CV paradigm provides submodels allowing the definition of a local *uncertainty* measure for the master surrogate model $\hat{s}_n$. This distribution is estimated by using LOO-CV predictions. This is one of the easiest ways to build an uncertainty measure based on resampling.

3. UP distribution.

3.1. Overview. As discussed in the previous section, CV is used as a method for estimating the prediction error of a given model. In our case, we introduce a novel use of CV in order to estimate the local uncertainty of a surrogate model prediction. Hence, for a given surrogate model $\hat{s}$ and for any $\mathbf{x} \in \mathbb{X}$, $\hat{s}_{n,-1}(\mathbf{x}), \ldots, \hat{s}_{n,-n}(\mathbf{x})$ define an empirical distribution of $\hat{s}(\mathbf{x})$ at $\mathbf{x}$. In the case of an interpolating surrogate model and a deterministic simulation code s, it is natural to enforce a zero variance at design points. Consequently, when predicting on a design point $\mathbf{x_i}$ we have to neglect the prediction $\hat{s}_{n,-i}$. This can be achieved by introducing weights on the empirical distribution. These weights avoid the pessimistic submodel predictions that might occur in a region while the global surrogate model fits the data well in that region.

Let $\hat{F}^{(0)}_{n,\mathbf{x}} = \sum_{i=1}^{n} w^0_{i,n}(\mathbf{x})\delta_{\hat{s}_{n,-i}(\mathbf{x})}(dy)$ be the weighted empirical distribution based on the n different predictions of the LOO-CV submodels $\{\hat{s}_{n,-i}(\mathbf{x})\}_{1\leq i\leq n}$ and weighted by $w^0_{i,n}(\mathbf{x})$ defined in (3.1):

$$w^0_{i,n}(\mathbf{x}) = \begin{cases} \dfrac{1}{n-1} & \text{if } \mathbf{x_i} \neq \arg\min\{d(\mathbf{x},\mathbf{x_j}), j=1,\dots,n\}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

For $i = 1,\dots,n$, let R_i be the Voronoi cell of the point $\mathbf{x_i}$. The weights can be written as $w^0_{i,n}(\mathbf{x}) = \frac{1-\mathbf{1}_{R_i}(\mathbf{x})}{\sum_{j=1}^{n}(1-\mathbf{1}_{R_j}(\mathbf{x}))}$, where $\mathbf{1}_{R_i}$ is the indicator function on R_i. Such binary weights lead to unsmooth design criteria. In order to avoid this drawback, we smooth the weights. Direct smoothing based on convolution would lead to the computations of Voronoi cells. We prefer to use a simpler technique. Indeed, $w^0_{i,n}(\mathbf{x})$ can be seen as a Nadaraya–Watson weight with the kernel $1 - \mathbf{1}_{R_j}(\mathbf{x}) = \mathbf{1}_{R_i}(\mathbf{x_i}) - \mathbf{1}_{R_j}(\mathbf{x})$. Instead of the unsmooth indicator function $\mathbf{1}_{R_i}$, we use the Gaussian kernel, but other smooth kernels could also be used. This leads to the following weights

$$w_{i,n}(\mathbf{x}) = \frac{1 - e^{-\frac{d(\mathbf{x},\mathbf{x_i})^2}{\rho^2}}}{\sum_{j=1}^{n}\left(1 - e^{-\frac{d(\mathbf{x},\mathbf{x_j})^2}{\rho^2}}\right)}. \tag{3.2}$$

Notice that $w_{i,n}(\mathbf{x})$ increases with the distance between the ith design point $\mathbf{x_i}$ and $\mathbf{x}$. In fact, the least weighted prediction is $\hat{s}_{n,-p_{\mathrm{nn}}(\mathbf{x})}$, where $p_{\mathrm{nn}}(\mathbf{x})$ is the index of the nearest design point to $\mathbf{x}$. In general, the prediction $\hat{s}_{n,-i}$ is locally less reliable in a neighborhood of $\mathbf{x_i}$. The proposed weights determine the local relative confidence level of a given submodel prediction. The term "relative" means that the confidence level of one submodel prediction is relative to the remaining submodel predictions due to the normalization factor in (3.2). The smoothing parameter ρ tunes the amount of uncertainty of $\hat{s}_{n,-i}$ in a neighborhood of $\mathbf{x_i}$. Several options are possible to choose ρ. It can be either related to the distance of a point to its nearest neighbor or common for all the points. We suggest using $\rho^\star = \bar{d}(\mathbf{X_n})$. Indeed, this is a well-suited choice for practical cases.

Definition 3.1. *The universal prediction distribution (UP distribution) is the weighted empirical distribution*

$$\mu_{(n,\mathbf{x})}(dy) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})\delta_{\hat{s}_{n,-i}(\mathbf{x})}(dy). \tag{3.3}$$

This probability measure is nothing more than the empirical distribution of all the predictions provided by CV submodels weighted by local smoothed masses.

Definition 3.2. *For $\mathbf{x} \in \mathbb{X}$ we call $\hat{\sigma}^2_n(\mathbf{x})$ (3.5) the local UP variance and $\hat{m}_n(\mathbf{x})$ (3.4) the UP expected value:*

$$\hat{m}_n(\mathbf{x}) = \int y\mu_{(n,\mathbf{x})}(dy) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})\hat{s}_{n,-i}(\mathbf{x}), \tag{3.4}$$

$$\hat{\sigma}^2_n(\mathbf{x}) = \int (y - \hat{m}_n(\mathbf{x}))^2\,\mu_{(n,\mathbf{x})}(dy) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})(\hat{s}_{n,-i}(\mathbf{x}) - \hat{m}_n(\mathbf{x}))^2. \tag{3.5}$$

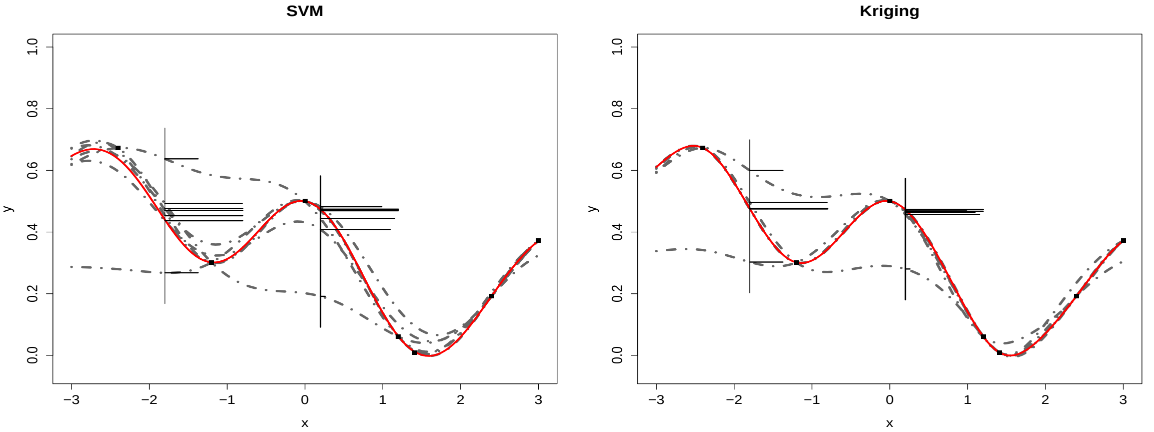


Figure 1. *Illustration of the UP distribution for an SVM surrogate (left) and a kriging surrogate (right). Dashed lines: CV submodel prediction; solid red line: master model prediction; horizontal bars: local UP distribution at $x_a = -1.8$ and $x_b = 0.2$; black squares: design points.*

3.2. Illustrative example.

Let us consider the Viana function defined over $[-3, 3]$,

$$f(\mathbf{x}) = \frac{10\cos(2x) + 15 - 5x + x^2}{50}. \tag{3.6}$$

Let $\mathbf{Z_n} = (\mathbf{X_n}, \mathbf{Y_n})$ be the design of experiments such that

$$\mathbf{X_n} = (x_1 = -2.4, \mathbf{x}_2, = -1.2, x_3 = 0, x_4 = 1.2, x_5 = 1.4, x_6 = 2.4, x_7 = 3)$$

and $\mathbf{Y_n} = (y_1, \ldots, y_7)$ their image by f. We used a Gaussian process regression [27, 21, 18] with constant trend function and Matérn 5/2 covariance function $\hat{s}$ and an SVM regression [36]. We display in Figure 1 the design points, the CV submodel predictions $\hat{s}_{n,-i}$, $i = 1, \ldots, 7$, and the master model prediction $\hat{s}_n$ of each surrogate model.

Notice that in the interval $[1, 3]$ (where we have 4 design points) the discrepancy between the master model and the CV submodels prediction is smaller than in the remaining space. Moreover, we displayed horizontally the *UP distribution* at $x_a = -1.8$ and $x_b = 0.2$ to illustrate the weighting effect. One can notice that

- at x_a the least weighted predictions are $\hat{s}_{n,-1}(x_a)$ and $\hat{s}_{n,-2}(x_a)$. These predictions do not use the two closest design points to x_a: x_1 (respectively, x_2);
- at x_b, $\hat{s}_{n,-3}(x_b)$ is the least weighted prediction.

Furthermore, we display in Figure 2 the master model prediction and the region delimited by $\hat{s}_n(\mathbf{x}) + 3\hat{\sigma}_n(\mathbf{x})$ and $\hat{s}_n(\mathbf{x}) - 3\hat{\sigma}_n(\mathbf{x})$. Contrary to the Gaussian case, this region cannot be interpreted as the 99.7% traditional prediction interval. Nevertheless, it can be interpreted as a prediction interval with level greater than or equal to 88.8%. Indeed, Chebyshev's inequality states that for any squared integrable random variable X and $k > 0$, $\Pr(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$. In particular, $\Pr(|X - \mu| < 3\sigma) \geq 1 - \frac{1}{3^2} \approx 88.8\%$.

Notice that here the UP standard deviation is null at design points for the interpolating surrogate model. In addition, its local maxima in the interval $[1, 3]$ (where we have greater design points density) are smaller than its maxima in the remaining space region.

