Adaptive Sampling for Global Meta Modeling Using a Gaussian Process Variance Measure

Johannes Westermann, Antonio Zea, and Uwe D. Hanebeck

***Abstract*— Adaptive sampling methods have been widely used for meta modeling, when dealing with expensive-to-evaluate experiments in practical design and optimization tasks to approximate their performance measure. Existing methods try to minimize the global model error in various ways. However, there is still no general method for the denser sampling of regions with high performance. In this work, we introduce a new adaptive sampling approach that samples regions of high performance more densely, while also exploring unseen regions. A Gaussian process is used as meta model and a variance-based measure is defined for computing the adaptive sample points. Furthermore, Voronoi tessellation is used to reduce the complexity for application in high-dimensional design spaces. The proposed approach allows for higher model accuracy in regions of high performance by efficiently placing the available samples.**

I. Introduction

In many engineering applications [1], [2], performance measures are not given in closed form [3]. Instead, computer experiments (e.g., numerical simulations) or physical experiments are used to obtain the performance subject to a given set of design parameters [4]. These experiments are often expensive or time-consuming, making it impossible to evaluate any number of parameter combinations in design space [5]. Consequently, easy to evaluate meta models are often created based on samples of the real experiments as an approximation.

For example, in the ROBDEKON project [6], expensive and time-consuming sampling of contaminated sites with subsequent laboratory evaluation have to be carried out in order to reconstruct the pollutant distribution in the soil. To ensure that the pollutants can be removed precisely, regions with high concentration values are to be sampled more densely and the meta model is to become more accurate there.

However, the accuracy of the model depends decisively on the choice of the points in the design space [7], [8] to obtain corresponding samples. Design of Experiments (DoE) methods provide points in the design space such that parameter changes are reflected in the variable of interest over the whole design space [9] under consideration of an objective function. In the context of meta modeling, the objective function usually requires maximizing the global model accuracy using the least number of samples.

*This work is supported by the German Federal Ministry of Education and Research (BMBF) within the ROBDEKON project.

The authors are with the Intelligent Sensor-Actuator-Systems (ISAS) laboratory at the Karlsruhe Institute of Technology (KIT), Germany westermann@kit.edu, antonio.zea@kit.edu, uwe.hanebeck@ieee.org.

One-shot approaches like fractional designs, latin hypercubes, and orthogonal arrays [10], [11] try to achieve this goal by evenly covering the design space [11]. The advantage of these methods is their simplicity. Since the points are predefined, too many or too few samples might be taken [12] and it is not clear what needs to be done if the desired model accuracy after sampling is not sufficient [9].

In many cases, one-shot methods are used to create a few points as a basis for more sophisticated sequential design approaches [9], [11], [13], which determine one or more next points per iteration based on the already existing samples. Thus, they also allow for a termination criterion (e.g., reaching a desired estimated model accuracy) and hence, a variable number of samples adapted to the specific setting.

Meta models based on Gaussian processes are among the most frequently used [1]. For example, in [9], an information maximization criterion based on the covariance matrix of a Gaussian process is used to select the next points in the design space. A similar approach is followed in [1] by selecting the next point where the prediction variance of the Gaussian process is maximized.

All adaptive sampling methods (also called hybrid sequential designs [11]) need to perform a trade-off between exploration and exploitation [10], [12], where exploitation increases the local model accuracy and exploration ensures that no relevant regions of the design space are omitted.

A popular way to choose the next point is cross-validation, where the meta model is constructed on a subset of the gathered samples followed by the evaluation of the model on the remaining points to obtain a measure of the corresponding model errors. In the CV-Voronoi approach [12], leave-one-out cross-validation is applied to existing samples where the sample furthest from the model is used for choosing the next sample. A Voronoi tessellation is constructed on all existing samples and the next sample is sequentially chosen as the furthest corner of the associated Voronoi cell.

The LOLA-Voronoi algorithm [11] estimates the local nonlinearity of the unknown function at each sample point. Just like in the CV-Voronoi method, the next sample point is then chosen to be the furthest corner of the Voronoi cell corresponding to the largest estimated nonlinearity. The advantage of this method is its independence from the applied meta model. However, it has been shown that CV-Voronoi outperforms LOLA-Voronoi [12].

Existing methods are either space-filling [10] or try to minimize the global model error in different ways [9], [11], [12], [14]. However, in parameter optimization tasks with a positive performance measure that is to be maximized, as

well as in the earlier mentioned decontamination scenario, more samples in regions of large function values are desired. Consequently, a better resolution in relevant regions can be achieved with the same number of samples.

This paper proposes a sampling algorithm for meta modeling of positive continuous functions that generates more samples in regions of large function values than in regions of low function values, while executing the trade-off between exploration (space-filling) and exploitation (sampling at high function values), in a deterministic and adaptive manner. This leads to a higher resolution and thus to an improved robustness of the model in these regions. As a result, the optima of the underlying function are modeled accurately and serve for optimization as well as a replacement of the experiment. Two versions of the algorithm are presented. The first version computes an optimal next point in the continuous design space. To reduce the search space, the second version uses Voronoi tessellations to obtain a finite number of discrete points in the design space to optimally choose the next point from.

