Hindawi
International Journal of Aerospace Engineering
Volume 2019, Article ID 4937261, 34 pages
https://doi.org/10.1155/2019/4937261

Research Article

Efficient Design Optimization Assisted by Sequential Surrogate Models

Emiliano Iuliano

CIRA, The Italian Aerospace Research Center, Via Maiorise, 81043 Capua, Italy

Correspondence should be addressed to Emiliano Iuliano; e.iuliano@cira.it

Received 17 July 2018; Accepted 12 March 2019; Published 12 May 2019

Academic Editor: Hikmat Asadov

The paper proposes a global optimization algorithm employing surrogate modeling and adaptive infill criteria. The surrogates are exploited to screen the design space and provide lower-fidelity predictions across it; on the other hand, specific criteria are designed to suggest new points for high-fidelity evaluation so as to enrich the optimizer database. Both Kriging and radial basis function network are used as surrogates with different training strategies. Sequential design is achieved by introducing several infill criteria according to the realization of the exploration-exploitation trade-off. Optimization results are provided both for scalable and analytical test functions and for a practical aerodynamic shape optimization problem.

1. Introduction

The trade-off between high fidelity and short response time is an essential part of today's real-world engineering design applications. On the one hand, the industrial request to shift the focus and part of the costs from experimental to numerical design and analysis leads to the introduction of more and more physics modeling into numerical simulation codes. On the other hand, the price to pay is related to increasing computational time of single analyses and design cycles which is detrimental for the purpose of reducing the time to market. This trade-off is even more evident when computational fluid dynamics (CFD) is involved in the design loop as the need to accurately evaluate very complex configurations and the required high number of CFD simulations represents a further issue. The engineering computational design process may be speeded up by either accelerating the high-fidelity evaluation or reducing/accelerating the steps of the numerical algorithm used for exploring the design space. In the first group, High-Performance Computing (HPC) methods are numbered to increase the global performance of a single call to the CFD solver (through code parallelization, vectorization, and profiling). With reference to the second group, several research trends are focused on reducing the problem dimensionality (hence, the iterations needed to find the solution), improving the search algorithms for achieving faster and better solutions, and tuning the algorithm parameters to automatically and efficiently adapt to the response. An interesting alternative branch is represented by surrogate-based or metamodel-assisted optimization (SBO) which was originally conceived to relieve the computational loading associated with the usage of black box response functions. SBO consists in replacing the high-fidelity model (or "truth" model, e.g., the CFD simulation) with a fast, lower-fidelity model which has preliminarily "learned" from high-fidelity data. Since the pioneering work by Jones et al. [1], several theoretical studies [2–18] have been published on the topic. The proposed methods differ for one of the following items: the employed surrogate model (e.g., model type and single or multiple models), the training approach (e.g., optimizing the prediction error, the cross-validation error, the generalized cross-validation error, and the likelihood function), the model updating strategy (e.g., usage of surrogate minimizers, infill criteria, and random criteria), and the optimization method adopted to find the model parameters and to explore the surrogate (e.g., heuristic, gradient-free or gradient-based, and global or local). SBO has been successfully applied in the aerospace engineering field [19–27]. In continuity with other works by the author [28–31], the present paper proposes an adaptive SBO framework for design optimization with

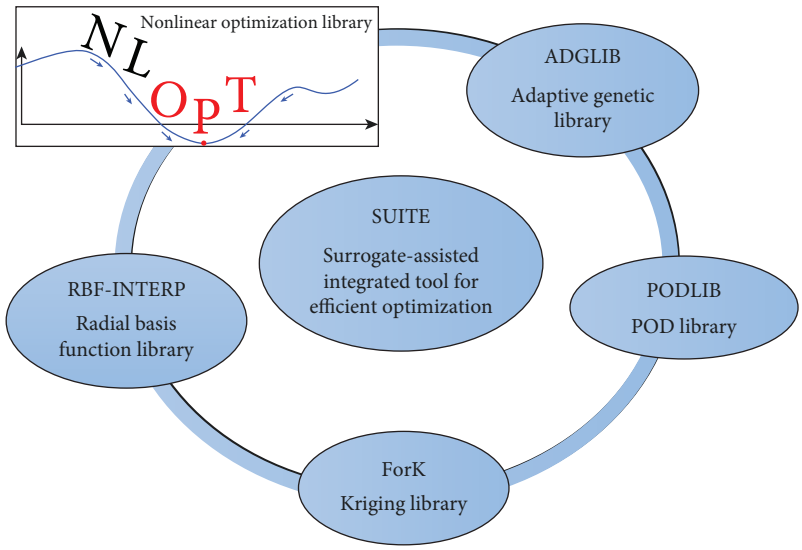


Figure 1: Surrogate-based optimization suite.

different updating strategies and optimization algorithms. A sketch of the main building blocks is provided in Figure 1. A choice of surrogate models is also available for selection. The main novelty of the paper is in the proposal of two groups of diverse infill strategies and in the capability to apply many of them during the adaptive sampling cycle by defining activation probabilities. Moreover, the usage of two different optimization libraries (public domain NLopt and in-house ADGLIB) allows performing both hyperparameter optimization for surrogate model training and objective function minimization with a variety of approaches. Finally, while previous investigations focused only on aerodynamic optimization cases, here an extensive study is carried out on analytic test functions in order to quickly assess the performance of the algorithm on known data. The paper is structured as follows: the first section is devoted to the surrogate model definition and to the training methods; then, the sequential design by means of various infill criteria is discussed and some examples on a basic test function are proposed; furthermore, having introduced all the computational pieces, the whole surrogate-based sequential optimization algorithm is described in detail and interactions between subphases are highlighted; finally, the experimental test campaign on multidimensional scalable test functions is discussed as well as the results obtained by using different setups; an example of real-world application is given in the very final section where a benchmark aerodynamic shape optimization case is faced and results are compared with a previous work by the same author [31].

2. Surrogate Models

2.1. Kriging Model. The Kriging model assumes that the function value at each point in the domain is represented by a separate random variable correlated with all the other points. Given that $f(\mathbf{x})$ is the function response of interest, a Kriging surrogate is defined as a realization of a regression model h and a stochastic process z having zero mean, process variance σ_k^2, and covariance model $\mathbf{R}(\theta, \mathbf{x_1}, \mathbf{x_2})$ between $z(\mathbf{x_1})$ and $z(\mathbf{x_2})$ with parameter vector θ [32]:

$$f(\mathbf{x}) = h(\beta, \mathbf{x}) + z(\mathbf{x}), \tag{1}$$

$$h(\beta, \mathbf{x}) = \mathbf{h}\beta, \tag{2}$$

$$\mathbb{E}[z(\mathbf{x_1}), z(\mathbf{x_2})] = \sigma_k^2 \mathbf{R}(\theta, \mathbf{x_1}, \mathbf{x_2}), \tag{3}$$

where β is the regression coefficient vector and $\mathbf{h}$ is the regression vector. The correlation between training sites $\{\mathbf{x}_j\}_{j=1,\cdots,M}$ is condensed within the covariance matrix and given by $\mathbf{K}_{ij} = \mathbf{R}(\theta, \mathbf{x}_i, \mathbf{x}_j)$. In multidimensional cases, the covariance is obtained as a tensor product of one-dimensional covariance functions:

$$\mathbf{R}(\theta, \mathbf{x}_i, \mathbf{x}_j) = \prod_p^D Kr\left(\left|\frac{x_{ip} - x_{jp}}{\theta_p}\right|\right), \tag{4}$$

where D is the dimension of the problem, θ_p is the length scale in the p-th dimension, x_{ip} is the p-th component of the vector $\mathbf{x}_i$, and Kr is the one-dimensional Matern function:

$$Kr(d) = \exp\left(-\sqrt{2v}d\right)\frac{\Gamma(t+1)}{\Gamma(2t+1)}\sum_{i=0}^{t}\frac{(t+1)!}{i!(t-1)!}\left(\sqrt{8v}d\right)^{t-i}, \tag{5}$$

with Γ the Gamma function, $\nu = (t+1)/2$, and three possible values of the parameter t:

$$Kr(d) = \begin{cases} \exp(-d), & \text{for } t = 0, \\ \left(1 + \sqrt{3}d\right) \exp\left(-\sqrt{3}d\right), & \text{for } t = 1, \\ \left(1 + \sqrt{5}d + \frac{5}{3}d^2\right) \exp\left(-\sqrt{5}d\right), & \text{for } t = 2. \end{cases} \tag{6}$$

In practice, the covariance matrix may become ill-conditioned or regression features are required to handle noisy functions. As a consequence, "nugget" or noise terms are plugged in the covariance matrix diagonally which becomes $K_{ij} = R(\theta, \mathbf{x}_i, \mathbf{x}_j) + \lambda\delta_{ij}$, where the Kronecker convention has been used and λ is the noise ratio. Predictions at a generic location $\mathbf{x}$ are obtained by using the following expression:

$$\widehat{f}(\mathbf{x}) = \mathbf{H}^T\beta + \mathbf{k}^T\mathbf{K}^{-1}\left(\mathbf{f} - \mathbf{H}^T\beta\right), \tag{7}$$

where $\mathbf{H}$ is the linear regression matrix and β is the generalized least square estimate of β, $\mathbf{K}$ is the covariance matrix, $\mathbf{k}$ is the covariance vector between the generic design site $\mathbf{x}$ and the training sites, and $\mathbf{f} = [f_1, f_2, \cdots, f_M]^T$ is the vector of function values. The stochastic nature of the Kriging model allows the obtainment of an estimate of the prediction variance in the form:

$$\widehat{s}^2(\mathbf{x}) = \sigma_k^2\left[1 - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k} + \mathbf{u}^T\left(\mathbf{H}^T\mathbf{K}^{-1}\mathbf{H}\right)^{-1}\mathbf{u}\right], \tag{8}$$

where σ_k^2 is the estimated process variance, $\mathbf{u} = \mathbf{H}^T\mathbf{K}^{-1}\mathbf{k} - \mathbf{h}$, and $\mathbf{h} = [h_1, h_2, \cdots, h_M]^T$. The prediction and its variance are both functions of the hyperparameters, i.e., the length scales θ_p, the process variance σ_k^2, and the noise magnitude λ. The hyperparameters have to be tuned in order to make the model output more "likely" against a given set of training data. In other words, the aim is to maximize the probability that the observed data follows the Gaussian process assumed with a specific set of hyperparameters. This is typically achieved by optimizing the maximum likelihood estimator (MLE) [33]. In the following, two optimization strategies are proposed and hereinafter referred to as "full optimization" and "partial optimization." The first is required when the function noise has to be fitted, while the second is required when ill conditioning of the covariance matrix is likely to occur. In both cases, the optimization algorithms are called from the NLopt library (available online at http://ab-initio.mit.edu/nlopt) in a loosely coupled global-local approach: first, a global exploration is performed with the evolutionary strategy ESCH algorithm [34]; then, the best solution is taken as the starting point of a local refinement by using a reviewed version of the Nelder-Mead simplex algorithm [35].

2.1.1. Full Optimization. In this approach, the regression parameters are found by imposing an optimality condition and given by

$$\beta = \mathbf{A}^{-1}\mathbf{H}\mathbf{K}^{-1}\mathbf{f}. \tag{9}$$

All the hyperparameters (length scales, process variance, and noise level) are obtained through the maximization of the likelihood function given by

$$\log p\left(f \mid x; \theta_p\right) = -\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\left(\mathbf{f} - \mathbf{H}^T\beta\right) - \frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}\log|\mathbf{A}| - \frac{M-S}{2}\log 2\pi, \tag{10}$$

where M is the number of training points, S is the number of terms in the regression, and the regression matrix $\mathbf{A}$ is defined as

$$\mathbf{A} = \mathbf{H}\mathbf{K}^{-1}\mathbf{H}^T. \tag{11}$$

2.1.2. Partial Optimization. In this formulation, the process variance and regression parameters are both computed thanks to the optimality condition; hence, the optimization process involves only the covariance length scales θ_p. The optimal process variance is set to the value which cancels the partial derivative of the likelihood function with respect to σ_k, given by

$$\widehat{\sigma}_k^2 = \frac{\left(\mathbf{f} - \mathbf{H}^T\widehat{\beta}\right)^T\mathbf{K}^{-1}\left(\mathbf{f} - \mathbf{H}^T\widehat{\beta}\right)}{M}. \tag{12}$$

The likelihood function to be maximized reduces to

$$\log p\left(f \mid x; \theta_p\right) = -\frac{M}{2}\log\widehat{\sigma}_k^2 - \frac{1}{2}\log\left|\widehat{\mathbf{K}}\right| - \frac{M}{2} - \frac{M}{2}\log 2\pi, \tag{13}$$

$$\mathbf{K} = \widehat{\sigma}_k^2\widehat{\mathbf{K}}. \tag{14}$$

This final formula is a function of the length scales θ_p and the noise level ratio λ. The optimization is performed only over the length scales, and the noise level ratio is fixed throughout the optimization. A typical choice is to set the noise level λ to a small fraction of the process variance $\widehat{\sigma}_k$ to avoid ill conditioning.