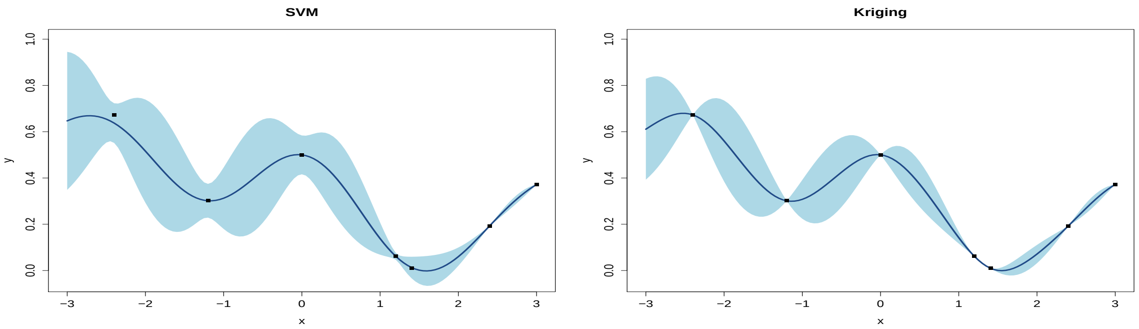


Figure 2. *Uncertainty quantification based on the UP distribution for an SVM surrogate (left) and a kriging surrogate (right). Blue solid line: master model prediction $\hat{s}_n(\mathbf{x})$; light blue area: region delimited by $\hat{s}_n(\mathbf{x}) \pm 3\hat{\sigma}_n(\mathbf{x})$.*

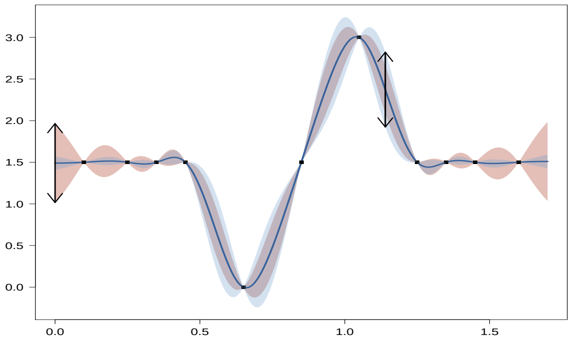


Figure 3. *UP-variance for kriging. Black squares: design points; blue line: $\hat{s}_n(\mathbf{x})$; light blue: area delimited by $\hat{s}_n(\mathbf{x}) \pm 3\hat{\sigma}_n(\mathbf{x})$; light red: area delimited by $\hat{s}_n(\mathbf{x}) \pm 3\hat{\sigma}_{GP}(\mathbf{x})$.*

3.3. UP distribution in action.

Case of the kriging surrogate model. Without loss of generality, let us consider the simple kriging framework. Recall that the conditional mean and variance are given by

$$\begin{aligned} m(\mathbf{x}) &= \mathbf{k}(\mathbf{x})^\top K_n^{-1}\mathbf{Y_n}, \\ \hat{\sigma}^2_{GP}(\mathbf{x}) &= k(\mathbf{x},\mathbf{x}) - \mathbf{k_n}(\mathbf{x})^\top K_n^{-1}\mathbf{k_n}(\mathbf{x}), \end{aligned} \tag{3.7}$$

where $k(\mathbf{x},\mathbf{x'})$ is a covariance function, $\mathbf{k_n}(\mathbf{x})$ is the vector $(k(\mathbf{x},\mathbf{x_1}),\dots,k(\mathbf{x},\mathbf{x_n}))^\top$, and K_n is the invertible matrix with entries $k_{i,j} = k(\mathbf{x_i},\mathbf{x_j})$ for $1 \leq i,j \leq n$.

Notice that for an interpolating kriging, both kriging variance and UP variance vanish at design points. Further, kriging variance does not depend on the output values once the kernel parameters are fixed; however, UP variance does. It is not everywhere smaller or larger than kriging variance; for instance, consider the toy example in Figure 3.

On one hand UP variance is a maximum in the interval $[0.4, 1.3]$. Indeed, if we remove one point in that region we significantly increase the variability of the submodel predictions. Similarly, UP variance is minimum in the nearly linear region $[0, 0.4] \cup [1.3, 2]$. On the other

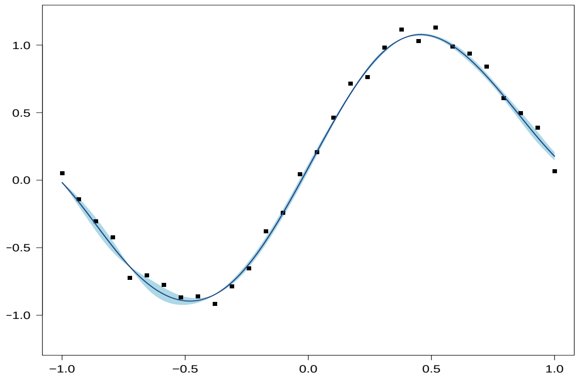


Figure 4. *Illustration of the UP variance using a low variance surrogate model (SVM). Black squares: design points; Blue solid line: master model prediction $\hat{s}_n(\mathbf{x})$; shaded area: delimited by $\hat{s}_n(\mathbf{x}) \pm 3\hat{\sigma}_n(\mathbf{x})$.*

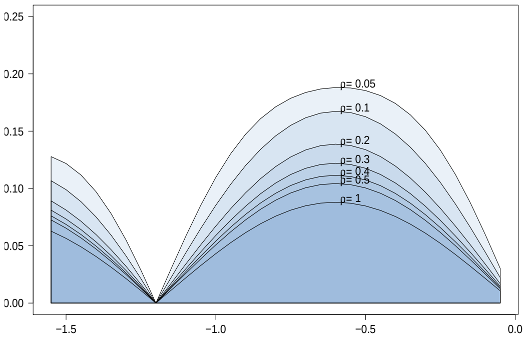


Figure 5. *ρ effect on the UP variance locally for the Viana function using kriging.*

hand, kriging variance is a maximum in that region precisely at $x = 0$. This example highlights how the UP variance of an interpolating kriging is sensitive to output values.

Surrogate inheritance. The UP distribution depends, among others, on the structure of the surrogate model. It inherits some of its properties. For instance, in Figure 4 we consider a set of points sampled from a noisy sinusoid and an SVM that filters the noise. In terms of bias-variance dilemma, the SVM surrogate model may have higher bias and lower variance than interpolating surrogates. As a consequence, the LOO submodel predictions do not vary significantly and the UP variance is low in the whole design space.

Further, let us consider a degenerated regression problem, e.g., 3 aligned points for a first order regression model. The LOO submodel predictions are all the same. The UP-distribution is then degenerated everywhere.

On the role of the bandwidth parameter ρ. Notice that the bandwidth parameter ρ has no effect on the support of the UP distribution. However, it impacts its statistical features. For instance, we notice in Figure 5 that ρ does not modify the general shape of the variance but controls its magnitude.

In section 3.1, we suggested using $\rho^\star = \bar{d}(\mathbf{X_n})$ which depends on the interdistance between design points. Such a value has the desirable property $\rho^\star \to 0$ as $n \to +\infty$ if the design of experiment is dense in the design space. This choice guarantees the convergence of the sequential algorithms presented in sections 4 and 5.

Computational aspects. When the number of design points is large, the computational cost can be a drawback. In fact, the construction time of the submodels is $O(nT)$, where n is the number of data and T is the construction time of one submodel. Nevertheless, for some surrogate models, closed form formulas are available for the LOO-submodel predictions. For instance, Chevalier, Ginsbourger, and Emery [43] presented a formula for kriging. Another way to reduce the computational cost is to use parallel computing where each submodel is computed on a separate thread. Finally, we can replace the use of LOO-CV by the KFCV.

4. Sequential refinement.

In this section, we use the UP distribution to define an adaptive refinement technique called the UP-based surrogate modeling adaptive refinement technique UP-SMART.

4.1. Introduction.

The main goal of sequential design is to minimize the number of calls of a computationally expensive function. Gaussian surrogate models [18] are widely used in adaptive design strategies. Indeed, Gaussian modeling gives a Bayesian framework for sequential design. In some cases, other surrogate models might be more accurate although they do not provide a theoretical framework for uncertainty assessment. We propose here a new universal strategy for adaptive sequential design of experiments. The technique is based on the UP distribution, so, it can be applied to any type of surrogate model.

In the literature, many strategies have been proposed to design the experiments (for an overview, the interested reader is referred to [12, 40, 34]). Some strategies, such as Latin hypercube sampling [28], maximum entropy design [35], and maximin distance designs [16] are called one-shot sampling methods. These methods depend neither on the output values nor on the surrogate model. However, one would naturally expect to design more points in the regions with high nonlinear behavior. This intuition leads to adaptive strategies. A design of experiments approach is said to be adaptive when information from the experiments (inputs and responses) as well as information from surrogate models are used to select the location of the next point.

By adopting this definition, adaptive design of experiments methods include, for instance, surrogate model-based optimization algorithms, probability of failure estimation techniques, and sequential refinement techniques. Sequential refinement techniques aim at creating a more accurate surrogate model. For example, Lin et al. [25] use multivariate adaptive regression splines and kriging models with the sequential exploratory experimental design method. It consists in building a surrogate model to predict errors based on the errors on a test set. Goel et al. [13] use a set of surrogate models to identify regions of high uncertainty by computing the empirical standard deviation of the predictions of the ensemble members. Our method is based on the predictions of the CV submodels. In the literature, several CV-based techniques have been discussed. Li and Azarm [23] propose to add the design point that maximizes the accumulative error (AE). The AE on $\mathbf{x} \in \mathbb{X}$ is computed as the sum of the LOO-CV errors on the design points weighted by influence factors. This method could lead to clustered samples. To avoid this effect, the authors [24] propose to add a threshold constraint in the maximization problem. Busby, Farmer, and Iske [6] propose a method based on a grid and CV. It affects the CV prediction errors at a design point to its containing cell in the grid.

Then, an entropy approach is performed to add a new design point. More recently, Xu et al. [42] suggest the use of a method based on Voronoi cells and CV. Kleijnen and Van Beers [19] propose a method based on the jackknife's pseudovalues predictions variance. Jin, Chen, and Sudjianto [15] present a strategy that maximizes the product between the deviation of CV submodel predictions with respect to the master model prediction and the distance to the design points. Aute et al. [2] introduce the space-filling CV trade-off approach. It consists in building a new surrogate model over LOO-CV errors and then adding a point that maximizes the new surrogate model prediction under some space-filling constraints. In general CV-based approaches tend to allocate points close to each other resulting in clustering [2]. This is not desirable for deterministic simulations.

4.2. UP-SMART. The idea behind UP-SMART is to sample points where the UP distribution variance (3.5) is maximal. Most of the CV-based sampling criteria use CV errors. Here, we use the local predictions of the CV submodels. Moreover, notice that the UP variance is null on design points for interpolating surrogate models. Hence, UP-SMART does not naturally promote clustering.