The paper is structured as follows: Sec. II provides a formal description of the considered problem. The key idea to solve the problem is presented in Sec. III. Sec. IV introduces our adaptive sampling approach, which is adjusted for reduced computational complexity based on Voronoi tessellation in Sec. V. Finally, we present our evaluation in Sec. VI and conclusions in Sec. VII.

II. Problem Formulation

An unknown continuous function $f : \mathbb{R}^n \to \mathbb{R}_0^+$ is to be approximated within the bounded set $\mathcal{X} \subset \mathbb{R}^n$ based on noisy samples $y_i \in \mathbb{R}$ taken at points $\underline{x}_i \in \mathcal{X}$ for $1 \leq i \leq m$, where m is the number of existing samples, n is the number of design parameters, and $\mathcal{X}$ is the design space. The samples

$$y_i = f(\underline{x}_i) + v_i \tag{1}$$

are given by the value of the true function $f(\underline{x})$ at points $\underline{x}_i$ disturbed by additive, zero-mean noise approximated by a Gaussian noise term $v_i \sim \mathcal{N}(0, \sigma_i^2)$. Furthermore, the approximation should be "more accurate" (according to a given metric) where the true function $f(\underline{x})$ takes large function values and should hence be sampled more densely in the corresponding regions of the design space. To allow for an adaptive sampling approach, the samples y_i are taken sequentially. Given the existing samples $(y_1, \underline{x}_1), \ldots, (y_m, \underline{x}_m)$, the task is to find the optimal choice of $\underline{x}_{m+1}$.

III. Methodology

Our approach to solving the problem stated in Sec. II is to construct a meta model $\hat{f}(\underline{x})$ using the existing samples $(y_i, \underline{x}_i),\ i = 1, ..., m$. As a meta model we choose a Gaussian process (GP) model [15] as it is well suited for approximation of continuous functions, can be trained on little data, and the model prediction variance can be computed easily. A GP is a non-parametric regression approach that can be considered a multivariate Gaussian distribution over functions. The joint distribution

$$\begin{bmatrix} \underline{y} \\ \underline{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\Sigma} & \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_*) \\ \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) \end{bmatrix}\right) \tag{2}$$

of the samples $\underline{y} = [y_1, \ldots, y_m]^\top$ and the distribution of the predictive model function values $\underline{f}_*$ using the kernel (or covariance) function $k(\cdot, \cdot)$ can be conditioned as follows to obtain the predictive distribution

$$\underline{f}_* | \boldsymbol{X}, \underline{y}, \boldsymbol{X}_* \sim \mathcal{N}\big(\hat{\underline{f}}_*, \mathrm{cov}(\underline{f}_*)\big)\ , \tag{3}$$

where $\boldsymbol{X} = [\underline{x}_1, \ldots, \underline{x}_m]^\top$ are the training inputs, $\boldsymbol{X}_*$ are the test points,

$$\hat{\underline{f}}_* = \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X})\big(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\Sigma}\big)^{-1} \underline{y} \tag{4}$$

is the predictive mean,

$$\begin{aligned} \mathrm{cov}(\underline{f}_*) =& \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) \\ & - \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X})\big(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\Sigma}\big)^{-1} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_*) \end{aligned} \tag{5}$$

is the predictive covariance, $\boldsymbol{K}(\cdot, \cdot)$ is a batch evaluation of the kernel function and $\boldsymbol{\Sigma} = \mathrm{diag}\big([\sigma_1^2, \ldots, \sigma_m^2]^\top\big)$ contains the sample noise variances. As a kernel function, we use the squared exponential kernel

$$k(\underline{a}, \underline{b}) = \beta^2 \exp\left(-\frac{1}{2} \underline{a}^\top \boldsymbol{L}^{-1} \underline{b}\right) \tag{6}$$

with hyperparameters β and $\underline{l}$, where $\boldsymbol{L} = \mathrm{diag}(\underline{l})$.

In order to obtain the next point $\underline{x}_{m+1}$ in the design space, we need to first define an objective function for it. The most naïve approach is to select the point that maximizes the prediction variance $\sigma^2_{\text{pred}}(\underline{x})$ of the GP as the next point. In this way, the global model uncertainty is minimized. However, this does not guarantee the model error to be smaller in regions of large function values of $f(\underline{x})$ than in regions of small values. To achieve this goal, we weight the prediction variance with the function value of $f(\underline{x})$, leading to

$$J(\underline{x}) = f(\underline{x}) \cdot \sigma^2_{\text{pred}}(\underline{x}) \tag{7}$$

as the objective function. Since $f(\underline{x})$ takes non-negative values, (7) becomes maximal in regions of the design space where the model uncertainty is large and $f(\underline{x})$ takes large values. Thus, the prediction variance provides exploration and prevents sample clustering. On the other hand, weighting the prediction variance with $f(\underline{x})$ ensures more samples in regions of large function values (exploitation). But the formulation of the objective in (7) poses the following challenges:

- C1 The evaluation of (7) requires the true function $f(\underline{x})$ to be known on the entire design space. Since this is not possible, we use the meta model $\hat{f}(\underline{x}) \approx f(\underline{x})$ instead, which approximates the true function.
- C2 In unexplored regions of the design space the model values $\hat{f}(\underline{x}) \approx 0$ vanish. Hence, those regions are not getting explored, even though the true function might take significant values there.
- C3 $J(\underline{x})$ generally exhibits local maxima, which prohibits global optimization, especially in case of a high-dimensional design space.