2.2. Radial Basis Function Network. Radial basis functions (RBF) are a powerful tool for data interpolation and regression. The response depends on the location of centres and on the Euclidean distance from the centres. By appropriately defining M centres and weighting the contribution of each RBF, a RBF network is obtained as

$$\widehat{f}(\mathbf{x}, \theta_1, \cdots, \theta_M, \lambda) = \sum_{i=1}^{M} k_i(\lambda) r(|\mathbf{x} - \mathbf{x}_i|, \theta_i). \tag{15}$$

Here, the centre of each RBF is $\mathbf{x}_i$ and the weights are k_i (which, in turn, depend on a regularization parameter λ). The amplitude of the radial basis functions is controlled by means of the width parameters θ_i. The type of the RBF kernel r may assume various mathematical forms; here, the following types are considered ($d = |\mathbf{x} - \mathbf{x}_i|$ is the Euclidean distance from the centre):

$$r(d,\theta) = \begin{cases} \exp\left(-\dfrac{d^2}{\theta^2}\right), & \text{Gaussian,} \\ \sqrt{1+\dfrac{d^2}{\theta^2}}, & \text{multiquadric,} \\ \dfrac{1}{\sqrt{1+\left(d^2/\theta^2\right)}}, & \text{inverse multiquadric,} \\ \left(\dfrac{d}{\theta}\right)^2 \ln\dfrac{d}{\theta}, & \text{thin plate spline,} \\ 1-30\left(\dfrac{d}{\theta}\right)^2 - 10\left(\dfrac{d}{\theta}\right)^3 + & \\ +45\left(\dfrac{d}{\theta}\right)^4 - 6\left(\dfrac{d}{\theta}\right)^5 - 60\left(\dfrac{d}{\theta}\right)^3 \log\left(\dfrac{d}{\theta}\right), & \text{Wendland } C^2 \text{ thin plate spline.} \end{cases} \tag{16}$$

Great emphasis has been put in the RBF literature on the choice of the width parameters. Gutmann [36] showed that both the interpolation error and the solution matrix conditioning are highly sensible to the value of θ: in particular, theta should be low enough to improve stability; however, the highest approximation accuracy is often found for large θ value which may lay in the unstable region. As a consequence, a conflict arises between accuracy and stability, sometimes referred to as the "trade-off principle" [37]. This is solved by using state-of-the-art optimization techniques and considering a unique scalar width θ for each RBF centre. The optimization algorithm autonomously chooses the kernel function type and optimizes the width parameters taking the leave-one-out cross-validation error as the objective function to be minimized. Similar to Tenne and Armfield [23], the procedure works as follows:

(1) All the kernel functions are trained on the current training set

(2) The leave-one-out (LOO) error norm is computed as

$$\varepsilon_{\mathrm{LOO}}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M, \theta, \lambda) = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\left[f_j - \widehat{f}_{-j}(\mathbf{x}_j, \theta, \lambda)\right]^2}, \tag{17}$$

where f_j is the value of the function at the j^{th} training site $\mathbf{x}_j$ and $\widehat{f}_{-j}$ is the RBF prediction at $\mathbf{x}_j$ when the model is trained without $\mathbf{x}_j$ and f_j. The computation of the M terms $\widehat{f}_{-j}$ does not require the training of M RBF models; indeed, it can be computed effortlessly thanks to Rippa's formula [38]

(3) The combination of the RBF kernel and width parameter which gives the lowest LOO error norm is sought by solving, for each kernel, the optimization problem:

$$\min_{\theta,\lambda} \varepsilon_{\mathrm{LOO}}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M, \theta, \lambda) \tag{18}$$

Two options are available:

(i) Grid search: a grid of discrete couples (θ, λ) is generated, all the possible combinations of parameter values are evaluated, and the best combination (in terms of minimum $\varepsilon_{\mathrm{LOO}}$) is retained

(ii) Numerical optimization: the minimization problem (18) is solved considering the hyperparameters as continuous variables and using the same algorithms for searching the Kriging hyperparameters

Once the optimal width parameters were found, the weights k_i have to be computed. A regularization parameter λ (also known as ridge regression parameter in the RBF literature) is introduced to avoid overfitting and improve

the interpolation matrix conditioning. By imposing the interpolation condition [37] on the training set, the weights are the solution of the linear system:

$$\mathbf{Rk} + \lambda \mathbf{I} = \mathbf{f}, \tag{19}$$

where

$$\mathbf{R} = \begin{Bmatrix} r(0,\theta_1) & \cdots & r(|\mathbf{x}_1 - \mathbf{x}_M|,\theta_M) \\ r(|\mathbf{x}_2 - \mathbf{x}_1|,\theta_1) & \cdots & r(|\mathbf{x}_2 - \mathbf{x}_M|,\theta_M) \\ \vdots & \vdots & \vdots \\ r(|\mathbf{x}_M - \mathbf{x}_1|,\theta_1) & \cdots & r(0,\theta_M) \end{Bmatrix} \tag{20}$$

$\mathbf{k} = [k_1, k_2, \cdots, k_M]^T$ are the unknown RBF weights, and $\mathbf{f} = [f_1, f_2, \cdots, f_M]^T$ are the function values at the training sites. The prediction variance is estimated as

$$\hat{s}^2(\mathbf{x}) = \sigma^2_{\text{RBF}} \left| 1 - \mathbf{r}\mathbf{R}^{-1}\mathbf{r}^T \right|, \tag{21}$$

where $\mathbf{r} = [r(|\mathbf{x} - \mathbf{x}_1|,\theta_1), \cdots, r(|\mathbf{x} - \mathbf{x}_M|,\theta_M)]$, $\sigma^2_{\text{RBF}} = (1/M-1)\Sigma_{i=1}^{M}(f_i - f_{\text{mean}})^2$, and $f_{\text{mean}} = (1/M)\Sigma_{i=1}^{M} f_i$.

3. Sequential Design and Infill Criteria

This section will give details about the sequential enrichment of a surrogate model assisting the optimization task. Indeed, once a surrogate model is trained, a basic and simplistic approach would be to optimize the surrogate and find the model minimizers. A further step could be to reevaluate the suggested points with the high-fidelity model, update the training dataset, build a new surrogate, and then optimize again in an iterative manner to drive to true optimality quickly. However, feeding the surrogate back with no information about its error measure and no tendency to exploration would easily lead the updating process to get trapped in local minima. The weak point lies in considering the model prediction as the only source of "knowledge" for increasing the quality of the approximation ("exploitation"). An advanced solution is obtained by mixing the information coming from the available data, the surrogate prediction, and the estimation of the predictive behaviour away from the training set ("exploration"): this could drive to a "smarter" selection of new points and, thus, improve the surrogate.

The aim is to obtain a model that cleverly supports the optimization path, being ideally very accurate near the global/local optimal location and acceptably rough elsewhere. Of course, the updating strategy has to take into account the specific surrogate model at hand, so that the adaptive sampling criteria should be designed upon its features (e.g., vector or scalar data, availability of an estimated prediction error, and interpolation/regression character). Adaptivity is another key point in view of inserting new points depending on the samples and response function values collected so far. The exploration/exploitation trade-off usually drives the adaptation by mixing the contribution of high prediction error areas (exploration) with potentially promising regions (exploitation). Indeed, explorative search is a cornerstone for global optimization; however, it may lead to the continuous unveiling of poor regions of the design space; on the other hand, exploitation induces to trust the surrogate prediction, which surely improves the local accuracy but may also lead to being stuck in local minima. Only a proper and balanced combination of both components will be effective in leading to an efficient global optimization.

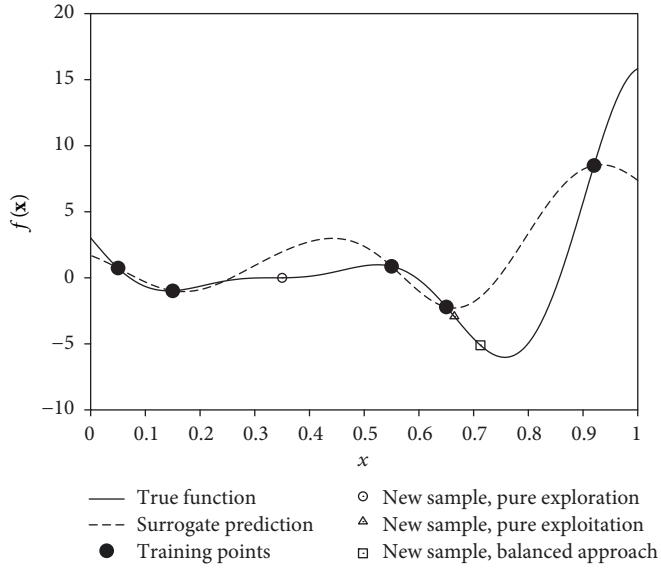


Figure 2: Exploitation vs. exploration, 1D example (from Iuliano and Pérez 2016 [30]).

By way of example, Figure 2 illustrates the addition of a new point obtained, respectively, by employing pure exploitation, pure exploration, and balanced approaches. The picture shows the set of training points (black-filled circles), the true function (solid black line), and the surrogate prediction (dashed black line) built on the training points. Starting from this, a pure exploitation criterion places a new point at the surrogate global minimum, i.e., very close to one of the training points (triangle mark); by pursuing pure exploration, instead, the new point is located very far from the training set (circle mark), thus sampling is performed where the prediction uncertainty is highest; finally, the new point predicted with a balanced exploration/exploitation approach (square mark) is located in an interesting position very close to the true optimum: a new surrogate model trained on the old training set plus the new point will surely lead to the detection of the global optimum quickly.

Infill criteria are here referred to as means for adding new samples to the training set by designing auxiliary functions for generic surrogate models. Let us say $\mathbf{x}$ is the design vector which defines a generic location in the design space, $f(\mathbf{x})$ is the objective function to be minimized, $\{X_n\} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ is the set of n available training points, $\{F_{X_n}\} = \{f(\mathbf{x}_1), f(\mathbf{x}_2), \cdots, f(\mathbf{x}_n)\}$ is the corresponding set of true objective function values, and $\widehat{f}(\mathbf{x})$ is the response at $\mathbf{x}$ of the surrogate model built on $(\{X_n\}, \{F_{X_n}\})$.

An infill criterion is defined as finding a new sample $\mathbf{x}_{n+1}$ which maximizes an auxiliary function v called the potential of improvement: $\mathbf{x}_{n+1} = \arg\ \max_{\mathbf{x}} v(\mathbf{x}, \widehat{f}(\mathbf{x}), X_n, F_{X_n})$.

Hereinafter, the maximization of v is not achieved by using numerical optimization techniques, but rather by hugely sampling the design space (e.g., five hundred times its dimension) with a Latin hypercube sampling method and selecting the point at which the maximum value of v is met. Despite the size of the test dataset, the evaluation of v requires limited computational effort as the function evaluation involves the surrogate prediction, which is fast to compute, and the true objective function values (already collected). An important point is related to the fact that, as the test dataset is generated many times along with multiple updating iterations, duplication of the selected sample may occur. In order to avoid this, the seed of the Latin hypercube is changed unambiguously at each iteration.

As concerns the type and nature of the potential of improvement function v, previous investigations [30] showed that error-driven infill criteria may lead to the intensive exploration of the design space in order to reduce the prediction error, but, conversely, this resulted in a lack of efficiency of the whole optimization process at fixed total computational budget. Hence, in the following section, the discussion will focus on approaches which proved to be more suitable to global optimization. In particular, two sets of criteria will be proposed: the first is based on the factorization of the potential of improvement in order to explicitly realize the trade-off between exploration and exploitation and the second is defined according to the expected improvement concept.

3.1. Factorized Infill Criteria. The first proposal of adaptive infill criterion is aimed at combining exploration and exploitation by means of a generic factorization as follows:

$$v\left(\mathbf{x}, \widehat{f}(\mathbf{x}), X_n, F_{X_n}\right) = g(\mathbf{x}, X_n) h\left(\widehat{f}(\mathbf{x}), F_{X_n}\right). \tag{22}$$

The functions g and h measure the exploration and model trust contribution, respectively. In particular, the exploration function g should estimate how strong the influence of the set of already collected samples X_n on a generic candidate $\mathbf{x}$ is. One of the preferred approaches is to make the function dependent on the Euclidean distance $d(\mathbf{x}, \mathbf{x}_i)$ between the generic design space location $\mathbf{x}$ and the i-th element of the training set X_n:

$$g(\mathbf{x}, X_n) = g(d(\mathbf{x}, \mathbf{x}_1), d(\mathbf{x}, \mathbf{x}_2), \cdots, d(\mathbf{x}, \mathbf{x}_n)). \tag{23}$$

On the other hand, the exploitation function h should take into account how the surrogate prediction $\widehat{f}$ compares with the available set of true objective function values F_{X_n}. In particular, this contribution should put emphasis on trusting the model prediction; hence, the h function should exhibit its maxima in correspondence with some identified features of $\widehat{f}$ (e.g., minima, discontinuities, or local strong nonlinearities). Of course, different infill criteria can be designed by properly defining the functions g and h. A set of choices is presented here.