However, $\hat{\sigma}_n^2(\mathbf{x})$ can vanish even if $\mathbf{x}$ is not a design points. To overcome this drawback, we add a distance penalization. This leads to the UP-SMART sampling criterion γ_n (4.1)

$$\gamma_n(\mathbf{x}) = \hat{\sigma}_n^2(\mathbf{x}) + \delta \underline{d}_{\mathbf{X_n}}(\mathbf{x}), \tag{4.1}$$

where $\delta > 0$ is called the exploration parameter. One can set δ as a small percentage of the global variation of the output. UP-SMART is the adaptive refinement algorithm consisting in adding at step n a point $x_{n+1} \in \arg\max_{\mathbf{x}\in\mathbb{X}}(\gamma_n(\mathbf{x}))$.

4.3. Performances on a set of test functions. In this subsection, we present the performance of the UP-SMART. We present first the used surrogate models.

4.3.1. Used surrogate models.

Kriging. Kriging [27] or Gaussian process regression is an interpolation method. Universal kriging fits the data using a deterministic trend and is governed by prior covariances. Let $k(\mathbf{x}, \mathbf{x}')$, be a covariance function on $\mathbb{X} \times \mathbb{X}$, and let $(h_i)_{1\leq i \leq p}$ be the basis functions of the trend. Let us denote $\mathbf{h}(\mathbf{x})$ by the vector $(h_1(\mathbf{x}), .., h_p(\mathbf{x}))^\top$ and let H be the matrix with entries $h_{ij} = h_j(\mathbf{x_i}), 1 \leq i, j \leq n$. Furthermore, let $\mathbf{k_n}(\mathbf{x})$ be the vector $(k(\mathbf{x}, \mathbf{x_1}), .., k(\mathbf{x}, \mathbf{x_n}))^\top$ and K_n the matrix with entries $k_{i,j} = k(\mathbf{x_i}, \mathbf{x_j})$ for $1 \leq i, j \leq n$.

Then, the conditional mean of the Gaussian process with covariance $k(\mathbf{x}, \mathbf{x}')$ and its variance are given in (4.2), (4.3),

$$m_{G_n}(\mathbf{x}) = \mathbf{h}(\mathbf{x})^\top \hat{\beta} + \mathbf{k_n}(\mathbf{x})^\top K_n^{-1}(Y - H^\top \hat{\beta}), \tag{4.2}$$

$$\sigma^2_{GP_n}(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k_n}(\mathbf{x})^\top K_n^{-1} \mathbf{k_n}(\mathbf{x})^\top + \mathbf{V}(\mathbf{x})^\top (H^\top K_n^{-1} H)^{-1} \mathbf{V}(\mathbf{x}), \tag{4.3}$$

$$\hat{\beta} = (H^\top K_n^{-1} H)^{-1} H^\top K_n^{-1} Y \text{ and } \mathbf{V}(\mathbf{x}) = \mathbf{h}(\mathbf{x})^\top + \mathbf{k_n}(\mathbf{x})^\top K_n^{-1} H. \tag{4.4}$$

Note that the conditional mean is the prediction of the Gaussian process regression. Further, we used two kriging instances with different sampling schemes in our test bench. Both use a constant trend function and a Matérn 5/2 covariance function. The first design is obtained by maximizing the UP distribution variance (3.5). And the second one is obtained by maximizing the kriging variance $\sigma^2_{GP_n}(\mathbf{x})$.

Table 1
Used test functions.

Function	Dimension d	n_0	N_{max}	N_t
Viana	1	5	7	500
Branin	2	10	10	1600
Camel	2	20	10	1600
Extended Rosenbrock	6→10	60	100	10000
Hartmann 6-dim	6→10	100	100	10000

Genetic aggregation. The genetic aggregation response surface is a method that aims at selecting the best response surface for a given design of experiments. It uses several surrogate models. In our examples, we use several kriging and SVM models with different settings (kernels, trend functions, ...) and select the best weighted aggregation:

$$\hat{A}_m(\mathbf{x}) = \sum_{l=1}^{m} \omega_l \hat{s}^{(l)}(\mathbf{x}). \tag{4.5}$$

$\hat{s}^{(l)}$ are the surrogate models and the weights ω_l are computed in order to minimize a criterion combining CV errors, surrogate model errors, and a smoothness penalty. The use of such a response surface, in this test bench, aims at checking the universality of the UP distribution: the fact that it can be applied for all types of surrogate models.

4.3.2. Test bench. In order to test the performances of the method we launched different refinement processes for the following set of test functions:

- Branin: $f_b(x_1, x_2) = (x_2 - (\frac{5.1}{4\pi^2})x_1^2 + (\frac{5}{\pi})x_1 - 6)^2 + 10(1 - (\frac{1}{8\pi}))\cos(x_1) + 10$.
- Six-hump camel: $f_c(x_1, x_2) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + x_2^2(4x_2^2 - 4)$.
- Hartmann6: $f_h(\mathbf{X} = (x_1, \dots, x_6)) = -\sum_{i=1}^{4} \alpha_i \exp(-\sum_{j=0}^{6} A_{ij}(x_j - P_{ij})^2)$. A,P, and α can be found in [9].
- Viana: (3.6)

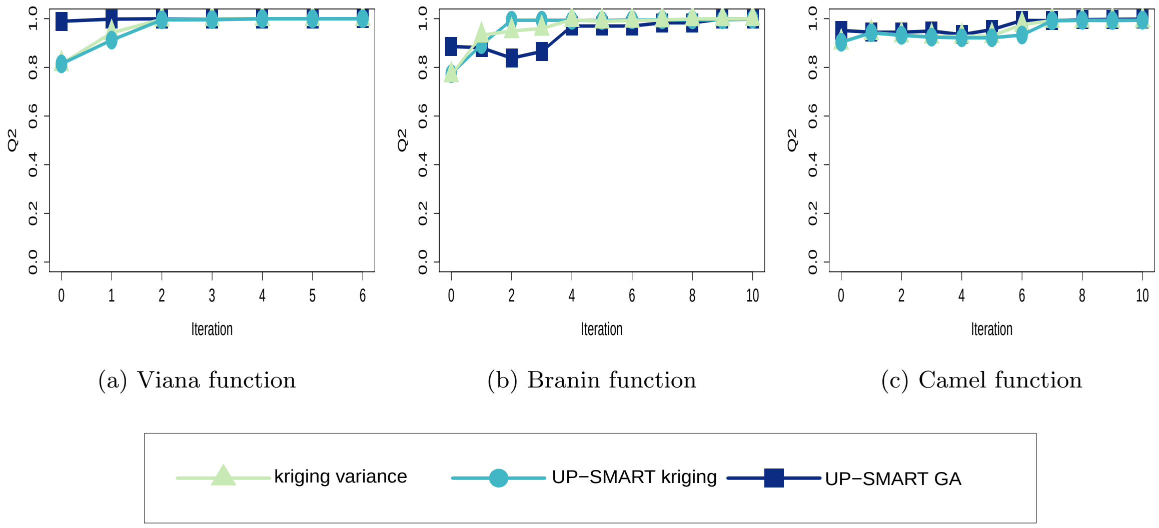
For each function we generated by optimal Latin hypersampling design the number of initial design points n_0, the number of refinement points N_{max}. We also generated a set of N_t test points and their response $Z^{(t)} = (X^{(t)}, Y^{(t)})$. The used values are available in Table 1.

We fixed n_0 in order to get nonaccurate surrogate models at the first step. Usually, one follows the rule-of-thumb $n_0 = 10 \times d$ proposed in [26]. However, for Branin and Viana functions, this rule leads to a very good initial fit. Therefore, we choose lower values.

- Kriging variance-based refinement process (4.3) as refinement criterion.
- Kriging using the UP-SMART: UP-variance as refinement criterion (4.1).
- Genetic aggregation using the UP-SMART: UP-variance as refinement criterion (4.1).

4.3.3. Results. For each function, we compute at each iteration the Q squared (Q^2) of the predictions of the test set $Z^{(t)}$, where

$$Q^2(\hat{s}, Z^{(t)}) = 1 - \frac{\sum_{i=1}^{N_t}(y_i^{(t)} - \hat{s}(\mathbf{x_i}^{(t)}))^2}{\sum_{i=1}^{N_t}(y_i^{(t)} - \bar{y})^2} \quad \text{and} \quad \bar{y} = \frac{1}{N_t}\sum_{i=1}^{N_t} y_i^{(t)}.$$



(a) Viana function (b) Branin function (c) Camel function

Figure 6. *Performance of three refinement strategies on three test functions measured by the Q^2 criterion on a test set. x axis: number of added refinement points.*

We display in Figure 6 the performances of the three different techniques described above for Viana (Figure 6(a)), Branin (Figure 6(b)), and Camel (Figure 6(c)) functions measured by the Q^2 criterion.

For these tests, the three techniques have comparable performances. The Q^2 converges for all of them. It appears that the UP variance criterion refinement process gives as good a result as the kriging variance criterion. In higher dimensions, we perform for each dimension (from 6 to 10) 10 tests with different initial design of experiments. The sequential algorithm based on kriging variance generates more points on the boundaries. This may be a good strategy when there is significant variability on the boundaries. On one hand consider the extended Rosenbrock function Figure 7(a). As the function varies significantly on the boundaries, UP-SMART and kriging variance strategies have comparable performances. On the other hand, when the function does not vary much on the boundaries such as Hartman, Figure 7(b), UP-SMART outperforms the kriging variance strategy.

The results show the following:

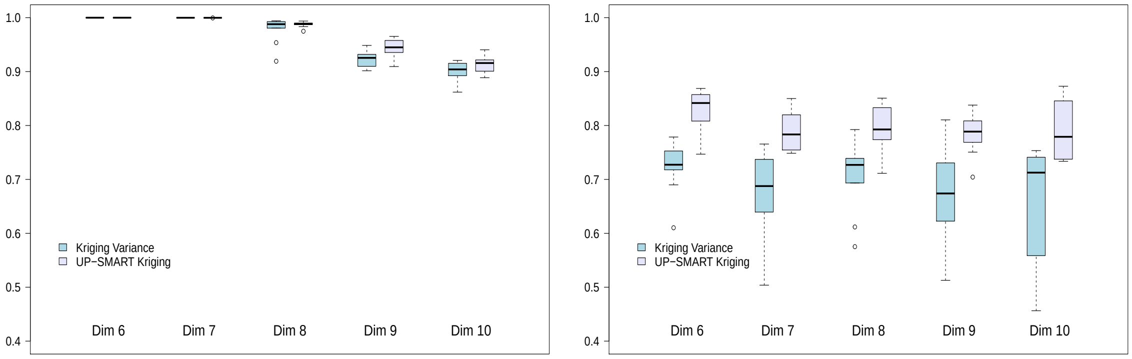
- UP-SMART gives for some problems better global response surface accuracy than the maximization of the kriging variance. This shows the usefulness of the method.
- UP-SMART is a universal method. Here, it has been applied with success to an aggregation of response surfaces. Such usage highlights the universality of the strategy.