IV. Weighted Maximum Variance Adaptive Sampling Approach

In this section, we propose our novel weighted maximum variance adaptive sampling (WMVAS) approach that in each iteration computes the optimal next point in the design space for evaluation of the true function based on a weighted variance measure similar to (7), given the samples of the former iteration. For its derivation, we need to gradually tackle the challenges C1 – C3 in Sec. III.

As mentioned in C1, the true function is unknown and can therefore not be evaluated in the objective function. Hence, we need to approximate the true function $f(\underline{x})$ in (7) by constructing a meta model using the already existing samples. Consequently, we select

$$\underline{x}_{m+1} = \arg\max_{\underline{x}} \Big(\hat{f}(\underline{x}) \cdot \sigma^2_{\text{pred}}(\underline{x}) \Big) \tag{8}$$

as the next point in the design space for evaluation of the true function f. The model $\hat{f}$ is trained on the existing samples in each iteration of the algorithm.

In unseen regions, the model predicts values according to the prior distribution of the GP, which are set to zero according to (2). Corresponding to challenge C2, this causes a vanishing objective function and therefore hinders exploration in the associated regions.

In order to ensure exploration of unseen regions, we clamp the model function value to the lower bound $\alpha \cdot \hat{f}_{\max}$, where $\alpha \in [0,1]$ is the *exploration parameter* and $\hat{f}_{\max} := \max_{\underline{x}} \{\hat{f}(\underline{x})\}$ is the largest value taken by the model function within the design space. This results in weighting the prediction variance of the GP model with the lower bound in regions where the mean of the GP model falls below the lower bound. Although low sample values can cause the model value to fall below the lower bound as well, these regions are not getting further explored, since the GP variance is not clamped. Taking the clamping of the model function value to the lower bound into account, the next point in the design space for sampling the true function is selected by

$$\underline{x}_{m+1} = \arg\max_{\underline{x}} J(\underline{x}) \ , \tag{9}$$

$$\text{where} \quad J(\underline{x}) = \hat{f}_{\text{clamp}}(\underline{x}) \cdot \sigma^2_{\text{pred}}(\underline{x}) \ , \tag{10}$$

$$\text{and} \quad \hat{f}_{\text{clamp}}(\underline{x}) = \max\big(\alpha \cdot \hat{f}_{\max}, \hat{f}(\underline{x})\big). \tag{11}$$

The algorithm is executed until a termination criterion is met, which depends on the concrete requirements of each particular scenario. Besides limiting the number of iterations, a common criterion is cross-validation (CV), where each measurement y_i is left out once, the meta model $\hat{f}_i^{\text{CV}}(\underline{x}_i)$ is constructed on the remaining samples, and the error between the left out sample and the meta model at the point of the left out sample is calculated (see e.g., [12]). Since our method emphasizes large function values, we introduce the weighted root mean square error for CV $e^{\text{cv}}_{\text{wrms}}$ that weighs the errors calculated during CV by adding them up. Relative function values are used as weights in order to reflect the lower required accuracy at lower function values. We propose the termination condition

$$\delta(e^{\text{cv}}_{\text{wrms}}) = \begin{cases} \text{true}, & e^{\text{cv}}_{\text{wrms}} < \epsilon \\ \text{false}, & \text{otherwise} \end{cases} \tag{12}$$

that outputs true if $e^{\text{cv}}_{\text{wrms}}$ is smaller than some $\epsilon > 0$ and false otherwise. The weighted root mean square error for CV

$$e^{\text{cv}}_{\text{wrms}} = \sqrt{\frac{\sum_{i=1}^{m} |y_i| \cdot \big(y_i - \hat{f}_i^{\text{CV}}(\underline{x}_i)\big)^2}{\sum_{i=1}^{m} |y_i|}} \tag{13}$$

can be computed by the CV error according to each sample and the largest sample value $y_{\max} = \max_i(y_i)$ in the sample set. The pseudo code for the WMVAS approach can be found in Algorithm 1.

Algorithm 1 WMVAS

1: Take M_0 initial samples $(y_i, \underline{x}_i),\ i = 1, ..., M_0$ according to a space-filling sampling scheme
2: Set $\alpha \in [0,1]$, $m = M_0$, $\epsilon > 0$
3: **while** Termination criterion not met **do**
4: Train meta model $\hat{f}(\underline{x})$ on $(y_i, \underline{x}_i),\ i = 1, ..., m$
5: $\underline{x}_{m+1} \leftarrow \arg\max_{\underline{x}} \big(\hat{f}_{\text{clamp}}(\underline{x}) \cdot \sigma^2_{\text{pred}}(\underline{x})\big)$
6: Obtain y_{m+1} by evaluating the true function $f(\underline{x})$ using parameters $\underline{x}_{m+1}$
7: $m \leftarrow m + 1$
8: **end while**

In each iteration, the proposed WMVAS algorithm needs to perform global optimization on the design space to select the optimal next parameter combination. This becomes computationally complex or even intractable (challenge C3) for high-dimensional design spaces. A solution of this problem is presented in Sec. V.