3.1.1. Leave-One-Out Error Criterion. The leave-one-out error (LOOE) criterion is aimed at individuating the regions where the surrogate model lacks accuracy and is much more sensible to the insertion of new designs. The factorization functions are as follows:

$$g(\mathbf{x}, X_n) = \frac{\min_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i)}{\max_{\mathbf{x}_i \mathbf{x}_j \in X_n} d(\mathbf{x}_i, \mathbf{x}_j)},$$
$$h\left(\widehat{f}(\mathbf{x}), F_{X_n}\right) = \frac{\left|\widehat{f}_{-i}(\mathbf{x}_i) - f(\mathbf{x}_i)\right|}{\Sigma_{j=1}^{n}\left|\widehat{f}_{-j}(\mathbf{x}_j) - f(\mathbf{x}_j)\right|}, \tag{24}$$

where, given a generic location $\mathbf{x}$, $\mathbf{x}_i$ is the location of the nearest training sample. Of course, as this criterion will tend to select new candidates around training samples exhibiting the highest LOO error, the clustering of points around specific regions will be avoided.

3.1.2. Weighted Leave-One-Out Error Criterion. As the LOOE criterion may be too much exploratory and ignore the information given by the surrogate model, an alternative is given by weighting the function with a term which measures the trust in the surrogate prediction. The weighted leave-one-out error (WLOOE) criterion modifies the h function as follows:

$$h\left(\widehat{f}(\mathbf{x}), F_{X_n}\right) = \frac{\left|\widehat{f}_{-i}(\mathbf{x}_i) - f(\mathbf{x}_i)\right|}{\Sigma_{j=1}^{n}\left|\widehat{f}_{-j}(\mathbf{x}_j) - f(\mathbf{x}_j)\right|} \exp\left(-\sigma \frac{\widehat{f}(\mathbf{x}) - f_{\min}}{f_{\max} - f_{\min}}\right), \tag{25}$$

where σ is a tuning parameter, $f_{\min} = \min\{f_{\mathbf{x}_1}, \cdots, f_{\mathbf{x}_n}\}$, and $f_{\max} = \max\{f_{\mathbf{x}_1}, \cdots, f_{\mathbf{x}_n}\}$. This choice of the h function provides two main features:

(1) The value of h approaches the LOOE prediction when $\widehat{f}(\mathbf{x})$ approaches $f_{\min}$

(2) For $\widehat{f}(\mathbf{x}) < f_{\min}$, the value of the h function is higher than the LOOE

With respect to the LOOE criterion, the multiplicative exponential term augments the error-minimizing term with a goal-oriented exploitation term: hence, "bad" candidates (according to the surrogate prediction) will be filtered out, while "good" candidates will be recognized and rewarded with higher rank.

3.1.3. Lipschitz Constant Criterion. This criterion (LC) is aimed at selecting new design samples where the local complexity of the function is high. The Lipschitz constant is considered here as an indicator of the local complexity. Given a

1: compute the K-means clusters $K_{j,j=1,r}$ of the set $X_n = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ with $r \leq \text{int}\left(\frac{n}{d}\right)$
2: **for all** sample $\mathbf{x}_i \in X_n$ **do**
3: Say K_i the cluster containing $\mathbf{x}_i$
4: **for all** sample $\mathbf{x}_j \in K_i, \mathbf{x}_j \neq \mathbf{x}_i$ **do**
5: compute $L_{ij} = \frac{|f(\mathbf{x}_i) - f(\mathbf{x}_j)|}{|\mathbf{x}_i - \mathbf{x}_j|}$
6: **end for**
7: Set $L(\mathbf{x}_i) = \max_j L_{ij}$
8: **end for**

Algorithm 1. Lipschitz constant estimation.

domain D and a function f defined in D, the Lipschitz constant denotes the smallest constant $L > 0$ in the Lipschitz condition, namely, the nonnegative number:

$$L_{f,D} \coloneqq \sup_{\substack{x_1, x_2 \in D \\ x_1 \neq x_2}} \frac{|f(x_1) - f(x_2)|}{|x_1 - x_2|}. \tag{26}$$

Such a constant is usually employed to bound the nonlinear character of the function f: for instance, it gives an upper bound on the number of oscillations of a given amplitude or it limits the maximum and minimum value a function can assume in a given range. Of course, it has a local character; hence, a function may exhibit subregions with either small or large values of the constant. In the present context, the Lipschitz constant has to be estimated at every possible location within the design space. This is done by computing the K-means clusters of X_n and considering the variation of f between the nearest training sample and the set of all nodes belonging to the cluster. Algorithm 1 details the estimation of the Lipschitz constant.

The effective number of clusters r is the result of an iterative cluster solution: indeed, it is required to have at least two candidates in each cluster in order to be able to compute the Lipschitz constant and, in some cases, this may not occur naturally (e.g., strong aggregation of training samples). As a consequence, r is initialized to int (n/d), but it is downgraded every time a single-component cluster is found. According to the algorithm, a Lipschitz constant value is associated with each training sample: at a generic location $\mathbf{x}$, it is assumed that the Lipschitz constant is the same as the nearest training sample, i.e., $L(\mathbf{x}) = L(\mathbf{x}_{nn})$, where $\mathbf{x}_{nn} = \arg\ \min_{\mathbf{x}_i \in X_n} |\mathbf{x}_i - \mathbf{x}|$.

The functions g and h for the LC criterion are defined as

$$\begin{aligned} g(\mathbf{x}, X_n) &= \min_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i), \\ h\left(\mathbf{x}, X_n, F_{X_n}\right) &= \frac{L(\mathbf{x})}{(f_{\max} - f_{\min})} \exp\left(-\gamma \frac{\max_{\mathbf{x}_i, \mathbf{x}_j \in X_n} d\left(\mathbf{x}_i, \mathbf{x}_j\right)}{\min_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i)}\right), \end{aligned} \tag{27}$$

where the division by $(f_{\max} - f_{\min})$ has been introduced for normalization purposes and the exponential term provides for a tuned decay weight (through parameter γ) of the h function. In fact, when the design candidate $\mathbf{x}$ is near to a training point, the term $\min_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i)$ is small and the weight of the local Lipschitz constant is accordingly decreased with respect to a design candidate belonging to the same cluster and located farther.

3.1.4. Weighted Distance Criterion. The weighted distance (WD) criterion is based on the following definitions for functions g and h:

$$\begin{aligned} g(\mathbf{x}, X_n) &= \frac{\min_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i)}{\max_{\mathbf{x}_i, \mathbf{x}_j \in X_n} d\left(\mathbf{x}_i, \mathbf{x}_j\right)}, \\ h\left(\widehat{f}(\mathbf{x}), F_{X_n}\right) &= \exp\left(-\sigma \frac{\widehat{f}(\mathbf{x}) - f_{\min}}{f_{\max} - f_{\min}}\right), \end{aligned} \tag{28}$$

where the nomenclature is the same as described above. Again, the exploitation is given by the exponential term which gives confidence in the surrogate prediction, while the exploration element is represented by how far the candidate $\mathbf{x}$ is from the nearest training sample. Analogous to the WLOOE criterion, candidates with a surrogate prediction lower than the current function minimum will be more likely to be selected. However, if they are too close to samples stored in X_n, they will be penalized by the g function. Hence, a trade-off is realized between surrogate prediction and location in the design space.

3.2. Expected Improvement-Based Infill Criteria. Another set of infill criteria is defined to accomplish the exploration-exploitation trade-off from a different point of view. The expected improvement algorithm by Schonlau et al. [39] and Jones et al. [1] represents a standard design strategy to add one new sample aimed at achieving the global optimum of the response surface. In the following, the original criterion and some variants of it are presented.

3.2.1. Expected Improvement (EI) Criterion. The sequential algorithm is based on the notion of "improvement" defined by

$$I(\mathbf{x}) = \begin{cases} f_{\min} - F(\mathbf{x}), & \text{if } F(\mathbf{x}) < f_{\min}, \\ 0, & \text{otherwise}, \end{cases} \tag{29}$$

where the function $F(\mathbf{x})$ is here supposed to be a random variable as for the Kriging model (see equation (1)). After integration with respect to the conditional distribution $F(\mathbf{x}) \sim N(\widehat{f}(\mathbf{x}), \widehat{s}^2(\mathbf{x}))$, the expected improvement (EI) function is expressed as

$$\begin{aligned} v\left(\mathbf{x}, \widehat{f}(\mathbf{x}), X_n, F_{X_n}\right) &= E[I(\mathbf{x})] \\ &= \left(f_{\min} - \widehat{f}(\mathbf{x})\right)\Phi\left(\frac{f_{\min} - \widehat{f}(\mathbf{x})}{\widehat{s}(\mathbf{x})}\right) \\ &\quad + \widehat{s}(\mathbf{x})\phi\left(\frac{f_{\min} - \widehat{f}(\mathbf{x})}{\widehat{s}(\mathbf{x})}\right), \end{aligned} \tag{30}$$

where $\widehat{f}(\mathbf{x})$ is the Kriging predictor (equation (7)), $\widehat{s}^2(\mathbf{x})$ is the corresponding prediction variance (equation (8)), and $\Phi(\mathbf{x})$ and $\phi(x)$ are, respectively, the $N(0,1)$ cumulative distribution and probability density functions. The search for the global minimum is enriched by finding the point $\mathbf{x}$ that maximized the EI function. Two additive contributions are clearly evident:

(i) The first term gives importance to sample points having predicted values much less than $f_{\min}$ (thus exploiting the surrogate model)

(ii) The second term enhances samples with high uncertainty about the prediction (thus fostering global search)

Both terms are weighted by the probability measure of the standard normal distribution defined in the modified variable $(f_{\min} - \widehat{f}(\mathbf{x}))/\widehat{s}(\mathbf{x})$: hence, depending on this ratio, the EI landscape is usually featured with many sharp peaks and wide regions where its value is almost zero. Equation (30) may be used even if the prediction model is not stochastic in nature: for example, the prediction function and the associated variance coming from a RBF network (equations (15) and (21)) may be plugged in it and adopted as EI infill criterion for choosing new points.

3.2.2. EI-Like Criterion. This criterion has been designed trying to mimic the same rationale of the expected improvement criterion, which was originally conceived for a Gaussian process surrogate. The present approach, referred to as "EI-like" hereinafter, represents a generalization of that algorithm: indeed, for a generic surrogate model, the information about the prediction uncertainty may not be available as in the case of Kriging or RBF network model. The idea is to define a model for the prediction error which is theoretically applicable to any function and any surrogate and to link it to the local complexity of the true function. The potential of improvement function is designed to have the same form of the expected improvement function (equation (30)), but here the prediction error is estimated as follows:

$$\widehat{s}(\mathbf{x}) = L(\mathbf{x}) \max_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i) \exp\left(-\gamma \frac{\max_{\mathbf{x}_i, \mathbf{x}_j \in X_n} d(\mathbf{x}_i, \mathbf{x}_j)}{\min_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i)}\right), \tag{31}$$

where γ is a tuning parameter and $L(\mathbf{x})$ is the local Lipschitz constant estimated as in Section 3.1.3. The $\widehat{s}$ function is related to the order of magnitude of the local maximum difference of f: indeed, the Lipschitz constant is multiplied by a distance so as to make the $\widehat{s}$ quantity similar to Δf from a dimensional point of view. The prediction variance has been designed in order to increase with increasing distance from an available sample. The negative exponential term and the γ parameter in it allow for adjusting the rate of change of $\widehat{s}$ while moving away from known points: for low values of γ, the variance quickly increases and plateau-like regions are generated between samples, while for high values of γ, the rise is milder and a series of hills (midsamples) and valleys (near-samples) are generated. An example will be provided at the end of the section to better explain the landscape of the EI-like function.