5. Empirical EGO.

In this section, we introduce UP distribution-based EGO (UP-EGO) algorithm. This algorithm is an adaptation of the well-known EGO algorithm.

5.1. Overview.

Surrogate model-based optimization refers to the idea of speeding optimization processes using surrogate models. In this section, we present an adaptation of the well-known EGO algorithm [17]. Our method is based on the weighted empirical distribution



(a) Rosenbrock function

(b) Hartmann function

Figure 7. *Performance of refinement strategies for different dimension on two test functions measured by the Q^2 on a test set UP-SMART with kriging in blue and kriging variance-based technique in violet.*

UP distribution. We show that asymptotically, the points generated by the algorithm are dense around the optimum. For EGO, such result was proved by Vazquez and Bect [37].

The basic unconstrained surrogate model-based optimization scheme can be summarized as follows [30]:

- Construct a surrogate model from a set of known data points.
- Define a sampling criterion that reflects a possible improvement.
- Optimize the criterion over the design space.
- Evaluate the true function at the criterion optimum/optima.
- Update the surrogate model using new data points.
- Iterate until convergence.

Several sampling criteria have been proposed to perform optimization. The EI is one of the most popular criteria for surrogate model-based optimization. Sasena, Papalambros, and Goovaerts [33] discussed some sampling criteria such as the threshold-bounded extreme, the regional extreme, the generalized EI, and the minimum surprises criterion. Almost all of the criteria are computed in practice within the frame of Gaussian processes. Consequently, among all possible response surfaces, Gaussian surrogate models are widely used in surrogate model-based optimization. Recently, Viana, Haftka, and Watson [39] performed multiple surrogate-based optimization by importing a Gaussian uncertainty estimate.

5.2. UP-EGO algorithm. Here, we use the UP distribution to compute an empirical EI. Then, we present an optimization algorithm similar to the original EGO algorithm that can be applied with any type of surrogate models. Without loss of generality, we consider the minimization problem

$$\underset{\mathbf{x}\in\mathbb{X}}{\text{minimize}} \qquad s(\mathbf{x}).$$

Let $(y(\mathbf{x}))_{\mathbf{x}\in\mathbb{X}}$ be a Gaussian process model. m_{G_n} and $\sigma^2_{GP_n}$ denote, respectively, the mean and the variance of the conditional process $y(\mathbf{x}) \mid \mathbf{Z_n}$. Further, let $y^\star_n$ be the minimum value at step n when using observations $\mathbf{Z_n} = (z_1, \dots, z_n)$, where $z_i = (\mathbf{x_i}, y_i)$ ($y_n^\star = \min_{i=1..n} y_i$). The EGO algorithm [17] uses the EI EI_n (5.1) as sampling criterion:

$$EI_n(\mathbf{x}) = \mathbb{E}[\max(y_n^\star - y(\mathbf{x}), 0) \mid \mathbf{Z_n}]. \tag{5.1}$$

The EGO algorithm adds the point that maximizes EI_n. Using some Gaussian computations, (5.1) is equivalent to (5.2):

$$EI_n(\mathbf{x}) = \begin{cases} (y_n^\star - m_{G_n}(\mathbf{x}))\Phi\left(\dfrac{y_n^\star - m_{G_n}(\mathbf{x})}{\sigma_{GP_n}(\mathbf{x})}\right) + \sigma_{GP_n}(\mathbf{x})\phi\left(\dfrac{y_n^\star - m_{G_n}(\mathbf{x})}{\sigma_{GP_n}(\mathbf{x})}\right) & \text{if } \sigma_{GP_n}(\mathbf{x}) \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

We introduce a similar criterion based on the UP distribution. With the notations of sections 2 and 3, EEI_n (5.3) is called the empirical EI:

$$\begin{aligned} EEI_n(\mathbf{x}) &= \int \max(y_n^\star - y, 0)\mu_{(n,\mathbf{x})}(dy) \\ &= \sum_{i=1} w_{i,n}(\mathbf{x}) \max(y_n^\star - \hat{s}_{n,-i}(\mathbf{x}), 0). \end{aligned} \tag{5.3}$$

We can remark that $EEI_n(\mathbf{x})$ can vanish even if $\mathbf{x}$ is not a design point. This is one of the limitations of the empirical UP distribution. To overcome this drawback, we suggest the use of the UP-EI κ_n (5.4),

$$\kappa_n(\mathbf{x}) = EEI_n(\mathbf{x}) + \xi_n(\mathbf{x}), \tag{5.4}$$

where $\xi_n(\mathbf{x})$ is a distance penalization. We use $\xi_n(\mathbf{x}) = \delta \underline{d}_{\mathbf{X_n}}(\mathbf{x})$, where $\delta > 0$ is called the exploration parameter. One can set δ as a small percentage of the global variation of the output for less exploration. Greater value of δ means more exploration. δ fixes the wished trade-off between exploration and local search.

Furthermore, notice that κ_n has the desirable property also verified by the usual EI.

Proposition 5.1. $\forall n > 1, \forall \mathbf{Z_n} = (\mathbf{X_n} = (\mathbf{x_1}, \dots, \mathbf{x_n})^\top, \mathbf{Y_n} = s(\mathbf{X_n}))$, *if the used model interpolates the data then* $\kappa_n(\mathbf{x_i}) = 0$ *for* $i = 1, \dots, n$.

The UP-EGO (Algorithm 1) consists in sampling at each iteration the point that maximizes κ_n. The point is then added to the set of observations and the surrogate model is updated.

5.3. UP-EGO convergence. We first recall the context. $\mathbb{X}$ is a nonempty compact subset of the Euclidean space $\mathbb{R}^p$, where $p \in \mathbb{N}^\star$. s is an expensive-to-evaluate function. The weights of the UP distribution are computed as in (3.2) with $\rho > 0$ a fixed real parameter. Moreover, we consider the asymptotic behavior of the algorithm so that, here, the number of iterations goes to infinity.

Let $\mathbf{x}^\star \in \arg\min\{s(\mathbf{x}), \mathbf{x} \in \mathbb{X}\}$ and $\hat{s}$ be a continuous interpolating surrogate model bounded on $\mathbb{X}$. Let $\mathbf{Z}_{n_0} = (X_{n_0} = (\mathbf{x_1}, \dots, \mathbf{x_{n_0}})^\top, Y_{n_0})$ be the initial data. For all $k > n_0$, $\mathbf{x_k}$

Algorithm 1 UP-EGO($\hat{s}$).

```
Inputs: $\mathbf{Z}_{n_0} = (X_{n_0}, Y_{n_0})$, $n_0 \in \mathbb{N} \setminus \{0, 1\}$ and a deterministic function $s$
(1) $m := n_0$, $\mathbf{S}_m := X_{n_0}$, $Y_m := Y_{n_0}$
(2) Compute the surrogate model $\hat{s}_{\mathbf{Z}_m}$
(3) $Stop\_conditions$ := False
(4) While Stop_conditions are not satisfied
    (4.1) Select $\mathbf{x_{m+1}} \in arg \max_{\mathbb{X}}(\kappa_m(\mathbf{x}))$
    (4.2) Evaluate $y_{m+1} := s(\mathbf{x_{m+1}})$
    (4.3) $\mathbf{S}_{m+1} := \mathbf{S}_m \cup \{\mathbf{x_{m+1}}\}$, $Y_{m+1} := Y_m \cup \{y_{m+1}\}$
    (4.4) $\mathbf{Z}_{m+1} := (\mathbf{S}_{m+1}, Y_{m+1})$,$m := m + 1$
    (4.5) Update the surrogate model
    (4.6) Check $Stop\_conditions$
  end loop
Outputs: $\mathbf{Z}_m := (\mathbf{S}_m, Y_m)$, surrogate model $\hat{s}_{\mathbf{Z}_m}$
```

denotes the point generated by the UP-EGO algorithm at step $k - n_0$. Let $\mathbf{S}_m$ denote the set $\{\mathbf{x_i}, i \leq m\}$ and $S = \{\mathbf{x_i}, i > 0\}$. Finally, $\forall m > n_0$ we note κ_m the UP-EI of $\hat{s}_{\mathbf{Z}_m}$. We are going to prove that $\mathbf{x}^\star$ is adherent to the sequence S generated by the UP-EGO($\hat{s}$) algorithm.

Lemma 5.2. $\exists \theta > 0$, $\forall m \geq n_0$, $\forall \mathbf{x} \in \mathbb{X}$, $\forall i \in 1, \ldots, m$, $\forall n > m$, $w_{i,n}(\mathbf{x}) \leq \theta d(\mathbf{x}, \mathbf{x_i})^2$.

Definition 5.3. *A surrogate model $\hat{s}$ is called an interpolating surrogate model if for all $n \in \mathbb{N}^\star$ and for all $\mathbf{Z_n} = (\mathbf{X_n}, \mathbf{Y_n}) \in \mathbb{X}^n \times \mathbb{R}^n$, $\hat{s}_{\mathbf{Z_n}}(\mathbf{x}) = s(\mathbf{x})$ if $\mathbf{x} \in \mathbf{X_n}$.*

Definition 5.4. *A surrogate model $\hat{s}$ is called bounded on $\mathbb{X}$ if for all s a continuous function on $\mathbb{X}$, $\exists L, U$, such that for all $n > 1$ and for all $\mathbf{Z_n} = (\mathbf{X_n}, \mathbf{Y_n} = s(\mathbf{X_n})) \in \mathbb{X}^n \times \mathbb{R}^n$, $\forall \mathbf{x} \in \mathbb{X}$, $L \leq \hat{s}_{\mathbf{Z_n}}(\mathbf{x}) \leq U$.*

Definition 5.5. *A surrogate model $\hat{s}$ is called continuous if $\forall n_0 > 1$ $\forall \mathbf{x} \in \mathbb{X}$ $\forall \varepsilon > 0$, $\exists \delta > 0$, $\forall n \geq n_0$, $\forall \mathbf{Z_n} = (\mathbf{X_n}, \mathbf{Y_n}) \in \mathbb{X}^n \times \mathbb{R}^n$, $\forall \mathbf{x}' \in \mathbb{X}$, $d(\mathbf{x}, \mathbf{x}') < \delta \implies |\hat{s}_{\mathbf{Z_n}}(\mathbf{x}) - \hat{s}_{\mathbf{Z_n}}(\mathbf{x}')| < \varepsilon$.*

Theorem 5.6. *Let s be a real function defined on $\mathbb{X}$ and let $\mathbf{x}^\star \in \arg\min\{s(\mathbf{x}), \mathbf{x} \in \mathbb{X}\}$. If $\hat{s}$ is an interpolating continuous surrogate model bounded on $\mathbb{X}$, then $\mathbf{x}^\star$ is adherent to the sequence of points S generated by UP-EGO($\hat{s}$).*

The proofs (section 9) show that the exploration parameter is important for this theoretical result. In our implementation, we scale the input spaces to be the hypercube $\big[-1, 1\big]$ and we set δ to 0.005% of the output variation. Hence, the exploratory effect only slightly impacts the UP-EI criterion in practical cases.