V. Voronoi-based Weighted Maximum Variance Adaptive Sampling Approach

The reason for the use of a meta model is its fast evaluation. This property also allows several evaluations of the meta model to be carried out per iteration of the algorithm in order to determine the optimal parameter combination for evaluating the true function. Nevertheless, global optimization of the meta model in (9) for high-dimensional, continuous design spaces also becomes challenging. The maximization in (9) can be simplified if the objective function in (10) is evaluated at discrete points of the design space only. The best next point of the N candidates $\mathcal{X}^{\text{cand}} \subset \mathcal{X}$ is then obtained according to

$$\underline{x}_{m+1} = \arg\max_{\underline{x}_j^{\text{cand}}} J(\underline{x}_j^{\text{cand}}) \tag{14}$$

with $\mathcal{X}^{\text{cand}} = \{\underline{x}_1^{\text{cand}}, \ldots, \underline{x}_j^{\text{cand}}, \ldots \underline{x}_N^{\text{cand}}\}$ by computing the maximum of N values using $J(\underline{x})$ from (10). Selecting these candidate points is is crucial for discretization of the design space, as they should be close to the optimal point.

By computing the Voronoi tessellation [16], [17] of $\mathcal{X}$ around the m existing samples, the design space is divided into cells. Each cell is assigned to one sample point containing all points in design space that are closer to this

sample than to all the other samples (Fig. 1). Consequently, the edges of the cells are constructed of all the midpoints of the neighboring samples. The corners of the cells are the mid points of at least $n+1$ samples, when n is the dimension of the design space.

Since the prediction variance $\sigma^2_{\text{pred}}(\underline{x})$ of the meta model is the basis of the objective function and increases with distance to the samples, we choose the corners of the Voronoi tessellation as candidates for the optimal point. In addition, we limit the Voronoi tessellation to the boundaries of the design space and add the resulting corners to the set of candidate points (see Fig. 1). Finally, in each iteration of the algorithm, the corner leading to the largest value of $J(\underline{x})$ is selected as the next point for sampling. The pseudo code for the Voronoi-based weighted maximum variance adaptive sampling (WMVAS-Voronoi) approach is provided in Algorithm 2.

Algorithm 2 WMVAS-Voronoi

1: Take M_0 initial samples $(y_i, \underline{x}_i),\ i = 1, ..., M_0$ according to a space-filling sampling scheme
2: Set $\alpha \in [0, 1]$, $m = M_0$, $\epsilon > 0$
3: **while** Termination criterion not met **do**
4: Train meta model $\hat{f}(\underline{x})$ on $(y_i, \underline{x}_i),\ i = 1, ..., m$
5: Construct voronoi tessellation on $\underline{x}_i,\ i = 1, ..., m$
6: Limit voronoi tessellation to the design space
7: $\mathcal{X}^{\text{cand}} \leftarrow$ corners of resulting voronoi cells
8: $\underline{x}_{m+1} \leftarrow \arg\max_{\underline{x}_j^{\text{cand}}} \left(\hat{f}_{\text{clamp}}(\underline{x}_j^{\text{cand}}) \cdot \sigma^2_{\text{pred}}(\underline{x}_j^{\text{cand}})\right)$
9: Obtain y_{m+1} by evaluating the true function $f(\underline{x})$ using parameters $\underline{x}_{m+1}$
10: $m \leftarrow m + 1$
11: **end while**

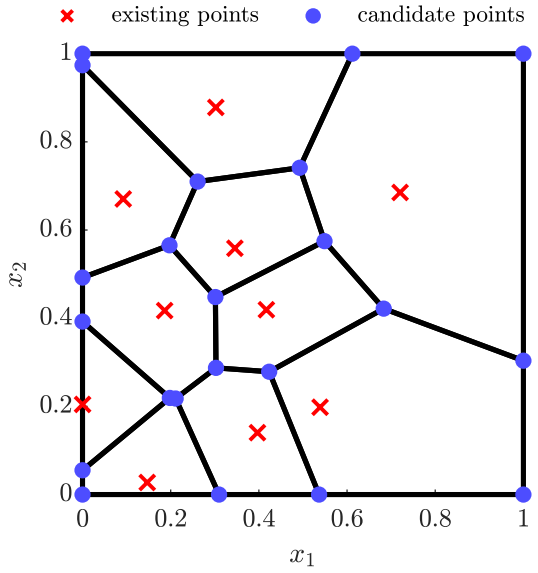


Fig. 1: Voronoi tessellation of existing sample points $\underline{x}_i,\quad i = 1, \ldots, m$ (red crosses) and candidate points $\underline{x}_j^{\text{cand}},\quad j = 1, \ldots, N$ (blue dots) in a 2D design space. The candidate points are located at the corners of the Voronoi cells.

VI. Evaluation

In this section, we evaluate our proposed adaptive sampling algorithm WMVAS (Sec. IV) and its Voronoi-based version WMVAS-Voronoi (Sec. V). The proposed algorithms are compared against regular grid sampling (RGS) and the CV-Voronoi approach [12].