3.2.3. Expected Improvement for Global Fit (EIGF) Criterion. A modified version of the expected improvement criterion is here considered as proposed by Lam [10]. Instead of locating the global optimum, the aim is to add new points which are located in regions with significant variation of the response function and to improve the global fit of the model. In other words, the rationale is close to the Lipschitz criterion. Similar to the EI criterion and adopting the same notation, the improvement function is here defined as

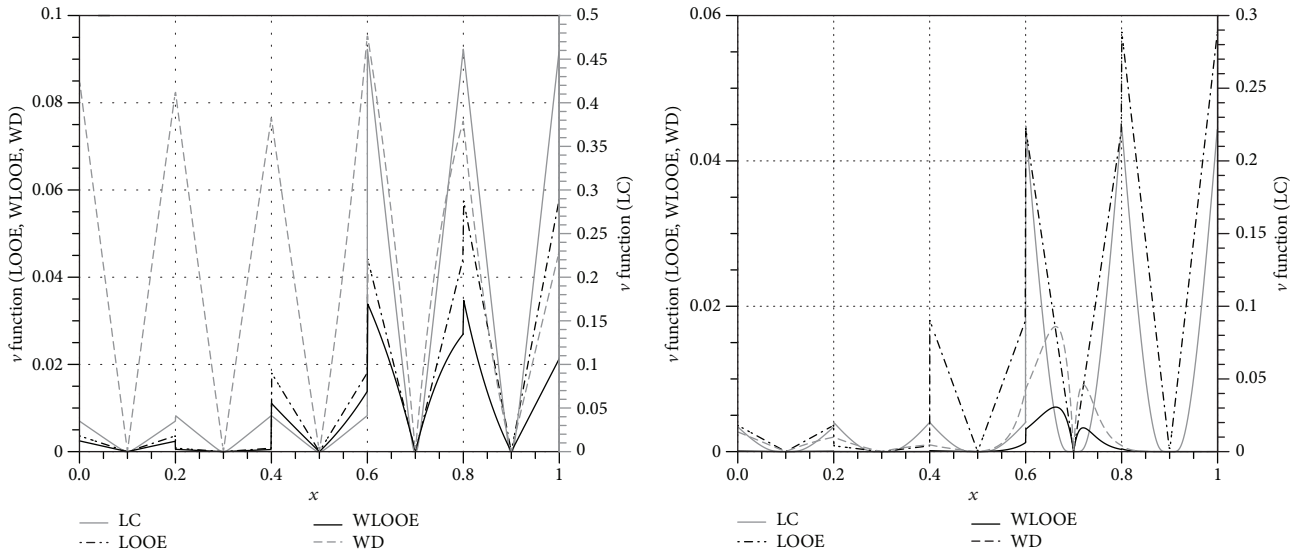
$$I(\mathbf{x}) = [F(\mathbf{x}) - f(\mathbf{x}_{nn})]^2, \tag{32}$$

where $\mathbf{x}_{nn} = \arg\min_{\mathbf{x}_i \in X_n} |\mathbf{x}_i - \mathbf{x}|$ is the training site closest (in distance) to $\mathbf{x}$. After taking the expected value of equation (32) and recalling that $F(\mathbf{x}) \sim N(\widehat{f}(\mathbf{x}), \widehat{s}^2(\mathbf{x}))$, the expected improvement for global fit function is derived as

$$v\left(\mathbf{x}, \widehat{f}(\mathbf{x}), X_n, F_{X_n}\right) = E[I(\mathbf{x})] = \left[\widehat{f}(\mathbf{x}) - f(\mathbf{x}_{nn})\right]^2 + \widehat{s}^2(\mathbf{x}). \tag{33}$$

Again, this potential of improvement function consists of two components, one local and one global. The first (local) is large where the predicted response varies significantly with respect to the nearest sample. The second (global) increases when the uncertainty in the prediction is high, i.e., far from the sampled points.

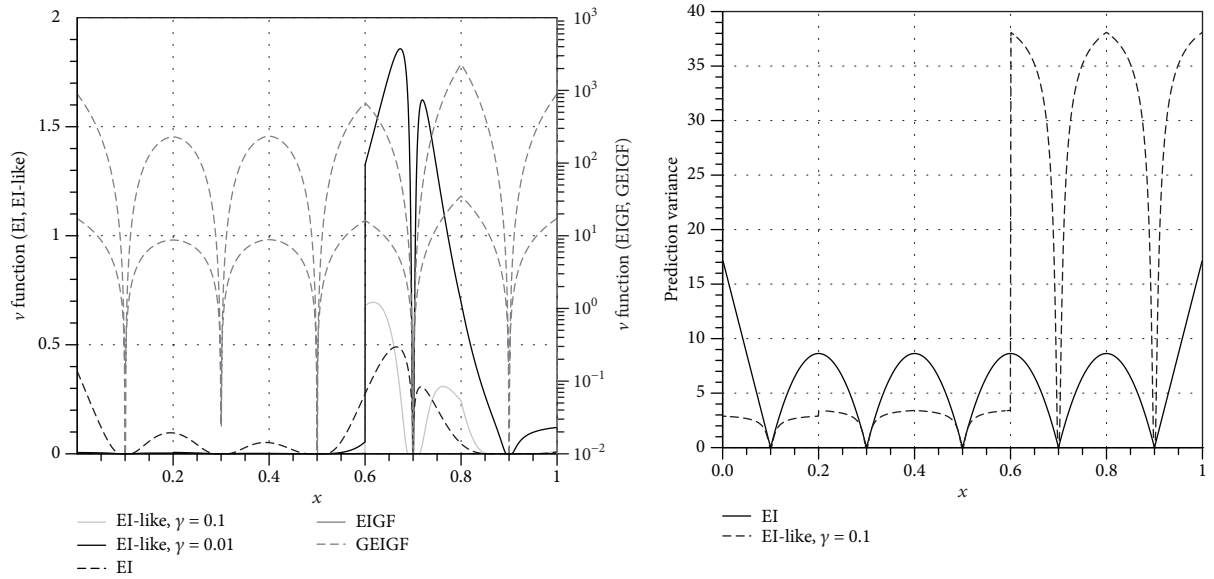
3.2.4. Generalized EIGF Criterion. Starting from equation (32) and taking the expected value of $I^2(\mathbf{x})$, the



(a) v functions for $\sigma = 1.0$ and $\gamma = 0.1$

(b) v functions for $\sigma = 10.0$ and $\gamma = 0.2$

Figure 3: Potential of improvement based on the factorization criterion, 1D example.



(a) v functions for EI-based criteria

(b) EI vs. EI-like prediction variance

Figure 4: EI-based criteria, 1D example.

generalized expected improvement for global fit function is obtained as

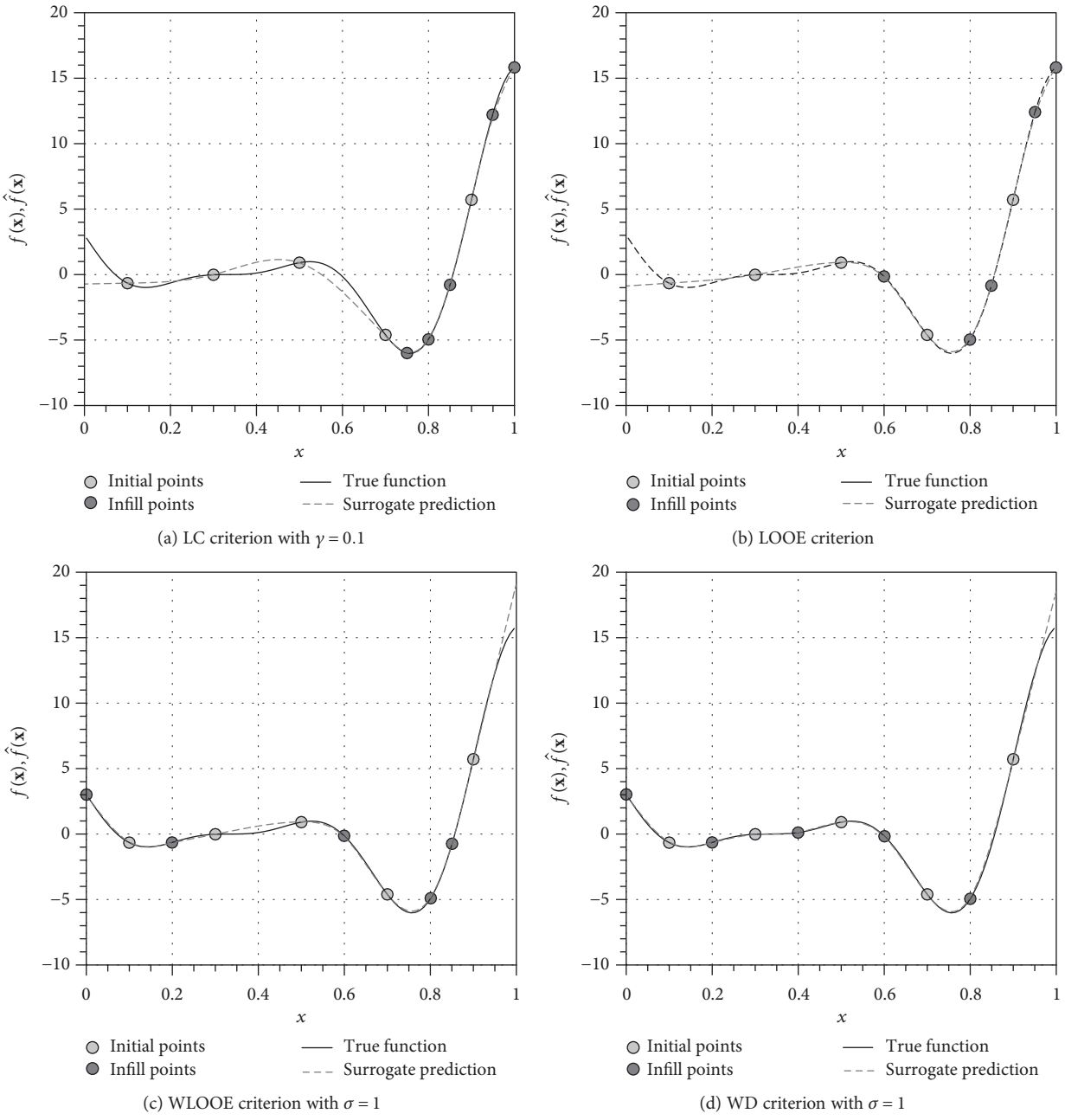
$$v\left(\mathbf{x}, \widehat{f}(\mathbf{x}), X_n, F_{X_n}\right) = E\left[I^2(\mathbf{x})\right] = \left[\widehat{f}(\mathbf{x}) - f(\mathbf{x}_{nn})\right]^4 + 6\left[\widehat{f}(\mathbf{x}) - f(\mathbf{x}_{nn})\right]^2 \widehat{s}^2(\mathbf{x}) + 3\widehat{s}^4(\mathbf{x}). \tag{34}$$

The main difference with respect to equation (33) is that now the change in response $[\widehat{f}(\mathbf{x}) - f(\mathbf{x}_{nn})]$ and the prediction uncertainty $\widehat{s}^2(\mathbf{x})$ are not separated as they interact with each other. This should drive the search to regions where both terms are important as their combined effect would be amplified.

3.3. Test Example. A simple illustrative example is given here by considering the one-dimensional function:

$$f(x) = \sin\,(12x - 4)(6x - 2)^2, \tag{35}$$

defined for $x \in [0, 1]$. A set of 5 training points $X_n = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ with associated set of function values



(a) LC criterion with $\gamma = 0.1$

(b) LOOE criterion

(c) WLOOE criterion with $\sigma = 1$

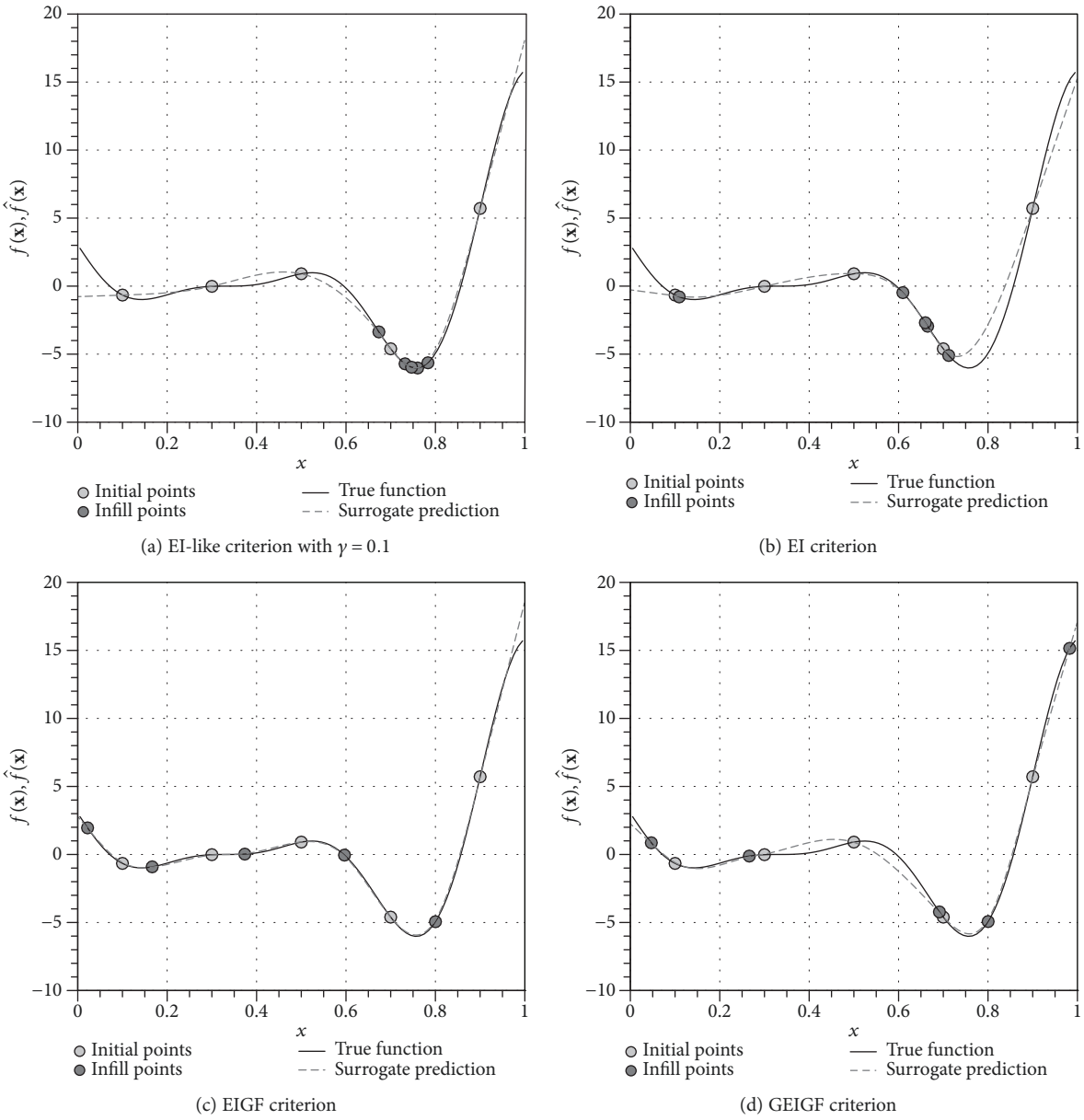
(d) WD criterion with $\sigma = 1$

Figure 5: Surrogate prediction after 5 infill points: factorization criteria.