5.4. Numerical examples.

Let us consider the set of test functions (Table 2).

We launched the optimization process for these functions with three different optimization algorithms:

- EGO [17] implementation of the R package DiceOptim [32] using the default parameters.
- UP-EGO algorithm applied to a universal kriging surrogate model $\hat{s_k}$ that uses a Matérn 5/2 covariance function and a constant trend function. We denote this algorithm UP-EGO($\hat{s_k}$).

Table 2
Optimization test functions.

Function $f^{(i)}$	Dimension $d^{(i)}$	Number of initial points $n_0^{(i)}$	Number of iterations $N_{max}^{(i)}$
Branin	2	5	40
Ackley	2	10	30
Six-hump camel	2	10	30
Hartmann 6-dim	6	20	40

- UP-EGO algorithm applied to the genetic aggregation $\hat{s_a}$. It is then denoted UP-EGO($\hat{s_a}$).

For each function $f^{(i)}$, we launched each optimization process for $N_{max}^{(i)}$ iterations starting with $N_{seed} = 20$ different initial design of experiments of size $n_0^{(i)}$ generated by an optimal space-filling sampling. The results are given using box plots in Appendix A. We also display the mean best value evolution in Figure 8.

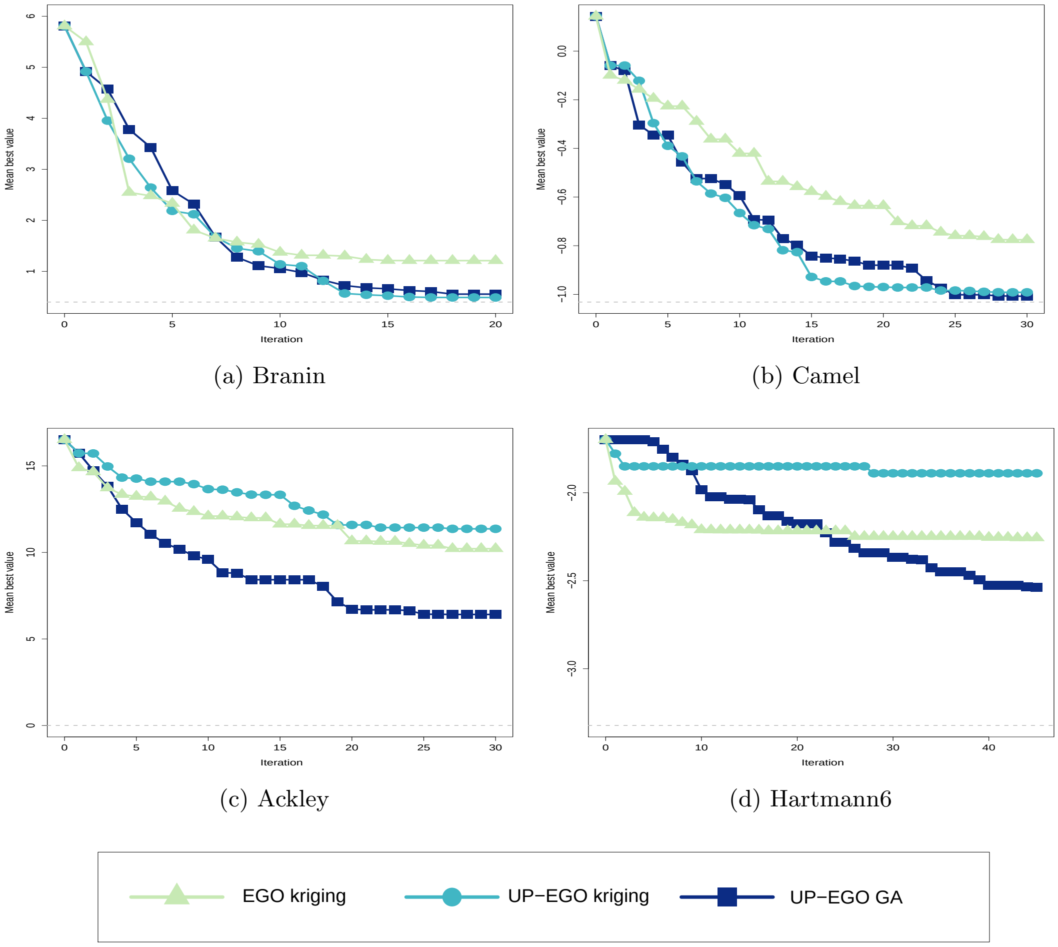
The results show that the UP-EGO algorithms give better results than the EGO algorithm for the Branin and Camel functions. These cases illustrate the efficiency of the method. Moreover, for Ackley and Harmtann6 functions the best results are given by UP-EGO using the genetic aggregation. Even if this is related to the nature of the surrogate model, it underlines the efficient contribution of the universality of UP-EGO. Further, let us focus on the box plots of the last iterations of Figures 11 and 14 (Appendix A). It is important to notice that UP-EGO results for the Branin function depend slightly on the initial design points. On the other hand, let us focus on the Hartmann function case. The results of UP-EGO using the genetic aggregation depend on the initial design points. In fact, more optimization iterations are required for full convergence. However, compared to the EGO algorithm, the UP-EGO algorithm has good performances for both cases:

- full convergence;
- limited-budget optimization.

Otherwise, the Branin function has multiple solutions. We are interested in checking whether the UP-EGO algorithm would focus on one local optimum or on the three possible regions. We present in Figure 9 the spatial distribution of the generated points by the UP-EGO (kriging) algorithm for the Branin function. We can notice that UP-EGO generated points around the three local minima.

6. Fluid simulation application: Mixing tank. The problem addressed here concerns a static mixer where hot and cold fluid enter at variable velocities (Figure 10). The objective of this analysis is generally to find inlet velocities that minimize pressure loss from the cold inlet to the outlet and minimize the temperature spread at the outlet. In our study, we are interested in a better exploration of the design using an accurate cheap-to-evaluate surrogate model.

The simulations are computed within the ANSYS Workbench environment and we used DesignXplorer to perform surrogate modeling. We started the study using 9 design points generated by a central composite design. We produced also a set of $N_t = 80$ test points $Z_t = (X_t = (x_1^{(t)}), \dots, \mathbf{x}_{N_t}^{(t)}), Y_t = (y_1^{(t)}), \dots, y_{N_t}^{(t)}))$. We launched UP-SMART applied to the



(a) Branin

(b) Camel

(c) Ackley

(d) Hartmann6

Figure 8. *Comparison of* 3 *surrogate-based optimization strategies. Mean over* N_{seed} *of the best value as a function of the number of iterations.*

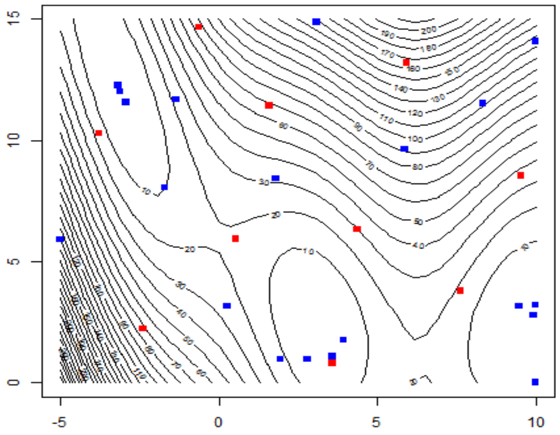


Figure 9. *Example of sequence generated by the UP-EGO(kriging) algorithm on the Branin function. Initial design points are in red, added points are in blue.*

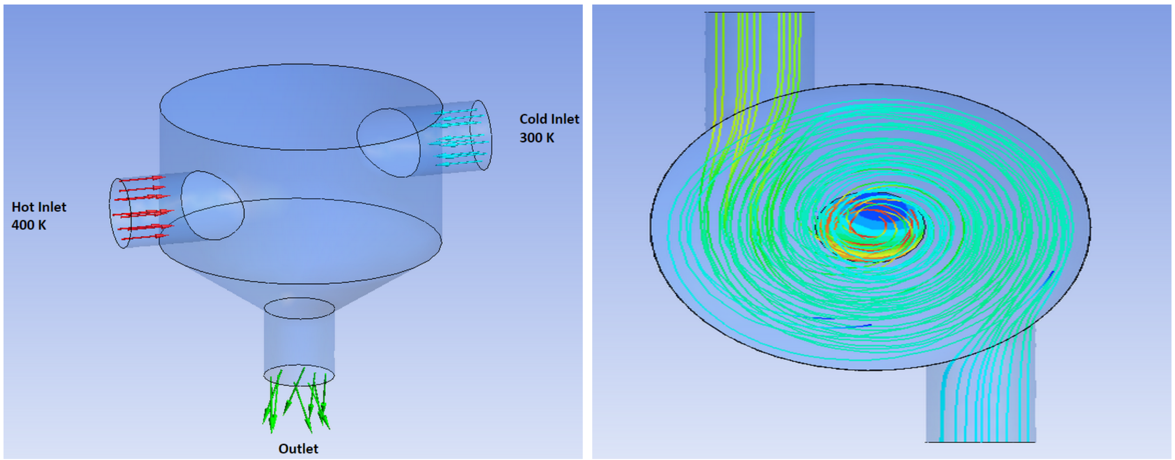


Figure 10. *Mixing tank.*

genetic aggregation response surface (GARS) in order to generate 10 suitable design points and a kriging-based refinement strategy. The GARS developed by DesignXplorer creates a mixture of surrogate models including SVM regression, Gaussian process regression, moving least squares, and polynomial regression. We computed the root mean square error (6.1), the relative root mean square error (6.2), and the relative average absolute error (6.3) before and after launching the refinement processes:

$$RMSE_{Z^{(t)}}(\hat{s}) = \frac{1}{N^t}\sum_{i=1}^{N_t}(y_i^{(t)} - \hat{s}(\mathbf{x_i}^{(t)}))^2, \tag{6.1}$$

$$RRMSE_{Z^{(t)}}(\hat{s}) = \frac{1}{N^t}\sum_{i=1}^{N_t}\left(\frac{y_i^{(t)} - \hat{s}(\mathbf{x_i}^{(t)})}{y_i^{(t)}}\right)^2, \tag{6.2}$$

$$RAAE_{Z^{(t)}}(\hat{s}) = \frac{1}{N^t}\sum_{i=1}^{N_t}\frac{\mid y_i^{(t)} - \hat{s}(\mathbf{x_i}^{(t)})\mid}{\sigma_Y}. \tag{6.3}$$

We give in Table 3 the obtained quality measures for the temperature spread output. In fact, the pressure loss is nearly linear and every method gives a good approximation.