A. Experiment Setup

The function $f : \mathcal{X} \to \mathbb{R}_0^+$ is to be approximated on the two-dimensional design space $\mathcal{X} = [0, 1] \times [0, 1] \subset \mathbb{R}^2$. For this, we model $f(\underline{x})$ as the weighted sum

$$f(\underline{x}) = \textstyle\sum_{i=1}^{5} c_i h_i(\underline{x}) \tag{15}$$

of five squared exponential basis functions

$$h_i(\underline{x}) = \exp\Big(-\frac{1}{2}\big(\underline{x} - \underline{\mu}_i\big)^\top \boldsymbol{S}_i^{-1}\big(\underline{x} - \underline{\mu}_i\big)\Big)\ , \tag{16}$$

where $c_i \in [0, 1]$ are the weights, $\underline{\mu}_i \in \mathcal{X}$ defines the location of the basis function in design space, and $\boldsymbol{S}_i$ is a covariance matrix. The covariance matrix is constructed by choosing on-diagonal elements within $[0.003, 0.05]$, followed by computing the off-diagonal elements dependent on a correlation coefficient $\rho \in [-0.3, 0.3]$. For example, $f(\underline{x})$ can be considered a function of concentration, where we are interested in regions of high concentration values (e.g., to decontaminate pollutants or mining of valuable material).

As described in Sec. III, a GP is used as a meta model for the approximation of $f(\underline{x})$. The change of $f(\underline{x})$ in this concrete case is similar in both considered dimensions of the design space, so that the corresponding hyperparameters are equated. Accordingly, the matrix $\boldsymbol{L}$ in (6) is reduced to the scalar value l. In order to compare the performance of the sampling algorithms, both the meta model $\hat{f}(\underline{x})$ and the true function $f(\underline{x})$ are evaluated on a regular grid of $N^{\text{eval}} = 10000$ evaluation points. Based on these evaluations, the root mean square error

$$\text{RMSE} := \sqrt{\frac{\sum_{i=1}^{N^{\text{eval}}} \big(\hat{f}(\underline{x}_i) - f(\underline{x}_i)\big)^2}{N^{\text{eval}}}} \tag{17}$$

and the weighted root mean square error

$$\text{WRMSE} := \sqrt{\frac{\sum_{i=1}^{N^{\text{eval}}} f(\underline{x}_i) \cdot \big(\hat{f}(\underline{x}_i) - f(\underline{x}_i)\big)^2}{\sum_{i=1}^{N^{\text{eval}}} f(\underline{x}_i)}} \tag{18}$$

are computed. While the RMSE indicates the mean global error, the WRMSE weights deviations where high function values occur more strongly and is therefore suitable for testing our proposed algorithms. In the performed simulations, the exploration parameter is set to $\alpha = 0.010$ for WMVAS and to $\alpha = 0.025$ for WMVAS-Voronoi. In addition, no sampling noise is applied, which translates to $y_i \equiv f(\underline{x}_i)$.

B. Results

First, a true function $f^{\text{eval}}(\underline{x})$ is created by randomly choosing function parameters according to their specification in Sec. VI-A. An illustration of the function can be found in Fig. 2. Then, each algorithm samples the function to obtain 49 samples. The adaptive sampling algorithms are initialized

with 16 grid samples and then perform 33 adaptive sampling iterations, whereas the grid samples are pre-calculated. In Fig. 3 the results of the simulation are displayed. Both the WMVAS and the WMVAS-Voronoi algorithm place the adaptive samples in regions of high functional values. Thus, the corresponding regions can be better approximated than with the comparison methods. If further samples are taken, the prediction variance of the meta model in regions of high function values become close to zero, and consequently, regions of low function values will also get explored due to the lower bound based on the exploration parameter in (9) – (11).

In order to evaluate the methods quantitatively, 100 runs are performed on the true function $f^{\text{eval}}(\underline{x})$ in Fig. 2 with 16 random initial points per run. The same 16 initial points are used for all adaptive algorithms in a run. In addition, another 100 runs of the algorithms were performed, where a random function was generated in each run according to the specifications in Sec. VI-A. The 16 initial points were distributed on a 4 by 4 grid, indicated by the red dots in Fig. 3.

For each run, algorithm and iteration, the errors between the meta model and the true function are calculated according to (17) and (18). These errors are then averaged over the runs separately for the simulations with $f^{\text{eval}}(\underline{x})$ from Fig. 2 and the simulations with different random functions. The error graphs are shown in Fig. 4.

With 100 runs of the algorithms on the function in Fig. 2, the adaptive sampling methods initially have similar RMSE and WRMSE. As the number of samples increases, the error variance of the WMVAS and WMVAS-Voronoi methods decreases significantly and their errors converge. At the beginning, the errors of the method CV-Voronoi show a similar course as those of WMVAS-Voronoi. From about 45 samples on, the error of method CV-Voronoi is then higher than the errors of the other methods and the variance is also larger than that of the other adaptive methods. RGS guarantees a better coverage of the design space than random initial points and consequently has a smaller RMSE and WRMSE at first. From 36 samples, the RMSE of RGS follows that of WMVAS, being slightly higher. However, the WMVAS and WMVAS-Voronoi methods achieve a good approximation of the regions of high function values starting from 60 samples, as can be seen from the small WRMSE and small corresponding variance. In particular, the WMVAS and WMVAS-Voronoi methods exhibit a significantly lower WRMSE than CV-Voronoi and RGS from about 50 samples.