$F_{X_n} = \{-0.656577, -0.015577, 0.909297, -4.605754, 5.711950\}$ has been extracted by uniformly sampling the design space $D = [0, 1]$: Figure 2 shows the function $f(x)$, the surrogate prediction $\widehat{f}(x)$ (here, a Kriging model is used), and the training samples. Figure 3 shows the potential of improvement functions obtained by applying the criteria described so far for two values of the tuning parameters. It can be observed that, apart from the LOOE criterion where no tuning parameter is introduced, the levels of ν are globally reduced with increasing σ or γ; indeed, a different scale of the y-axis is used.

The effect of the σ parameter is clearly observable by comparing the WLOOE curves: indeed, for $\sigma = 1$, the maximum value of the WLOOE criterion is located at $x = 0.8$ and the ν function is quite peaky, while for $\sigma = 10$, the peak shifts to $x = 0.65$ and the function is null almost everywhere. This occurs because the relative importance of the exploitation term (equation (25)) decreases rapidly with increasing σ, thus raising the exploration contribution within the factorization (equation (22)). The same behaviour is observed also for the WD criterion. The LC criterion, instead, suffers only a global and uniform damping of the peaks; hence, the location of high-ranking candidates is not altered by changing the tuning parameter.

Figure 4(a) shows the potential of improvement functions for EI-based criteria. The test function, the number



(a) EI-like criterion with $\gamma = 0.1$

(b) EI criterion

(c) EIGF criterion

(d) GEIGF criterion

Figure 6: Surrogate prediction after 5 infill points: EI-based criteria.

and location of the initial training samples, and the surrogate model are the same as in the previous section. Due to the different orders of magnitudes of the v function values, two scales are reported on the y-axis: the EI and EI-like functions have to be read on the left axis (ranging from 0.0 to 2.0), while the EIGF and GEIGF have to be read on the right axis (ranging from 10^{-2} to 10^{4} in a logarithmic scale). Both EI and EI-like functions have a global maximum around $x = 0.65$ (similar to WD and WLOOE criteria) and a lower peak around $x = 0.72$. The main difference between the two criteria is observable in the interval $x \in [0, 0.6]$, where the EI function exhibits three local maxima while the EI-like function is null. To better clarify this, Figure 4(b) shows the prediction variance as provided by the EI criterion (Kriging predictor variance, equation (8)) and by the EI-like criterion (equation (31)) for $\gamma = 0.1$. In fact, the EI-like prediction error is smaller where the variation in f is limited, as for $0 < x < 0.6$, and larger where significant gradients are present, while equation (8) depends only on the correlation between points, i.e., on distance and spatial distribution of points. EIGF and GEIGF criteria privilege the point at $x = 0.8$ (as LC and LOOE criteria) and show 5 more local maxima (corresponding to the interval end points and to the midpoints between the sample data) that may be selected in successive iterations of the method. This characteristic of placing samples "close to the midpoint" has been highlighted also by Lam who suggests to start with a small number of points from an LHS sampling in order to feed the EIGF function with a smooth predicted response.

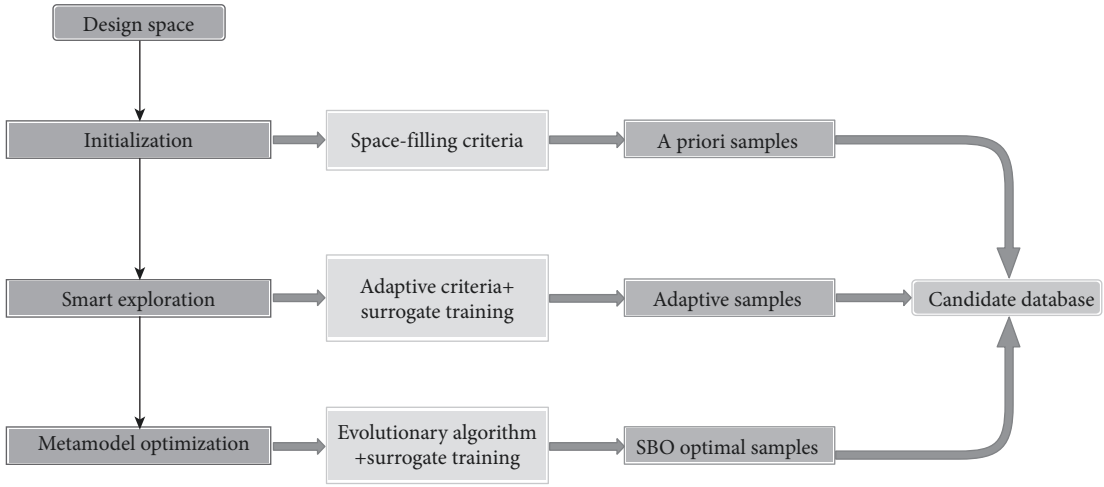


Figure 7: Workflow of surrogate-assisted optimization (from Iuliano, 2017 [31]).

Table 1: Global optimum of the Michalewicz function.

d	$f(\mathbf{x}_{\text{target}})$	$\mathbf{x}_{\text{target}}$
2	-1.8013034101	{2.2029, 1.5708}
5	-4.6876581791	{2.2029, 1.5708, 1.2850, 1.9231, 1.7205}
10	-9.6601517156	{2.2029, 1.5708, 1.2850, 1.9231, 1.7205, 1.5708, 1.4544, 1.7561, 1.6557, 1.5708}
20	-19.6370135993	{2.2029, 1.5708, 1.2850, 1.9231, 1.7205, 1.5708, 1.4544, 1.7561, 1.6557, 1.5708, 1.4977, 1.6966, 1.6301, 1.5708, 1.5175, 1.6661, 1.6163, 1.5708, 1.5289, 1.6475}

Figure 5 shows the result of 5 repeated updates of the surrogate model with each of the proposed criteria. The new 5 points are depicted with black circles, while the initial training points (the same set for every criterion) are represented by light gray circles. Each plot shows the test function and the surrogate prediction after the infill process. All proposed criteria manage to capture the location of the global optimum and to minimize the prediction error in the surrounding region. A strong clustering is observed for WLOOE and WD criteria, a clear effect of the exponential weight built on the surrogate prediction. On the other hand, LC and LOOE criteria are naturally more explorative as they place new points where the prediction error is high or where the function derivative is large.

Figure 6 shows the results of 5 iterations with each of the EI-based criteria. The EI-like criterion rapidly detects the global minimum and tends to cluster points around it, thus showing an "optimizer" behaviour. Similar results are obtained with the EI criterion, even if the final location of the global optimum is approximated. As predicted, EIGF and GEIGF provide a rather dispersed distribution of samples; however, a global optimization would take advantage of this as the region around the minimum is well captured.

4. Surrogate-Based Sequential Optimization

The workflow of the surrogate-based optimization is depicted in Figure 7. The method is built around the training database which is progressively fed and updated throughout the surrogate enrichment. Three major stages are conceived and designed, answering different needs in surrogate training and optimization: the space-filling initialization, the adaptive infill, and the sequential optimization infill. Each stage will be discussed and detailed in the following sections.

4.1. Space-Filling Initialization Stage. As a first step, the training database has to be initialized in order to build the first instance of the surrogate model. This stage is aimed at providing basic information about the objective space and selecting n_{apr} samples to maximize the informative level. This stage is usually referred to as a priori sampling because it does not require any detail about the response function. A space-filling design of the experiment technique is deemed appropriate to this aim, e.g., Latin hypercube sampling or Latinized central Voronoi tessellation techniques. Typically, according to literature results and the author's experience, the number of initial samples (n_{apr}) produced in this stage should not exceed one-third of the total computational budget. Moreover, as multiple and explorative infill criteria may be applied in the second stage, the number of initial samples has to be kept as lowest as possible. Generating multiple training samples all together has the great advantage that they can be evaluated simultaneously; thus, this stage can be executed in parallel to speed up the simulation. Once the evaluation process has finished, the selected surrogate model can be built as described in Section 2.

Table 2: Type of surrogate models.

Name	Model	Hyperparameters	Notes on training
rbf-grd	RBFN	(θ, λ)	$\varepsilon_{\mathrm{LOO}}$ minimization, grid search
rbf	RBFN	(θ, λ)	$\varepsilon_{\mathrm{LOO}}$ minimization, numerical opt.
krig-hp	Kriging	$(\theta_{p,p=1\cdots d})$	MLE maximization (equation (13)), numerical opt.
krig-all	Kriging	$(\theta_{p,p=1\cdots d}, \sigma_k, \lambda)$	MLE maximization (equation (10)), numerical opt.

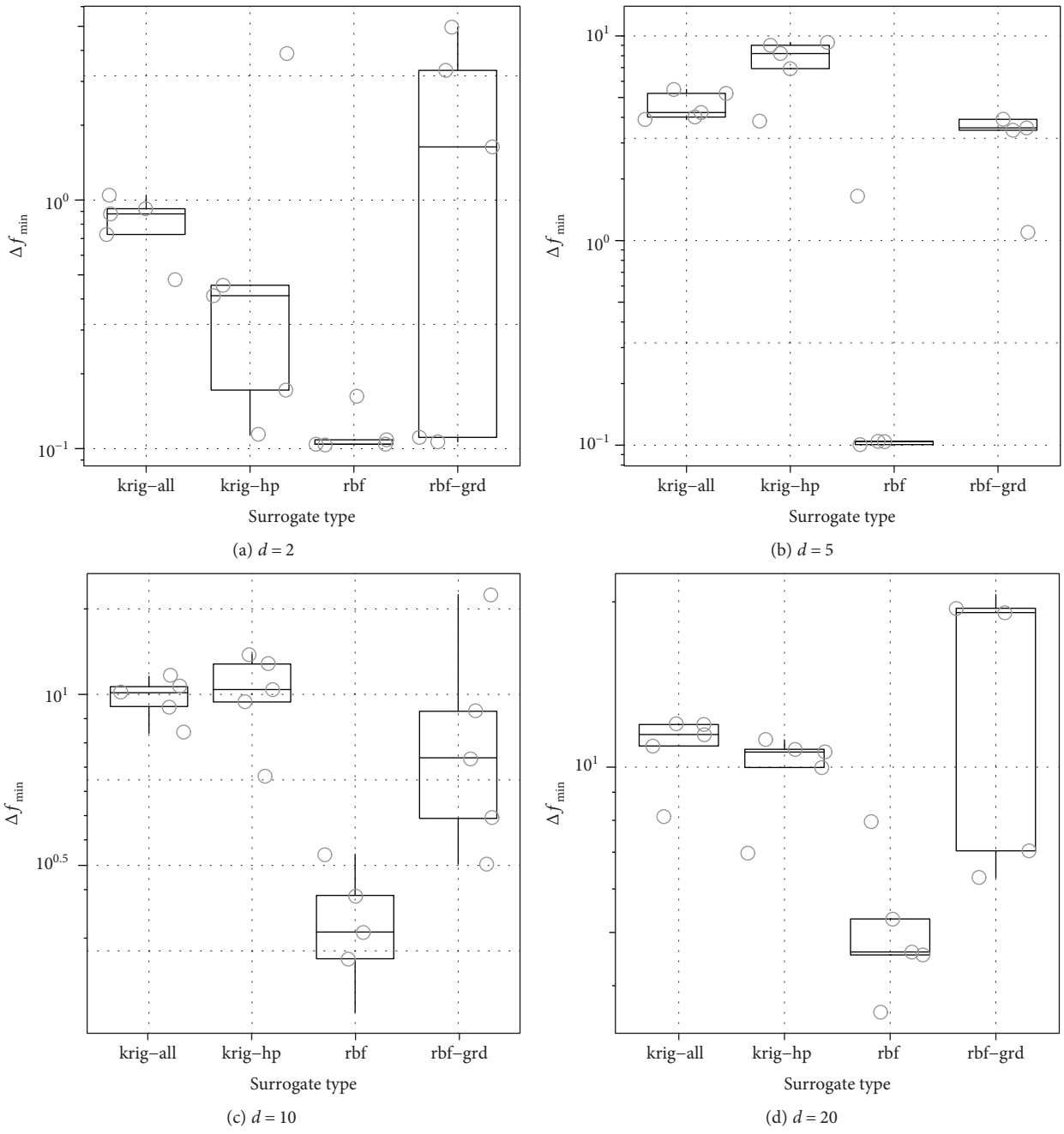


Figure 8: Boxplot of $|\Delta f|$ for the Ackley test function.