Table 3
Quality measures of different response surfaces of static mixer simulations.

Surrogate model	RRMSE	RMSE	RAAE
GARS initial	0.16	0.10	0.50
GARS final	0.10	0.07	0.31
Kriging initial	0.16	0.11	0.48
Kriging final	0.16	0.11	0.50

The results show that UP-SMART gives a better approximation. Here, it is used with a genetic aggregation of several response surfaces. Even if the good quality may be due to the response surface itself, it highlights the fact that UP-SMART made the use of such a surrogate model-based refinement strategy possible.

7. Empirical inversion.

7.1. Empirical inversion criteria adaptation. Inversion approaches consist in the estimation of contour lines, excursion sets, or probability of failure. These techniques are specially used in constrained optimization and reliability analysis.

Several iterative sampling strategies have been proposed to handle these problems. The empirical distribution $\mu_{n,\mathbf{x}}$ can be used for inversion problems. In fact, we can compute most of the well-known criteria such as the Bichon's criterion [4] or the Ranjan's criterion [31] using the UP distribution. In this section, we discuss some of these criteria: the targeted mean square error (TMSE) [29], Bichon [4], and the Ranjan criteria [31]. The reader can refer to Chevalier, Picheny, and Ginsbourger [7] for an overview.

Let us consider the contour line estimation problem: let T be a fixed threshold. We are interested in enhancing the surrogate model accuracy in $\{\mathbf{x} \in \mathbb{X}, s(\mathbf{x}) = T\}$ and in its neighborhood.

TMSE. TMSE [29] aims at decreasing the mean square error where the kriging prediction is close to T.

It is probable that the response lies inside the interval $\left[T-\varepsilon, T+\varepsilon\right]$, where the parameter $\varepsilon > 0$ tunes the size of the window around the threshold T. High values make the criterion more exploratory while low values concentrate the evaluation around the contour line.

We can compute an estimation of the value of this criterion using the UP distribution (7.1):

$$\begin{aligned} TMSE_{T,n}(\mathbf{x}) &= \sum_{i=1}^{n} w_{i,n}(\mathbf{x}) 1_{\left[T-\varepsilon, T+\varepsilon\right]}\left(\hat{s}_{n,-i}(\mathbf{x})\right) \\ &= \sum_{i=1}^{n} w_{i,n}(\mathbf{x}) 1_{\left[-\varepsilon, \varepsilon\right]}\left(\hat{s}_{n,-i}(\mathbf{x}) - T\right). \end{aligned} \tag{7.1}$$

Notice that the last criterion takes into account neither the variability of the predictions at $\mathbf{x}$ nor the magnitude of the distance between the predictions and T.

Bichon criterion. The expected feasibility defined in [4] aims at indicating how well the true value of the response is expected to be close to the threshold T.

The bounds are defined by $\varepsilon_{\mathbf{x}}$ which is proportional to the kriging standard deviation $\hat{\sigma}(\mathbf{x})$. Bichon proposes using $\varepsilon_{\mathbf{x}} = 2\hat{\sigma}(\mathbf{x})$ [4].

This criterion can be extended to the case of the UP distribution. We define in (7.2) EF_n the empirical Bichon's criterion, where $\varepsilon_{\mathbf{x}}$ is proportional to the empirical standard deviation $\hat{\sigma}_n^2(\mathbf{x})$ (3.5):

$$EF_n(\mathbf{x}) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})(\varepsilon_{\mathbf{x}} - |T - \hat{s}_{n,-i}(\mathbf{x})|)1_{[-\varepsilon_{\mathbf{x}},\varepsilon_{\mathbf{x}}]}(\hat{s}_{n,-i}(\mathbf{x}) - T). \tag{7.2}$$

Ranjan criterion. Ranjan, Bingham, and Michailidis [31] proposed a criterion that quantifies the improvement $I_{Ranjan}(\mathbf{x})$ defined in (7.3):

$$I_{Ranjan}(\mathbf{x}) = \left(\varepsilon_{\mathbf{x}}^2 - (y(\mathbf{x}) - T)^2\right) 1_{[-\varepsilon_{\mathbf{x}},\varepsilon_{\mathbf{x}}]}(y(\mathbf{x}) - T), \tag{7.3}$$

where $\varepsilon_{\mathbf{x}} = \alpha\hat{\sigma}(\mathbf{x})$ and $\alpha > 0$. $\varepsilon_{\mathbf{x}}$ defines the size of the neighborhood around the contour T.

It is possible to compute the UP distribution-based Ranjan's criterion (7.4). Note that we set $\varepsilon_{\mathbf{x}} = \alpha\hat{\sigma}_n^2(\mathbf{x})$:

$$\mathbb{E}\Big[I_{Ranjan}(\mathbf{x})\Big] = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})\Big(\varepsilon_{\mathbf{x}}^2 - (\hat{s}_{n,-i}(\mathbf{x}) - T)^2\Big)1_{[-\varepsilon_{\mathbf{x}},\varepsilon_{\mathbf{x}}]}(\hat{s}_{n,-i}(\mathbf{x}) - T). \tag{7.4}$$

7.2. Discussion.

The use of the pointwise criteria (7.1), (7.2), (7.4) might face problems when the region of interest is relatively small compared to the prediction jumps. In fact, as the cumulative distribution function of the UP distribution is a step function, the probability of the prediction being inside an interval can vanish even if it is around the mean value. For instance $\mu_{n,\mathbf{x}}\big(y(\mathbf{x}) \in [T-\varepsilon, T+\varepsilon]\big)$ can be zero. This is one of the drawbacks of the empirical distribution. Some regularization techniques are possible to overcome this problem; for instance, the technique that consists in defining the region of interest by a Gaussian density $\mathcal{N}(0,\sigma_\varepsilon^2)$ [29]. Let g_ε be this Gaussian probability distribution function.

The new TMSE denoted $TMSE_{T,n}^{(2)}(\mathbf{x})$ criterion is then, as in (7.5),

$$TMSE_{T,n}^{(2)}(\mathbf{x}) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x}) g_\varepsilon\big(\hat{s}_{n,-i}(\mathbf{x}) - T\big). \tag{7.5}$$

The use of the Gaussian density to define the targeted region seems more relevant when using the UP local variance. Similarly, we can apply the same method to the Ranjan's and Bichon's criteria.

8. Conclusion.

To perform surrogate model-based sequential sampling, several relevant techniques are required to quantify the prediction uncertainty associated with the model. Gaussian process regression provides directly this uncertainty quantification. This is the reason why Gaussian modeling is quite popular in sequential sampling. In this work, we defined a universal approach for uncertainty quantification that could be applied for any surrogate model. It is based on a weighted empirical probability measure supported by CV submodel predictions.

Hence, one could use this distribution to compute most of the classical sequential sampling criteria. As examples, we discussed sampling strategies for refinement, optimization, and inversion. Further, we showed that, under some assumptions, the optimum is adherent to the sequence of points generated by the optimization algorithm UP-EGO. Moreover, the optimization and the refinement algorithms were successfully implemented and tested both on single and multiple surrogate models. We also discussed the adaptation of some inversion criteria. The main drawback of the UP distribution is that it is supported by a finite number of points. To avoid this, we propose to regularize this probability measure. In a future work, we will study and implement such a regularization scheme and extend its applications to other application such as multiobjective constrained optimization and reliability-based design optimization.

9. Proofs.

We present in this section the proofs of Proposition 5.1, Lemma 5.2, and Theorem 5.6. Here, we use the notations of section 5.3.

Proof of Proposition 5.1. Let $n > 1$, $\mathbf{Z_n} = (\mathbf{X_n} = (\mathbf{x_1}, \dots, \mathbf{x_n})^\top, \mathbf{Y_n} = s(\mathbf{X_n}))$, and $\hat{s}$ a model that interpolates the data, i.e., $\forall i \in 1, \dots, n$, $\hat{s}_{\mathbf{Z_n}}(\mathbf{x_i}) = s(\mathbf{x_i}) = y_i$.

First, we have $\xi_n(\mathbf{x_i}) = \delta \underline{d}_{\mathbf{X_n}}(\mathbf{x_i})$. Since $\mathbf{x_i} \in \mathbf{X_n}$ then $\xi_n(\mathbf{x_i}) = 0$. Further,

$$EEI_n(\mathbf{x_i}) = w_{i,n}(\mathbf{x_i}) \max(y_n^\star - \hat{s}_{n,-i}(\mathbf{x_i}), 0) + \sum_{\substack{j=1 \\ j \neq i}}^{n} w_{j,n}(\mathbf{x_i}) \max(y_n^\star - y_i, 0).$$

Notice that

- $w_{i,n}(\mathbf{x_i}) = 0$,
- $\max(y_n^\star - y_i, 0) = 0$.

Then $EEI_n(\mathbf{x_i}) = 0$. Finally, $\kappa_n(\mathbf{x_i}) = EEI_n(\mathbf{x_i}) + \xi_n(\mathbf{x_i}) = 0$. ■

Proof of Lemma 5.2. Let us note

- $\phi_\rho(\mathbf{x}, \mathbf{x}') = 1 - e^{-\frac{d((\mathbf{x},\mathbf{x}'))^2}{\rho^2}}$,
- $w_{i,n}(\mathbf{x}) = \frac{\phi_\rho(\mathbf{x},\mathbf{x_i})}{\sum_{k=1}^{n} \phi_\rho(\mathbf{x},\mathbf{x_k})}$.