When the algorithms are applied to 100 randomly generated functions, the RMSE and associated variances are similar for the proposed methods. The RMSE of RGS follows their course, always being slightly higher. CV-Voronoi initially exhibits the same RMSE as RGS. In the range from about 40 to 80 samples, the RMSE of CV-Voronoi is significantly lower than that of the other methods. The WRMSE shows that the regions of high function values are best approximated by the WMVAS and WMVAS-Voronoi methods. RGS, on the other hand, has the largest WRMSE. CV-Voronoi shows the same behavior as RGS up to 25 samples, then falls below and reaches the WRMSE of WMVAS-Voronoi in the range of 50 to 60 samples. After that, it is significantly above the WRMSE of WMVAS and WMVAS-Voronoi and approaches the error of RGS.

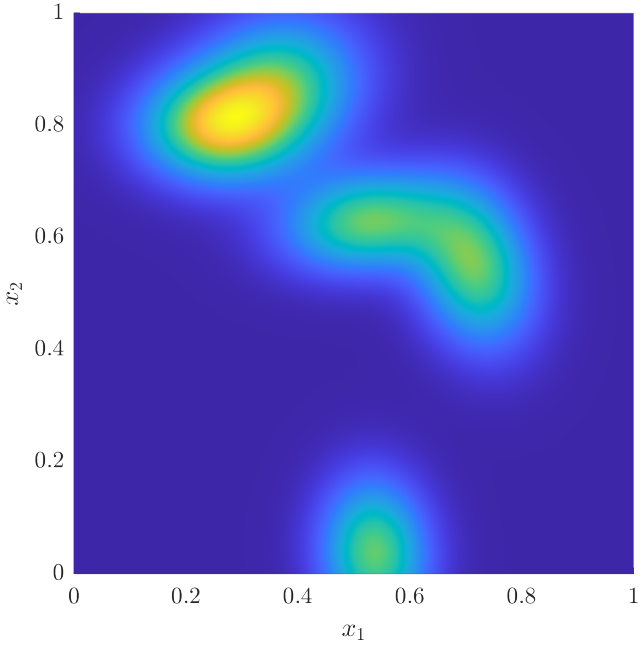


Fig. 2: Function $f^{\text{eval}}(\underline{x})$ evaluated on the design space. Low function values are represented in blue and high function values in yellow.

VII. Conclusion

This paper introduced an adaptive sampling approach for meta modeling of expensive-to-evaluate functions and real world experiments, with a positive performance measure. Furthermore, a Voronoi-based version of the algorithm with reduced computational complexity for high-dimensional scenarios was presented. A Gaussian process was used as a meta model and to formulate the optimization problem to determine the adaptive sample points. The performance of the proposed algorithms was illustrated. It was shown that our proposed approaches lead to better approximation results than grid sampling and the adaptive sampling approach cross-validation-Voronoi where the function of interest takes large values. Future work includes alternative objective functions for computing the adaptive sample points, to allow the applicability to a wider class of functions. In the ROBDEKON context, real robots must be moved to new sample points and additional costs (e.g., the way to go) for choosing the sample points will also be considered.