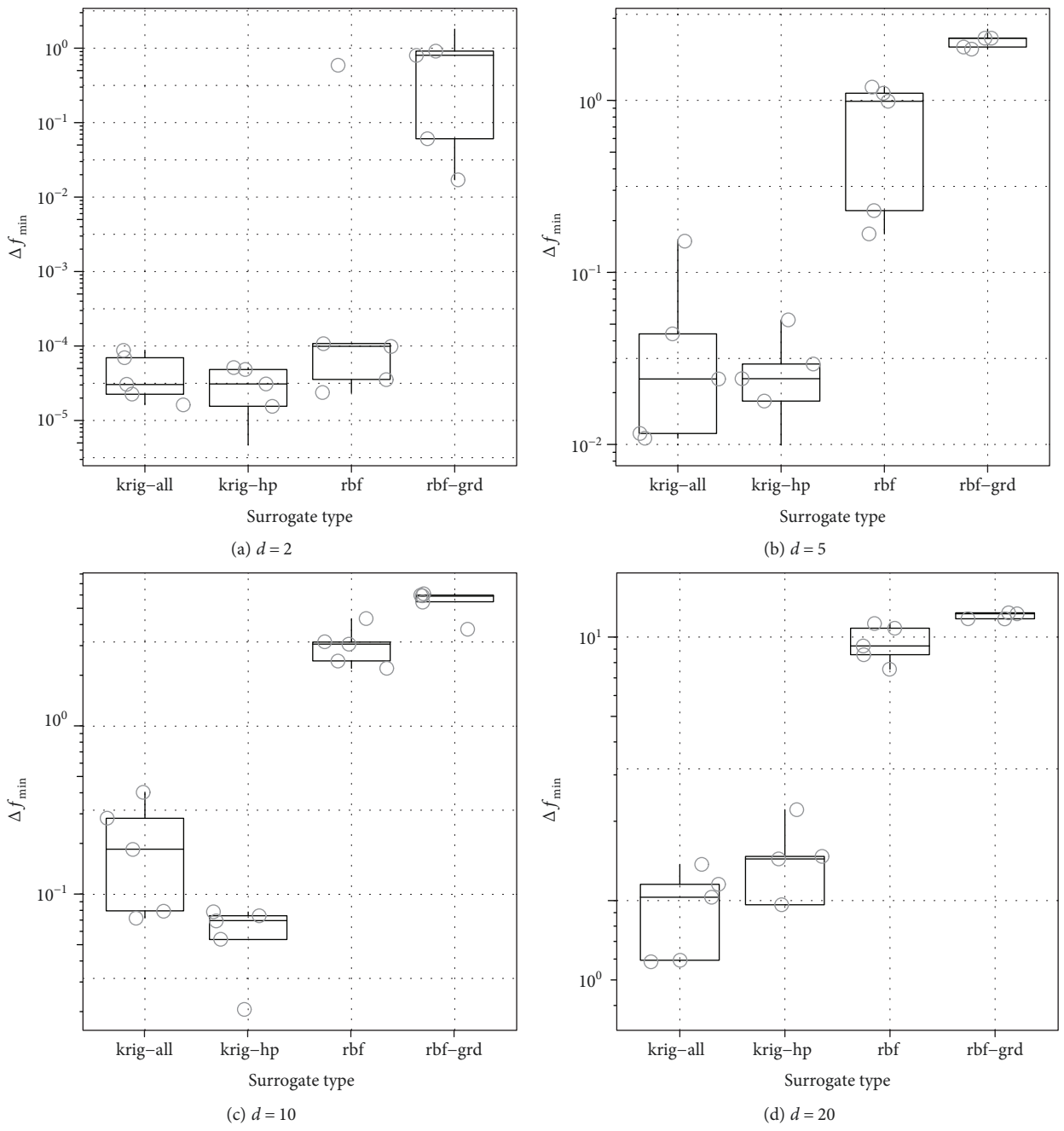


Figure 9: Boxplot of $|\Delta f|$ for the Michalewicz test function.

4.2. Design Exploration via Sequential Adaptive Sampling. The training of the initial surrogate is not driven by optimization purposes neither by considerations concerning the prediction error. Being a pure space-filling exercise, the aim is to evenly distribute the samples across the design space: as a consequence, the metamodel cannot be as accurate as required by the optimization task as (1) no control over the prediction errors has been introduced and (2) the proper identification of global minima is not investigated. The second stage of the SBO process (here referred to as "smart exploration" or sequential adaptive sampling stage) reflects the need to provide the optimizer with an improved and reliable surrogate model. The cycle iterates to update the training database with n_{adpt} new samples suggested by infill criteria as described in Section 3. The iterative procedure is structured as follows:

(1) Initialization step: the number n_{par} of new samples to be inserted at each infill iteration is defined and the infill criterion is chosen among the available ones. Two options are implemented: (1) the single predefined criterion or (2) the criterion is chosen randomly at each iteration according to a given activation probability for each criterion. The current training sample set is X_n of size $n = n_{\text{apr}}$

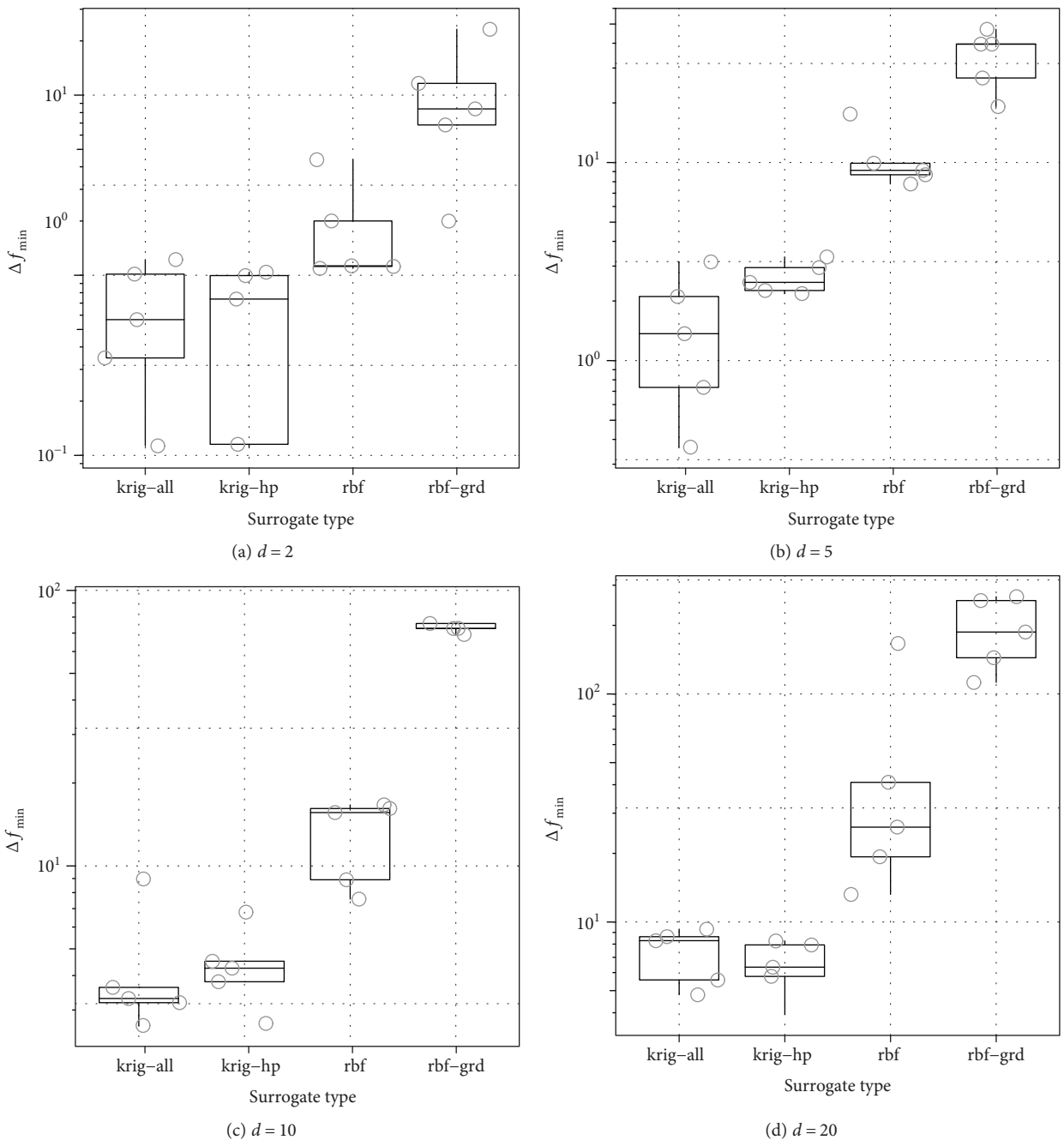


Figure 10: Boxplot of $|\Delta f|$ for the Rastrigin test function.

(2) Testing step: a huge space-filling testing dataset X_t is generated. The number of test samples is $n_t = 1000d$, with d being the dimension of the design space. The Latin hypercube is used with different seeds each time in order to avoid sample duplication issues. According to the selected criterion, the potential of improvement function v is evaluated at each point of the testing dataset and a vector V_t of size n_t is obtained

(3) Infill step: a single-point $(n_{\text{par}} = 1)$ or multipoint $(1 < n_{\text{par}} < n_{\text{adpt}} + n_{\text{apr}} - n)$ selection is available—in the first case, each infill iteration produces a single candidate to be evaluated; hence, a complete sequential infill approach is realized and the surrogate has to be built n_{adpt} times; in the second one, multiple samples are simultaneously selected, so that the true function can be evaluated in parallel before refitting the surrogate model. In the latter case, a batch sequential selection is performed. However, the two approaches pose different issues: the single-point approach gives the possibility to update the surrogate many times and to select more criteria during the sequential process, but of course it is much slower; on the other hand, by selecting the

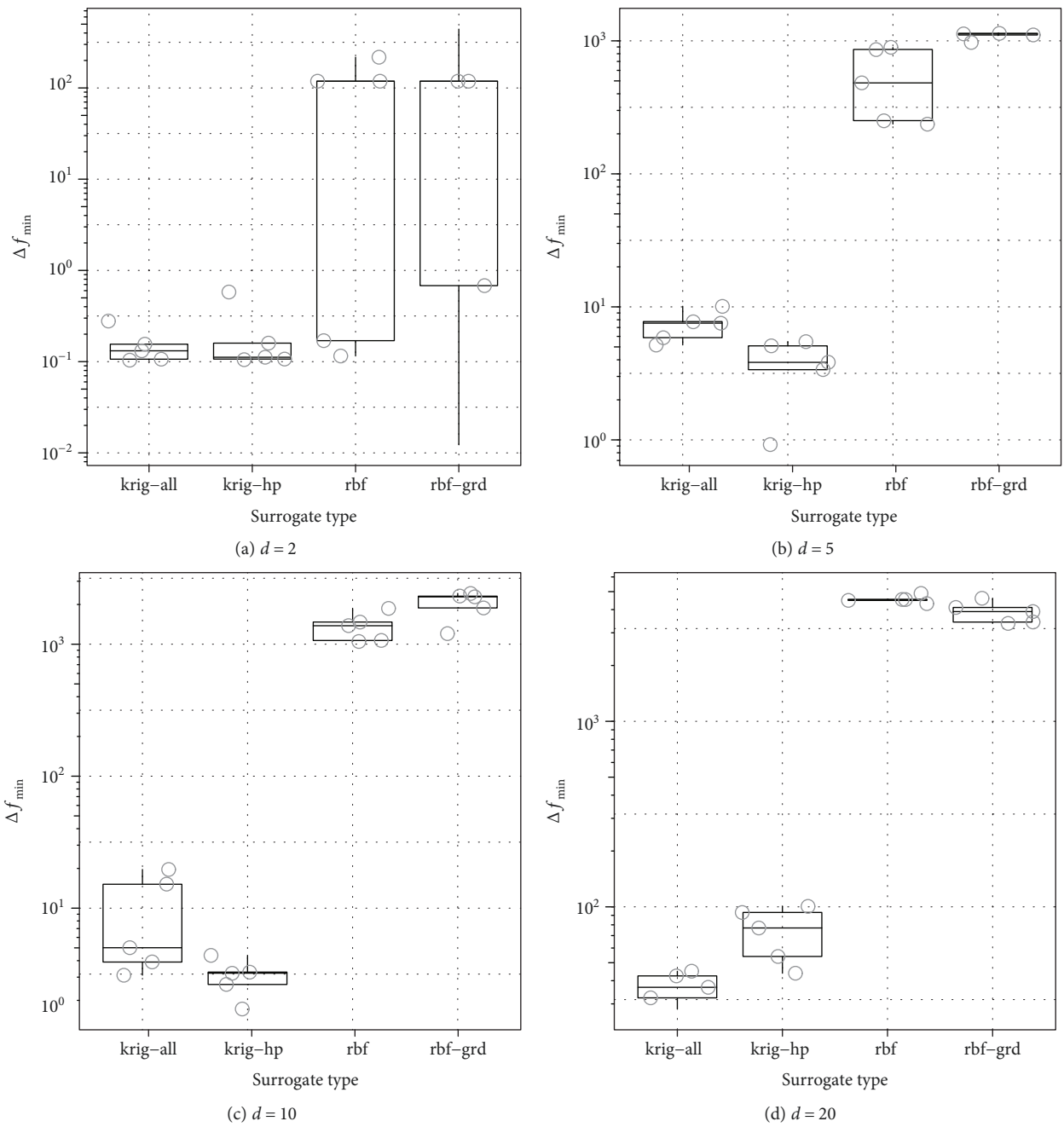


Figure 11: Boxplot of $|\Delta f|$ for the Schwefel test function.