Convex inequality gives $\forall a \in \mathbb{R}$, $1 - e^{-a} < a$ then $\phi_\rho(\mathbf{x}, \mathbf{x_k}) \leq \frac{d((\mathbf{x},\mathbf{x_k}))^2}{\rho^2}$. Further, let $\mathbf{x_{k_1}}, \mathbf{x_{k_2}}$ be two different design points of X_{n_0}, $\forall \mathbf{x} \in \mathbb{X}$, $\max_{i \in \{1,2\}}\{d(\mathbf{x}, \mathbf{x_{k_i}})\} \geq \frac{d(\mathbf{x_{k_1}},\mathbf{x_{k_2}})}{2}$, otherwise the triangular inequality would be violated. Consequently,

$\forall n > n_0$, $\sum\limits_{k=1}^{n} \phi_\rho(\mathbf{x}, \mathbf{x_k}) \geq \phi_\rho(\mathbf{x}, \mathbf{x_{k_1}}) + \phi_\rho(\mathbf{x}, \mathbf{x_{k_2}}) \geq \phi_{2\rho}(\mathbf{x_{k_1}}, \mathbf{x_{k_2}}) > 0$,

$\forall n > n_0$, $\forall \mathbf{x} \in \mathbb{X}$, $w_{i,n}(\mathbf{x}) = \frac{\phi_{i,n}(\mathbf{x})}{\sum\limits_{k=1}^{n} \phi_{k,n}(\mathbf{x})} \leq \frac{\phi_{i,n}(\mathbf{x})}{\phi_{2\rho}(\mathbf{x_{k_1}},\mathbf{x_{k_2}})} \leq \frac{d((\mathbf{x},\mathbf{x_i}))^2}{\rho^2 \phi_{2\rho}(\mathbf{x_{k_1}},\mathbf{x_{k_2}})}$.

Considering $\theta = \frac{1}{\rho^2 \phi_{2\rho}(\mathbf{x_{k_1}},\mathbf{x_{k_2}})}$ ends the proof. ■

Proof of Theorem 5.6. $\mathbb{X}$ is compact so S has a convergent subsequence in $\mathbb{X}^{\mathbb{N}}$ (Bolzano–Weierstrass theorem). Let $(x_{\psi(n)})$ denote that subsequence and $\mathbf{x}_\infty \in \mathbb{X}$ its limit. We can assume by considering a subsequence of ψ and using the continuity of the surrogate model $\hat{s}$ that

- $d(\mathbf{x}_\infty, \mathbf{x}_{\psi(n)}) \leq \frac{1}{n}$ for all $n > 0$,
- $\exists \nu_n \geq d(\mathbf{x}_\infty, \mathbf{x}_{\psi(n)})$ such that $\forall \mathbf{x}' \in \mathbb{X}$, $d(\mathbf{x}', \mathbf{x}_\infty) \leq \nu_n \implies |\hat{s}_{m,-i}(\mathbf{x}_\infty) - \hat{s}_{m,-i}(\mathbf{x}')| \leq \frac{1}{n}$, $\forall i \in 1, \dots, m$, where $m > n_0$.

For all $k > 1$, we note $v_k = \psi(k+1) - 1$, the step at which the UP-EGO algorithm selects the point $\mathbf{x}_{\psi(k+1)}$. So, $\kappa_{v_k}(\mathbf{x}_{\psi(k+1)}) = \max_{\mathbf{x} \in \mathbb{X}}\{\kappa_{v_k}(\mathbf{x})\}$.

Notice first that for all $n > 0$, $\mathbf{x}_{\psi(n)}, \mathbf{x}_{\psi(n+1)} \in \mathcal{B}(\mathbf{x}_\infty, \frac{1}{n})$, where $\mathcal{B}(\mathbf{x}_\infty, \frac{1}{n})$ is the closed ball of center $\mathbf{x}_\infty$ and radius $\frac{1}{n}$. So

$$\text{(i)} \qquad \xi_{v_n}(\mathbf{x}_{\psi(n+1)}) = \delta \underline{d}_{X_{v_n}}(\mathbf{x}_{\psi(n+1)}) \leq \delta d(\mathbf{x}_{\psi(\mathbf{n})}, \mathbf{x}_{\psi(n+1)}) \leq \frac{2\delta}{n}.$$

According to Lemma 5.2, $w_{\psi(n),v_n} \leq \theta \left(d(\mathbf{x}_{\psi(n+1)}, \mathbf{x}_{\psi(n)})\right)^2$ so $w_{\psi(n),v_n} \leq \frac{4\theta}{n^2}$. Consequently,

$$\text{(ii)} \qquad w_{\psi(n),v_n}(\mathbf{x}_{\psi(n+1)}) \max(y_{v_n}^\star - \hat{s}_{v_n,-\psi(n)}(\mathbf{x}_{\psi(n+1)}), 0) \leq \frac{4\theta(U - L)}{n^2}.$$

Further, $\forall i \in 1,\dots,v_n$, $i \neq \psi(n)$, $\hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n)}) = y_{\psi(n)}$ since the surrogate model is an interpolating one, hence, $\hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n)}) \geq y^\star_{v_n}$ and so

$$\max(y^\star_{v_n} - \hat{s}_{v_n,-i}, 0) \leq \max(\hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n)}) - \hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n+1)}), 0) \leq \left|\hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n)}) - \hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n+1)})\right|.$$

The triangular inequality gives

$$\max(y^\star_{v_n} - \hat{s}_{v_n,-i}, 0) \leq \left|\hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n)}) - \hat{s}_{v_n,-i}(\mathbf{x}_\infty)\right| + \left|\hat{s}_{v_n,-i}(\mathbf{x}_\infty) - \hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n+1)})\right|$$

and, finally,

$$\text{(iii)} \qquad \max(y^\star_{v_n} - \hat{s}_{v_n,-i}, 0) \leq \frac{2}{n}.$$

We have

$$\begin{aligned}\left|\kappa_{v_n}(\mathbf{x}_{\psi(n+1)})\right| &= \xi_{v_n}(\mathbf{x}_{\psi(n+1)}) + \sum_{i=1}^{v_n} w_{i,v_n}(\mathbf{x}_{\psi(n+1)}) \max(y^\star_{v_n} - \hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n+1)}), 0)\\ &= \xi_{v_n}(\mathbf{x}_{\psi(n+1)}) + w_{\psi(n),v_n}(\mathbf{x}_{\psi(n+1)}) \max(y^\star_{v_n} - \hat{s}_{v_n,-\psi(n)}(\mathbf{x}_{\psi(n+1)}), 0)\\ &+ \sum_{\substack{i=1\\ i\neq\psi(n)}}^{v_n} w_{i,v_n}(\mathbf{x}_{\psi(n+1)}) \max(y^\star_{v_n} - \hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n+1)}), 0)\\ &\leq \frac{2\delta}{n} + \frac{4\theta(U-L)}{n^2} + \frac{2}{n}.\end{aligned}$$

Considering (i), (ii), and (iii),

$$\left|\kappa_{v_n}(\mathbf{x}_{\psi(n+1)})\right| \leq \frac{2\delta}{n} + \frac{4\theta(U-L)}{n^2} + \frac{2}{n}.$$

Notice that $\kappa_{v_n}(\mathbf{x}_{\psi(n+1)}) = \max_{\mathbf{x}\in\mathbb{X}}\{\kappa_{v_n}(\mathbf{x})\}$ and $\delta \underline{d}_{\mathbf{S}_{v_n}}(\mathbf{x}^\star) = \xi_{v_n}(\mathbf{x}^\star) \leq \kappa_{v_n}(\mathbf{x}^\star) \leq \kappa_{v_n}(\mathbf{x}_{\psi(n)})$. Since $\lim_{n\to\infty}\left|\kappa_{v_n}(\mathbf{x}_{\psi(n+1)})\right| = 0$ then $\lim_{n\to\infty} \underline{d}_{\mathbf{S}_{v_n}}(\mathbf{x}^\star) \to 0$. ■

Appendix A. Optimization test results.

In this section, we use box plots (Figures 11–14) to display the evolution of the best value of the optimization test bench. For each iteration, we display left: EGO in light green; middle: UP-EGO using kriging in light blue; right: UP-EGO using genetic aggregation in dark blue.