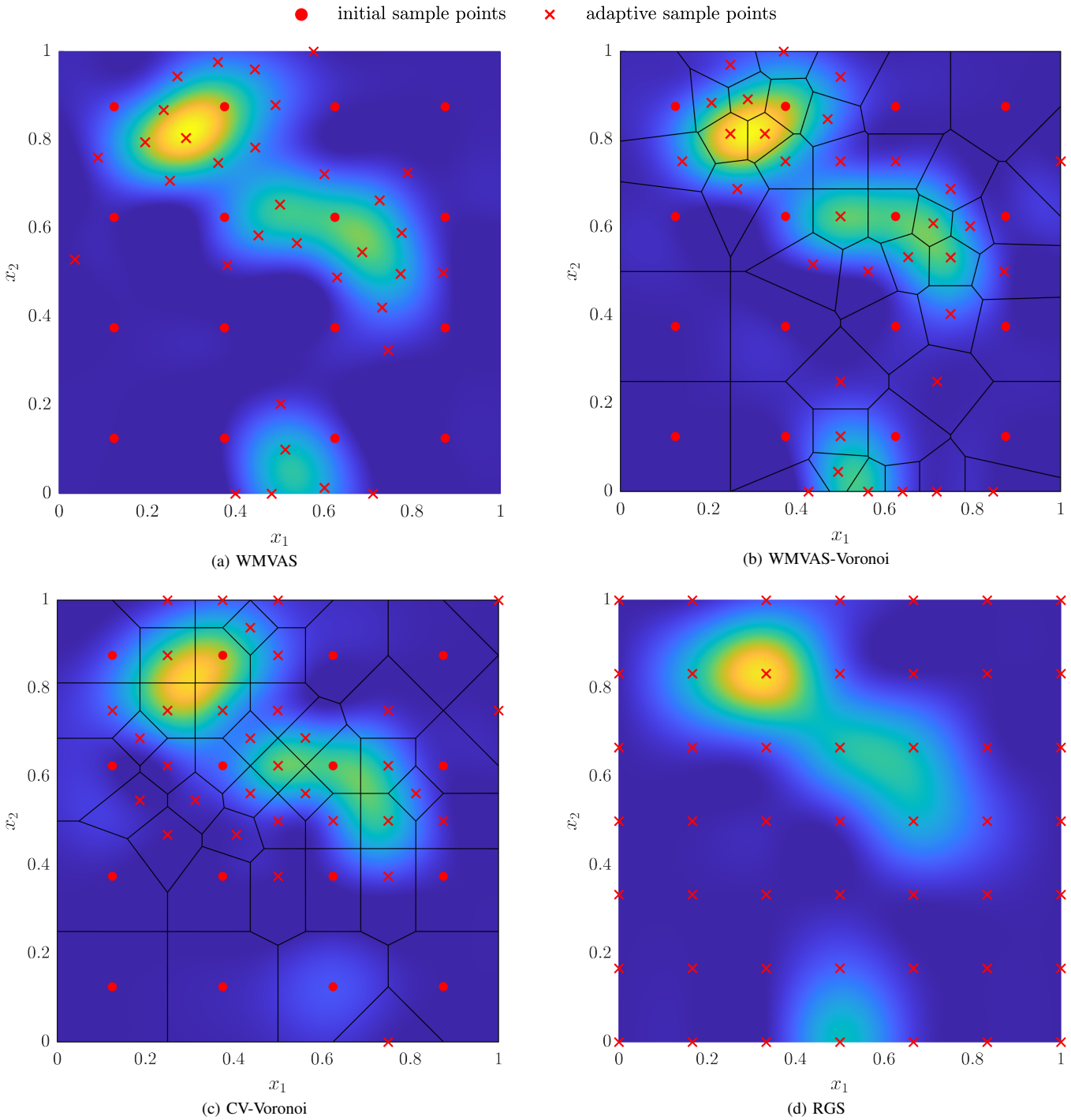
References

[1] U. Choi, J. Kim, and J. Ahn, "Cost-Aware Adaptive Design of Experiment with Nonstationary Surrogate Model for Wind Tunnel Testing," *International Journal of Aeronautical and Space Sciences*, vol. 21, no. 3, pp. 670–680, Sep. 2020.

[2] T. Braconnier, M. Ferrier *et al.*, "Towards an adaptive POD/SVD surrogate model for aeronautic design," *Computers & Fluids*, vol. 40, no. 1, pp. 195–209, Jan. 2011.

[3] J. Branke, K. Deb *et al.*, Eds., *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin, Heidelberg: Springer-Verlag, 2008.

[4] T. Simpson, J. Poplinski *et al.*, "Metamodels for Computer-based Engineering Design: Survey and recommendations," *Engineering with Computers*, vol. 17, no. 2, pp. 129–150, Jul. 2001.



(a) WMVAS (b) WMVAS-Voronoi

(c) CV-Voronoi (d) RGS

Fig. 3: Visualization of the meta model of the examined algorithms, evaluated on the two-dimensional design space. The space-filling initial points (red dots) and the adaptive points (red crosses) are shown on top. For the Voronoi-based approaches in Fig. 3b and 3c, the black lines represent the edges of the voronoi cells. The top row shows the proposed methods.

[5] H. Liu, Y.-S. Ong, and J. Cai, "A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design," *Structural and Multidisciplinary Optimization*, vol. 57, no. 1, pp. 393–416, Jan. 2018.

[6] J. Petereit, J. Beyerer *et al.*, "ROBDEKON: Robotic Systems for Decontamination in Hazardous Environments," in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Sep. 2019, pp. 249–255.

[7] A. L. Kaminsky, Y. Wang, and K. Pant, "An Efficient Batch K-Fold Cross-Validation Voronoi Adaptive Sampling Technique for Global Surrogate Modeling," *Journal of Mechanical Design*, pp. 1–14, May 2020.

[8] R. Jin, W. Chen, and A. Sudjianto, "On Sequential Sampling for Global Metamodeling in Engineering Design," in *Volume 2: 28th Design Automation Conference*. Montreal, Quebec, Canada: ASMEDC, Jan. 2002, pp. 539–548.

[9] H.-L. Choi, J. Ahn, and D.-H. Cho, "Information-maximizing adaptive design of experiments for wind tunnel testing," in *Engineering Optimization 2014*, A. Araujo, Ed. CRC Press, Sep. 2014, pp. 329–334.