multipoint approach, the best scoring candidates will probably lead to cluster points in a specific area with no diversity and hamper the surrogate adaptation. To overcome this issue, the following procedure is followed. The V_t vector is ranked according to the score represented by the value of the ν function. If $n_{\text{par}} = 1$, the highest scoring candidate $\mathbf{x}_h = \arg\ \max_{\mathbf{x}_i \in X_t} \nu(\mathbf{x}_i, \widehat{f}(\mathbf{x}_i), X_n, F_{X_n})$ is selected; if $n_{\text{par}} > 1$ and a multipoint selection is requested, the highest scoring candidate $\mathbf{x}_h$ is again selected and included in the batch selection set $X_h = \{\mathbf{x}_h\}$. The scores of the remaining $n_{\text{par}} - 1$ are reweighted according to the distance penalization introduced by Maljovec et al. [40]. For $\mathbf{x}_i \in X_t \setminus X_h$, the distance from the nearest training sample $\mathbf{x}_i^*$ is $d(\mathbf{x}_i, \mathbf{x}_i^*)$; the mean value of those distances over $X_t \setminus X_h$ is

$$d_0 = \frac{1}{n_t - n_h} \sum_{i=1}^{n_t - n_h} d(\mathbf{x}_i, \mathbf{x}_i^*), \tag{36}$$

where n_h is the dimension of the batch selection set (equal to 1 at this point). A reweighted score

Table 3: Summary of optimization results. The best performance for each case is highlighted in bold character.

Function	d	Surrogate	Mean $\Delta f_{\min}$ (st. deviation)
Ackley	2	rbf	**0.116533** (**0.0257383**)
Ackley	2	krig-hp	1.00747 (1.61502)
Ackley	2	krig-all	0.8108 (0.218127)
Ackley	2	rbf-grd	2.02785 (2.10997)
Ackley	5	rbf	**0.411399** (**0.690997**)
Ackley	5	krig-hp	7.45165 (2.22059)
Ackley	5	krig-all	4.56971 (0.730813)
Ackley	5	rbf-grd	4.61211 (3.76136)
Ackley	10	rbf	**2.17479** (**0.859483**)
Ackley	10	krig-hp	10.1923 (2.88222)
Ackley	10	krig-all	9.77272 (1.38084)
Ackley	10	rbf-grd	8.51475 (6.56961)
Ackley	20	rbf	**5.19498** (1.65741)
Ackley	20	krig-hp	9.9218 (1.70257)
Ackley	20	krig-all	10.8946 (**1.60441**)
Ackley	20	rbf-grd	14.507 (7.185)
Michalewicz	2	rbf	0.117523 (0.262639)
Michalewicz	2	krig-hp	**3.01527*e* − 05** (**2.02407*e* − 05**)
Michalewicz	2	krig-all	4.53489*e* − 05 (3.15939*e* − 05)
Michalewicz	2	rbf-grd	0.718937 (0.731796)
Michalewicz	5	rbf	0.735592 (0.496583)
Michalewicz	5	krig-hp	**0.0268443** (**0.0163042**)
Michalewicz	5	krig-all	0.0484547 (0.0593555)
Michalewicz	5	rbf-grd	2.24357 (0.243358)
Michalewicz	10	rbf	3.04617 (0.835715)
Michalewicz	10	krig-hp	**0.0594693** (**0.023635**)
Michalewicz	10	krig-all	0.205087 (0.141132)
Michalewicz	10	rbf-grd	5.4596 (0.98433)
Michalewicz	20	rbf	9.48643 (1.54065)
Michalewicz	20	krig-hp	1.29464 (0.676393)
Michalewicz	20	krig-all	**0.946973** (**0.348265**)
Michalewicz	20	rbf-grd	12.5354 (1.19365)
Rastrigin	2	rbf	1.94421 (1.41502)
Rastrigin	2	krig-hp	**0.599841** (**0.459052**)
Rastrigin	2	krig-all	0.6528 (0.460209)
Rastrigin	2	rbf-grd	10.4104 (7.95429)
Rastrigin	5	rbf	10.615 (3.95991)
Rastrigin	5	krig-hp	2.64103 (**0.490439**)
Rastrigin	5	krig-all	**1.54429** (1.11499)
Rastrigin	5	rbf-grd	34.425 (11.2453)
Rastrigin	10	rbf	12.9831 (4.36657)
Rastrigin	10	krig-hp	4.40714 (**1.50647**)
Rastrigin	10	krig-all	**4.34432** (2.61069)
Rastrigin	10	rbf-grd	75.4175 (6.69515)
Rastrigin	20	rbf	53.2498 (64.2505)
Rastrigin	20	krig-hp	**6.44738** (**1.75453**)
Rastrigin	20	krig-all	7.31605 (1.99999)
Rastrigin	20	rbf-grd	193.518 (68.0271)
Schwefel	2	rbf	90.8759 (92.1036)
Schwefel	2	krig-hp	0.212514 (0.206381)
Schwefel	2	krig-all	**0.155094** (**0.0721013**)
Schwefel	2	rbf-grd	135.79 (180.733)
Schwefel	5	rbf	544.699 (318.065)
Schwefel	5	krig-hp	**3.73618** (**1.79745**)
Schwefel	5	krig-all	7.2895 (1.91919)
Schwefel	5	rbf-grd	1114.49 (91.4709)
Schwefel	10	rbf	1367.87 (338.242)
Schwefel	10	krig-hp	**3.04829** (**0.975737**)
Schwefel	10	krig-all	9.36727 (7.5394)
Schwefel	10	rbf-grd	2022.02 (500.381)
Schwefel	20	rbf	4549.91 (211.859)
Schwefel	20	krig-hp	73.7412 (24.4627)
Schwefel	20	krig-all	**36.9454** (**6.95312**)
Schwefel	20	rbf-grd	3883.12 (511.637)

function v^* is then assigned to each test sample $\mathbf{x}_i$ and defined by using a penalization term ρ:

$$v^* = v\rho \begin{cases} \rho = 1, & \text{if } d(\mathbf{x}_i, \mathbf{x}_i^*) > d_0, \\ \rho = 1.5d(\mathbf{x}_i, \mathbf{x}_i^*) - 0.5 * d^3(\mathbf{x}_i, \mathbf{x}_i^*), & \text{if } d(\mathbf{x}_i, \mathbf{x}_i^*) \le d_0. \end{cases} \tag{37}$$

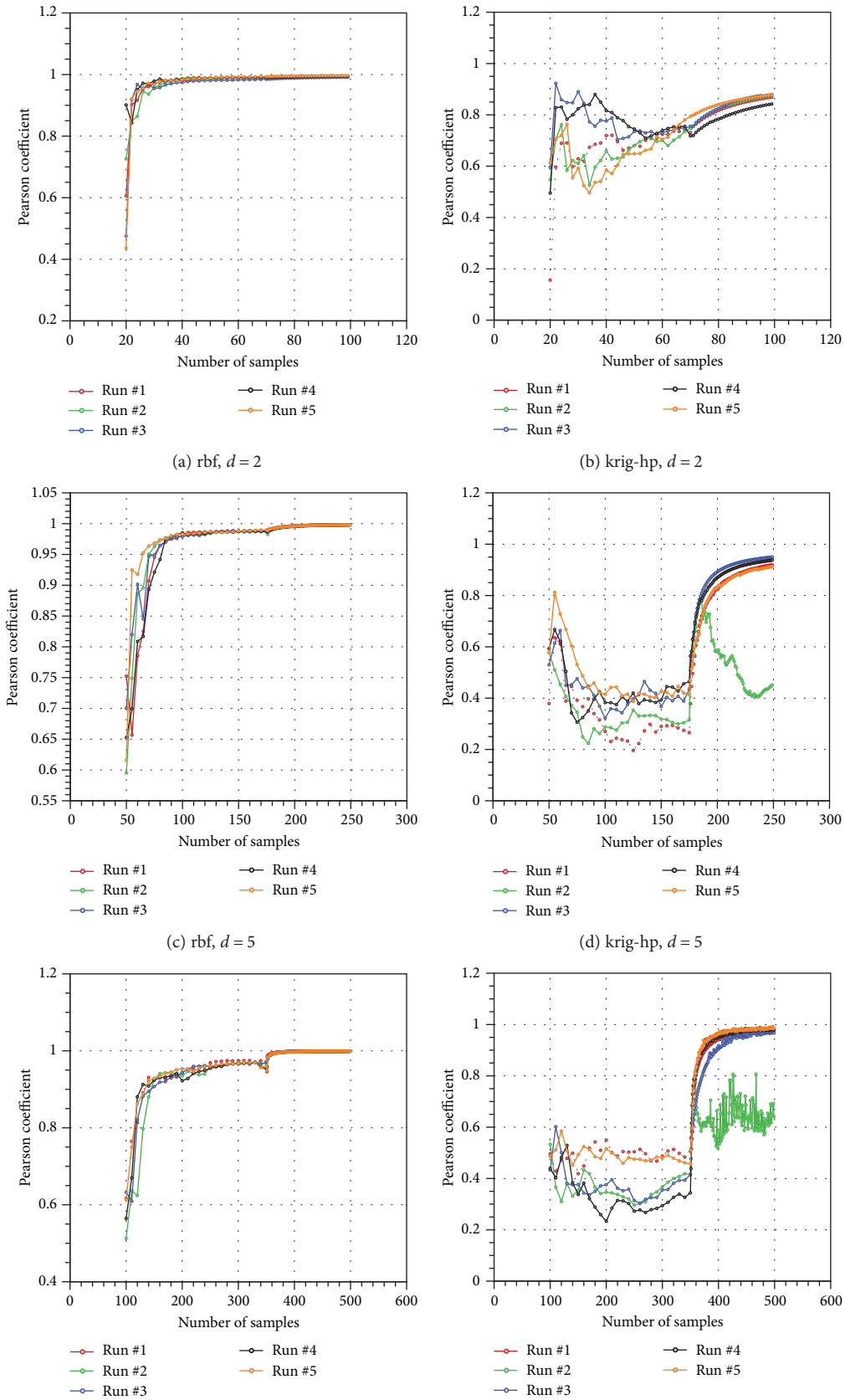
Finally, the new candidate is chosen by selecting $\mathbf{x}_{h+1} = \arg\ \max_{\mathbf{x}_i \in X_t \setminus X_n} v^*(\mathbf{x}_i, \widehat{f}(\mathbf{x}_i), X_n, F_{X_n})$. The batch selection set is now increased by the new sample $X_h = \{\mathbf{x}_h, \mathbf{x}_{h+1}\}$. By repeating the reweighting process and the highest score selection $n_{\text{par}} - 2$ more times, a set $X_h = \{\mathbf{x}_h, \mathbf{x}_{h+1}, \cdots, \mathbf{x}_{h+n_{\text{par}}-1}\}$ of n_{par} new candidates is obtained and passed to the parallel evaluation with the true model. Note that the new samples are selected by using the same surrogate prediction $\widehat{f}$ and the same training set (X_n, F_{X_n}); hence, the associated computational cost is negligible

(4) New candidate evaluation step: the parallel evaluation of the candidates in X_h is performed and the corresponding set of response values F_{X_h} is created

(5) Surrogate update step: the surrogate model is refitted over the updated training set $(X_n \cup X_h, F_{X_n} \cup F_{X_h})$, and the training set size is updated $n = n + n_{\text{par}}$

(6) Check step: if the infill computational budget is not over $(n < n_{\text{adpt}} + n_{\text{apr}})$, a new iteration is started by going back to point (2); otherwise, the sequential infill process is terminated