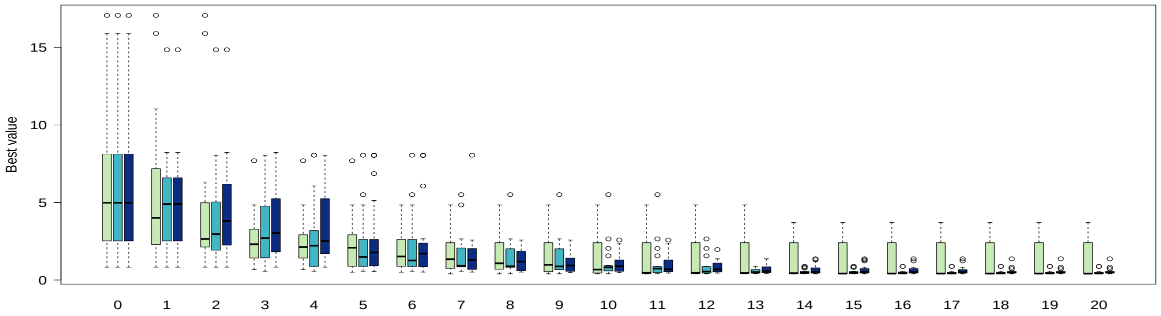


Figure 11. *Branin: Box plots convergence.*

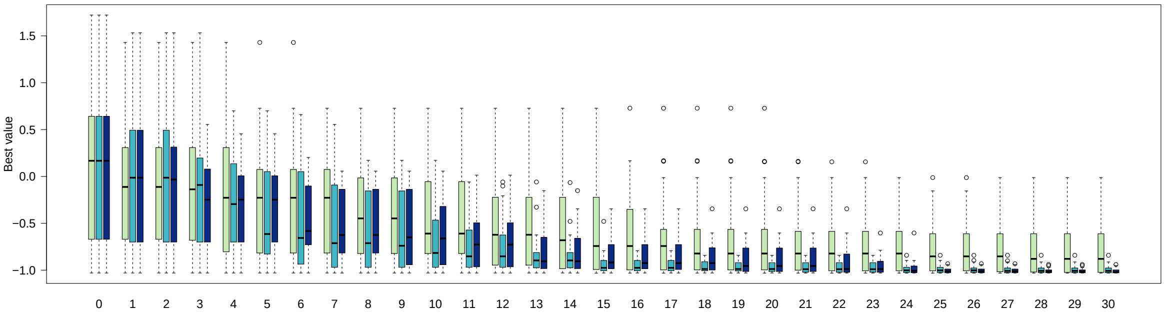


Figure 12. *Six-hump camel: Box plots convergence.*

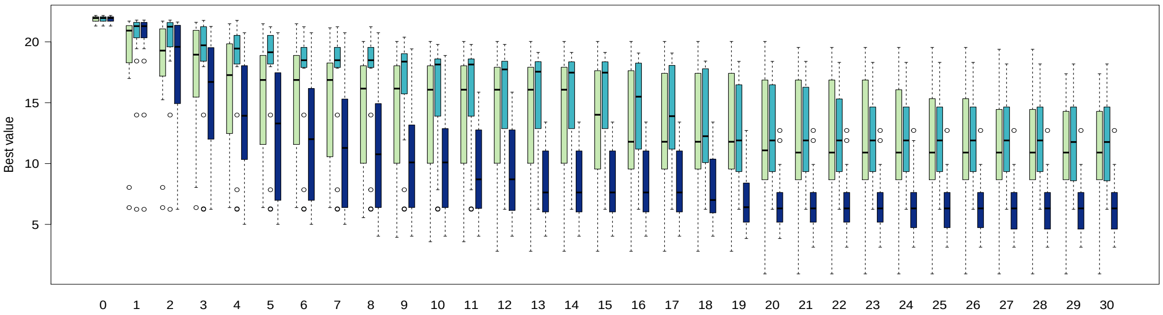


Figure 13. *Ackley: Box plots convergence.*

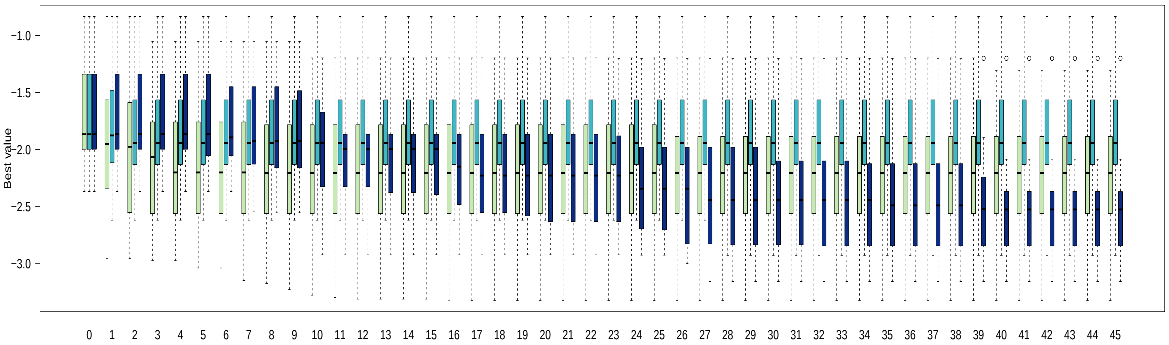


Figure 14. *Hartmann6: Box plots convergence.*

Acknowledgments. We warmly thank Andom Iyassu for his help on editing. We also thank two anonymous reviewers and the associate editor for their valuable comments. Part of this research was presented at the Chair in Applied Mathematics OQUAIDO, gathering partners in technological research (BRGM, CEA, IFPEN, IRSN, Safran, Storengy) and academia (CNRS, Ecole Centrale de Lyon, Mines Saint-Etienne, University of Grenoble, University of Nice, University of Toulouse) around advanced methods for Computer Experiments. We thank the participants for their feedback. An R package containing UP distribution tools is available on https://github.com/malekbs/UP.

REFERENCES

[1] S. Arlot and A. Celisse, *A survey of cross-validation procedures for model selection*, Stat. Surv., 4 (2010), pp. 40–79.

[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.

[3] J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez, *Sequential design of computer experiments for the estimation of a probability of failure*, Stat. Comput., 22 (2012), pp. 773–793.

[4] B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland, *Efficient global reliability analysis for nonlinear implicit performance functions*, AIAA J., 46 (2008), pp. 2459–2468.

[5] G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*, Wiley, New York, 1978.

[6] D. Busby, C. L. Farmer, and A. Iske, *Hierarchical nonlinear approximation for experimental design and statistical data fitting*, SIAM J. Sci. Comput., 29 (2007), pp. 49–69.

[7] C. Chevalier, V. Picheny, and D. Ginsbourger, *Kriginv: An efficient and user-friendly implementation of batch-sequential inversion strategies based on kriging*, Comput. Statist. Data Anal., 71 (2014), pp. 1021–1034.

[8] L. P. Devroye and T. J. Wagner, *Distribution-free performance bounds for potential function rules*, IEEE Trans. Inform. Theory, 25 (1979), pp. 601–604.

[9] L. C. W. Dixon and G. P. Szegö, *Towards Global Optimisation* 2, North-Holland, Amsterdam, 1978.

[10] S. Gazut, J.-M. Martinez, G. Dreyfus, and Y. Oussar, *Towards the optimal design of numerical experiments*, IEEE Trans. Neural. Netw., 19 (2008), pp. 874–882.

[11] S. Geisser, *The predictive sample reuse method with applications*, J. Amer. Statist. Assoc., 70 (1975), pp. 320–328.

[12] A. A. Giunta, S. F. Wojtkiewicz, and M. S. Eldred, *Overview of modern design of experiments methods for computational simulations*, in 41st AIAA Aerospace Sciences Meeting and Exhibit, AIAA, Reston, VA, AIAA-2003-0649, 2003.

[13] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo, *Ensemble of surrogates*, Struct. Multidiscip. Optim., 33 (2007), pp. 199–216.

[14] D. Gorissen, T. Dhaene, and F. De Turck, *Evolutionary model type selection for global surrogate modeling*, J. Mach. Learn. Res., 10 (2009), pp. 2039–2078.

[15] R. Jin, W. Chen, and A. Sudjianto, *On sequential sampling for global metamodeling in engineering design*, in Proceedings of the ASME Design Automation Conference, American Society of Mechanical Engineers, New York, 2002, pp. 539–548.

[16] M. E. Johnson, L. M. Moore, and D. Ylvisaker, *Minimax and maximin distance designs*, J. Statist. Plann. Inference, 26 (1990), pp. 131–148.

[17] 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.

[18] J. P. C. Kleijnen, *Kriging metamodeling in simulation: A review*, European J. Oper. Res., 192 (2009), pp. 707–716.

[19] J. P. C. Kleijnen and W. C. M. Van Beers, *Application-driven sequential designs for simulation experiments: Kriging metamodeling*, J. Oper. Res. Soc., 55 (2004), pp. 876–883.

[20] J. P. C. Kleijnen, W. van Beers, and I. van Nieuwenhuyse, *Expected improvement in efficient global optimization through bootstrapped kriging*, J. Global Optim., 54 (2012), pp. 59–73.

[21] D. G. Krige, *A statistical approach to some basic mine valuation problems on the Witwatersrand*, J. South. African Inst. Min. Metall., 52 (1951), pp. 119–139.

[22] P. Lancaster and K. Salkauskas, *Surfaces generated by moving least squares methods*, Math. Comp., 37 (1981), pp. 141–158.

[23] G. Li and S. Azarm, *Maximum accumulative error sampling strategy for approximation of deterministic engineering simulations*, in Proceedings of the AIAA-ISSMO Multidisciplinary Analysis and Optimization Conference, 2006, AIAA, Reston, VA, AIAA-2006-7051.

[24] G. Li, S. Azarm, A. Farhang-Mehr, and A. R. Diaz, *Approximation of multiresponse deterministic engineering simulations: A dependent metamodeling approach*, Struct. Multidiscip. Optim., 31 (2006), pp. 260–269.

[25] Y. Lin, F. Mistree, J. K. Allen, K. L Tsui, and V. Chen, *Sequential metamodeling in engineering design*, in Proceedings of AIAA-ISSMO Multidisciplinary Analysis and Optimization Conference, Reston, VA, 2004, AIAA-2004-4304.

[26] J. L. Loeppky, J. Sacks, and W. J. Welch, *Choosing the sample size of a computer experiment: A practical guide*, Technometrics, 51 (2009), pp. 366–376.

[27] G. Matheron, *Principles of geostatistics*, Econ. Geol., 58 (1963), pp. 1246–1266.

[28] M. D. McKay, R. J. Beckman, and W. J. Conover, *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.

[29] V. Picheny, D. Ginsbourger, O. Roustant, R. T. Haftka, and N. H. Kim, *Adaptive designs of experiments for accurate approximation of a target region*, AMSE. J. Mech. Des., 132 (2010), 071008.

[30] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, *Surrogate-based analysis and optimization*, Prog. Aerosp. Sci., 41 (2005), pp. 1–28.

[31] P. Ranjan, D. Bingham, and G. Michailidis, *Sequential experiment design for contour estimation from complex computer codes*, Technometrics, 50 (2008), pp. 527–541.

[32] 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. Stat. Softw., 51 (2012), pp. 1–55.

[33] M. J. Sasena, P. Y. Papalambros, and P. Goovaerts, *Metamodeling sampling criteria in a global optimization framework*, in Proceedings of the 8th Symposium Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics, Reston, VA, 2000, AIAA-2000-4921.

[34] T. Shao and S. Krishnamurty, *A clustering-based surrogate model updating approach to simulation-based engineering design*, AMSE. J. Mech. Des., 130 (2008), 041101.

[35] M. C. Shewry and H. P. Wynn, *Maximum entropy sampling*, J. Appl. Stat., 14 (1987), pp. 165–170.

[36] A. J. Smola and B. Schölkopf, *A tutorial on support vector regression*, Stat. Comput., 14 (2004), pp. 199–222.

[37] E. Vazquez and J. Bect, *Convergence properties of the expected improvement algorithm with fixed mean and covariance functions*, J. Statist. Plan. Inference., 140 (2010), pp. 3088–3095.

[38] 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.

[39] F. A. C. Viana, R. T. Haftka, and L. T. Watson, *Efficient global optimization algorithm assisted by multiple surrogate techniques*, J. Global Optim., 56 (2013), pp. 669–689.

[40] G. G. Wang and S. Shan, *Review of metamodeling techniques in support of engineering design optimization*, AMSE. J. Mech. Des., 129 (2007), pp. 370–380.

[41] G. Wahba, *Spline Models for Observational Data*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 59, SIAM, Philadelphia, 2007.

[42] S. Xu, H. Liu, X. Wang, and X. Jiang, *A robust error-pursuing sequential sampling approach for global metamodeling based on Voronoi diagram and cross validation*, AMSE. J. Mech. Des., 136 (2014), 071009.

[43] C. Chevalier, D. Ginsbourger, and X. Emery, *Corrected kriging update formulae for batch-sequential data assimilation*, in Proceedings of the 15th Annual Conference on International Association for Mathematical Geosciences, Springer, Heidelberg, 2014, pp. 119–122.