[10] K. Crombecq, E. Laermans, and T. Dhaene, "Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling," *European Journal of Operational Research*, vol. 214, no. 3,

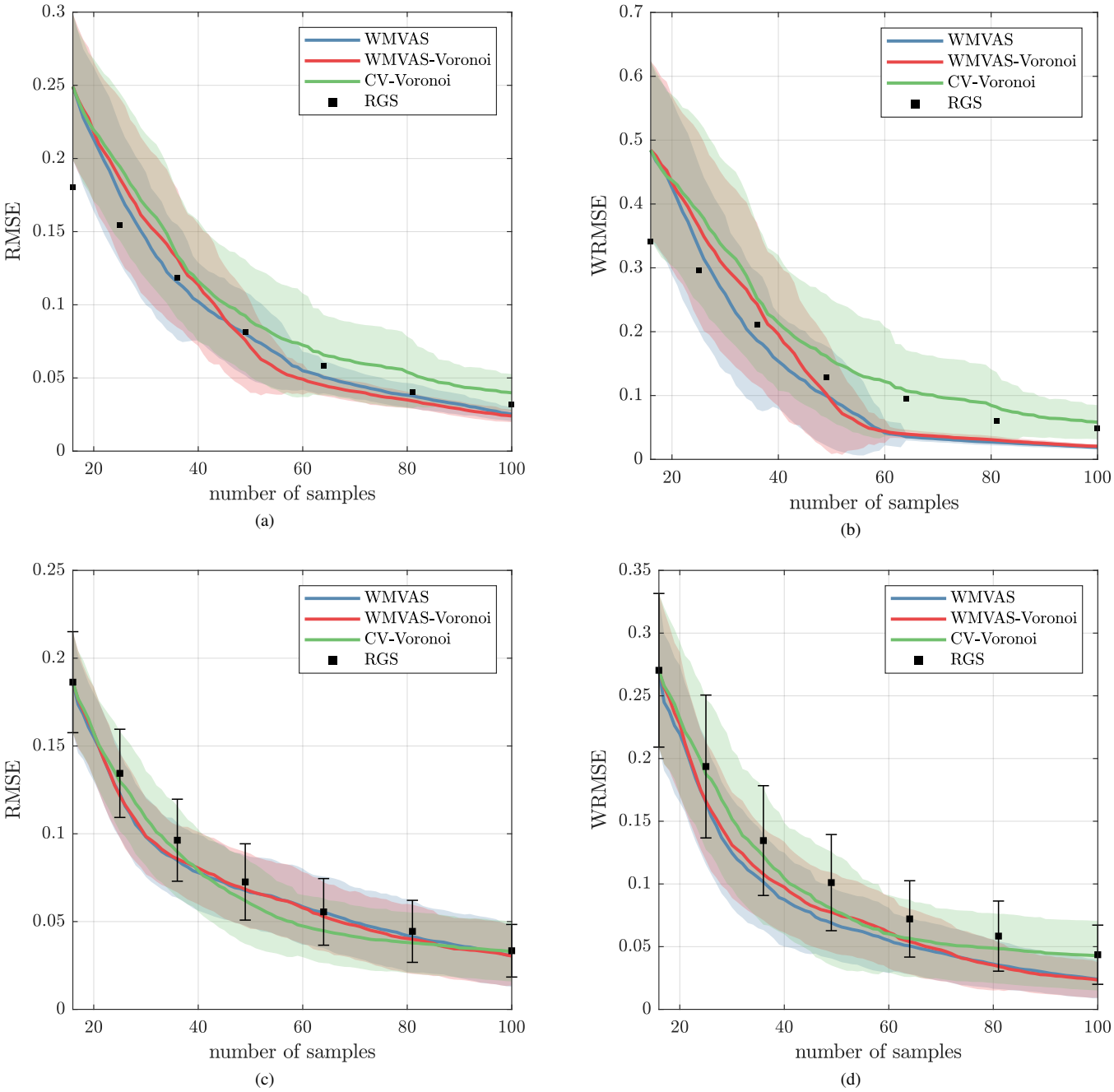


Fig. 4: Mean (solid lines, squares) and standard deviation (shaded areas, error bars) of the error measures (RMSE, WRMSE) vs. the number of samples. In 4a and 4b 100 runs were performed on the distribution in Fig. 2, where the algorithms were initialized with 16 random samples in each run. For 4c and 4d the algorithms were applied to 100 different distributions (one run per distribution). The same 16 initial grid points were used for each run.

pp. 683–696, Nov. 2011.

[11] K. Crombecq, D. Gorissen *et al.*, "A Novel Hybrid Sequential Design Strategy for Global Surrogate Modeling of Computer Experiments," *SIAM Journal on Scientific Computing*, vol. 33, no. 4, pp. 1948–1974, Jan. 2011.

[12] S. Xu, H. Liu *et al.*, "A Robust Error-Pursuing Sequential Sampling Approach for Global Metamodeling Based on Voronoi Diagram and Cross Validation," *Journal of Mechanical Design*, vol. 136, no. 7, p. 071009, Jul. 2014.

[13] A. Golzari, M. Haghighat Sefat, and S. Jamshidi, "Development of an adaptive surrogate model for production optimization," *Journal of Petroleum Science and Engineering*, vol. 133, pp. 677–688, Sep. 2015.

[14] H. Tong, C. Huang *et al.*, "Voronoi-based Efficient Surrogate-assisted Evolutionary Algorithm for Very Expensive Problems," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, Jun. 2019, pp. 1996–2003.

[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, 3rd ed., ser. Adaptive Computation and Machine Learning. Cambridge, Mass.: MIT Press, 2008.

[16] J.-R. Sack, Ed., *Handbook of Computational Geometry*, 1st ed. Amsterdam: Elsevier, 2000.

[17] M. L. Gavrilova, C. J. K. Tan *et al.*, Eds., *Transactions on Computational Science XX*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 8110.