4.3. Sequential Metamodel Optimization. The last phase of the surrogate optimization process is devoted to the



(a) rbf, $d = 2$

(b) krig-hp, $d = 2$

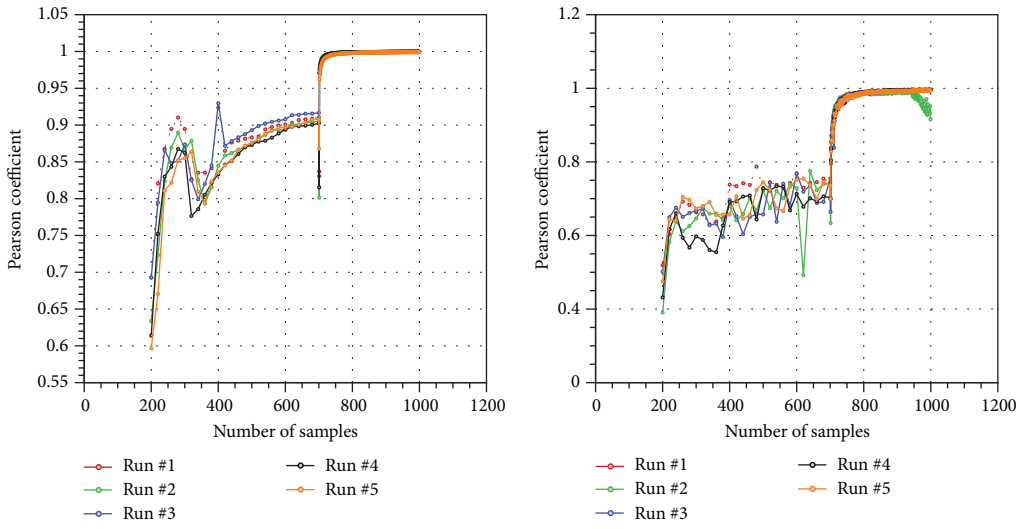
(c) rbf, $d = 5$

(d) krig-hp, $d = 5$

(e) rbf, $d = 10$

(f) krig-hp, $d = 10$

Figure 12: Continued.



(g) rbf, $d = 20$ (h) krig-hp, $d = 20$

FIGURE 12: Ackley function, progress of the Pearson correlation coefficient.

exploitation of the effort spent so far in fitting an accurate and improved metamodel. The training database of design solutions is here enriched with n_{opt} samples suggested by many sequential optimizations on the metamodel: at the end of each optimization, the best candidate is evaluated with the high-fidelity model and appended to the database for a new model fitting. This leads to the generation of a sequence of suboptimal candidates which continuously improves the objective function and approaches the design space region where the "true" optimum resides. The iterative procedure is structured as follows:

(1) Surrogate minimization step: given $\widehat{f}(\mathbf{x})$ the current metamodel built over the training set (X_n, F_{X_n}), the new suboptimal candidate is found by solving the optimization problem $\mathbf{x}_{\text{opt}} = \arg\min_{\mathbf{x}} \widehat{f}(\mathbf{x})$. The optimizer may be chosen among a wide range of options: the in-house genetic algorithm library ADGLIB [41], the CMA-ES algorithm [42], or a combination of global and local algorithms by the open-source library NLopt. Typically, as the metamodel evaluation is very quick, launching even the most computationally demanding algorithm does not represent an issue

(2) New candidate evaluation step: the new candidate $\mathbf{x}_{\text{opt}}$ is evaluated with the high-fidelity objective function, and the value of $f(\mathbf{x}_{\text{opt}})$ is obtained

(3) Surrogate update step: the surrogate model is refitted over the updated training set $(X_n \cup \{\mathbf{x}_{\text{opt}}\}, F_{X_n} \cup \{f(\mathbf{x}_{\text{opt}})\})$, and the training set size is updated to $n = n + 1$

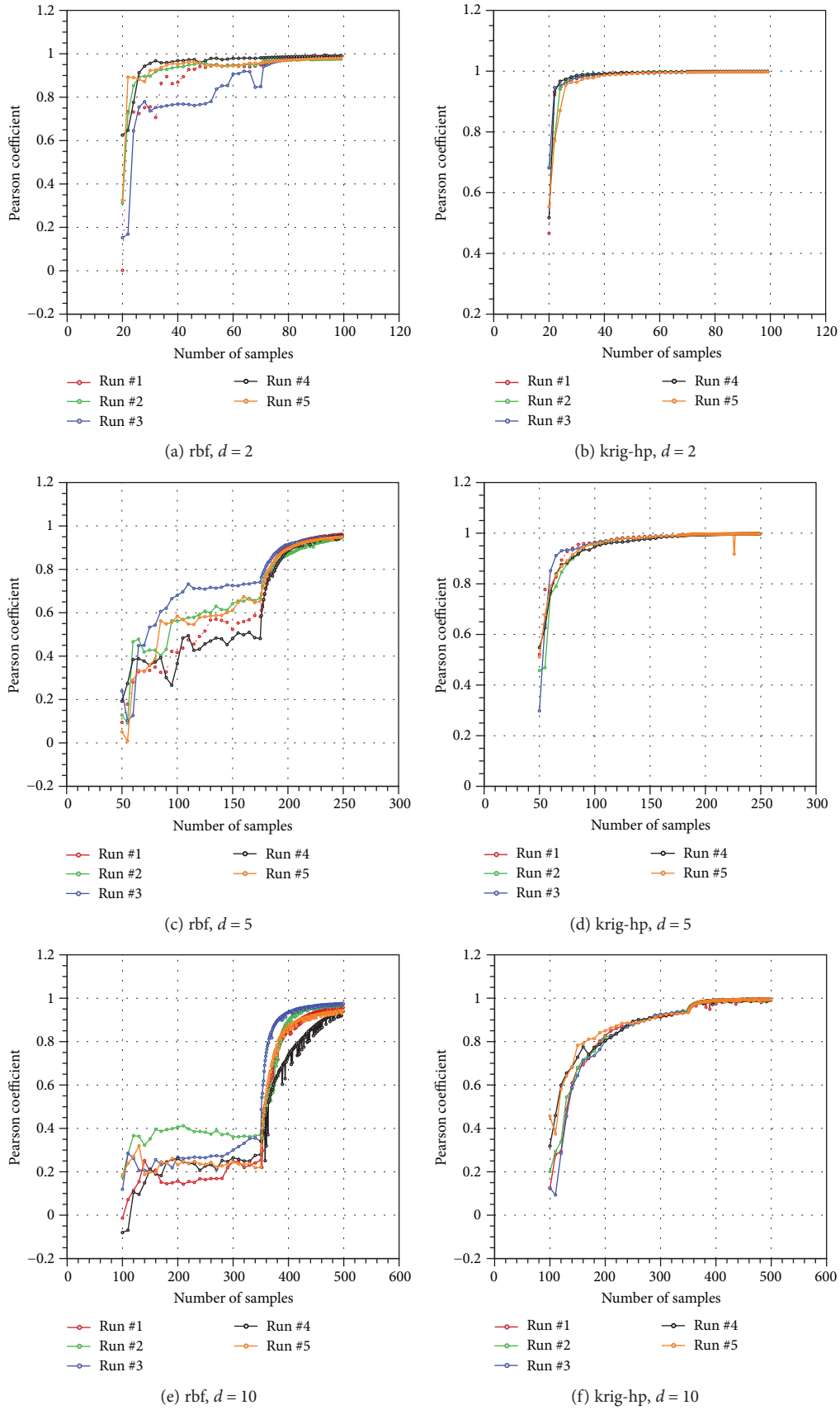
(4) Check step: if the computational budget allocated for sequential optimization is not over $(n < n_{\text{run}} = n_{\text{adpt}} + n_{\text{apr}} + n_{\text{opt}})$, a new iteration is started by going back to point (1); otherwise, the sequential optimization process is terminated

5. Experiments with Analytical Test Functions

5.1. Experimental Setup. A numerical campaign has been set up to experimentally test the optimization method. Despite the fact that the method is naturally conceived to reduce the computational load in engineering-based design cases, in the present section, representative test functions for global optimization are used to investigate the algorithm capabilities and limits. A simple and practical application is presented in the next section. Performances are evaluated in terms of $\Delta f = f_{\text{target}} - f_{\text{opt}}$, where f_{target} is the known function value at global optimum and f_{opt} is the numerical optimum function value found by the algorithm. Each run is launched at fixed computational budget n_{run} which, in turn, is a function of the design space dimension d. Recalling that n_{run} is a sum of single-stage contributions, the relation between n_{run} and d is given by $n_{\text{run}} = n_{\text{apr}} + n_{\text{adpt}} + n_{\text{opt}} = 10d + 25d + 15d = 50d$. This means that 50% of the total computational budget is devoted to the adaptive exploration: this is not surprising as goal-oriented infill criteria are used in combination with error-driven infill criteria to realize a high-level exploration-exploitation trade-off. The test functions are described in the following sections.

5.1.1. Ackley Function. The Ackley function is defined as

$$f(\mathbf{x}) = f(x_1, \cdots, x_d) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1), \tag{38}$$



(a) rbf, $d = 2$

(b) krig-hp, $d = 2$

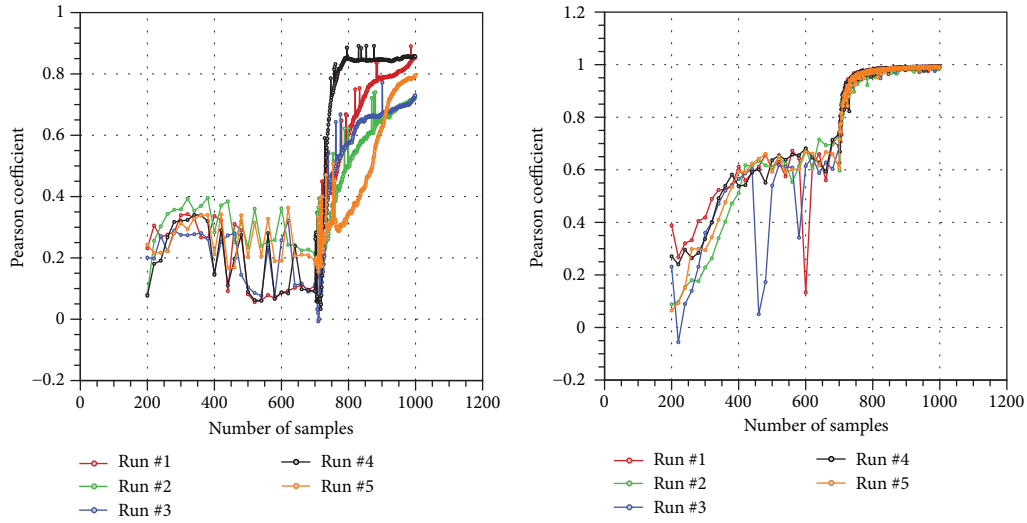
(c) rbf, $d = 5$

(d) krig-hp, $d = 5$

(e) rbf, $d = 10$

(f) krig-hp, $d = 10$

Figure 13: Continued.



(g) rbf, $d = 20$ (h) krig-hp, $d = 20$

FIGURE 13: Michalewicz function, progress of the Pearson correlation coefficient.

where $a = 20$, $b = 0.2$, and $c = 2\pi$. It is characterized by a nearly flat outer region, where many local minima are located, and a large hole at the centre. The function is continuous, nonconvex, multimodal, and scalable on the d-dimensional space. The input domain is $x_i \in [-13, 33]$ for all $i = 1, \cdots, d$. The function has one global minimum at $f(\mathbf{x}_{\text{target}}) = 0$ at $\mathbf{x}_{\text{target}} = (0, \cdots, 0)$.

5.1.2. Michalewicz Function. The Michalewicz function is defined as

$$f(\mathbf{x}) = f(x_1, \cdots, x_d) = -\sum_{i=1}^{d} \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right), \quad (39)$$

where $m = 10$. The function is featured with alternating steep valleys and ridges and has $d!$ local minima. The function is continuous, nonconvex, multimodal, and scalable on the d-dimensional space. The input domain is $x_i \in [0, \pi]$ for all $i = 1, \cdots, d$. As the global minimum and its location vary with the input dimension, Table 1 reports the corresponding values for $d = \{2, 5, 10, 20\}$.

5.1.3. Rastrigin Function. The Rastrigin function is defined as

$$f(\mathbf{x}) = f(x_1, \cdots, x_d) = 10d + \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i)\right]. \quad (40)$$

The function is continuous, convex, and scalable on the d-dimensional space, has several local minima, and is highly multimodal. However, the locations of the minima are regularly distributed. There is one global minimum $f(\mathbf{x}_{\text{target}}) = 0$ at $\mathbf{x}_{\text{target}} = (0, \cdots, 0)$. The function is evaluated on the hypercube $x_i \in [-5.12, 5.12]$ for all $i = 1, \cdots, d$.

5.1.4. Schwefel Function. The Schwefel function is defined as

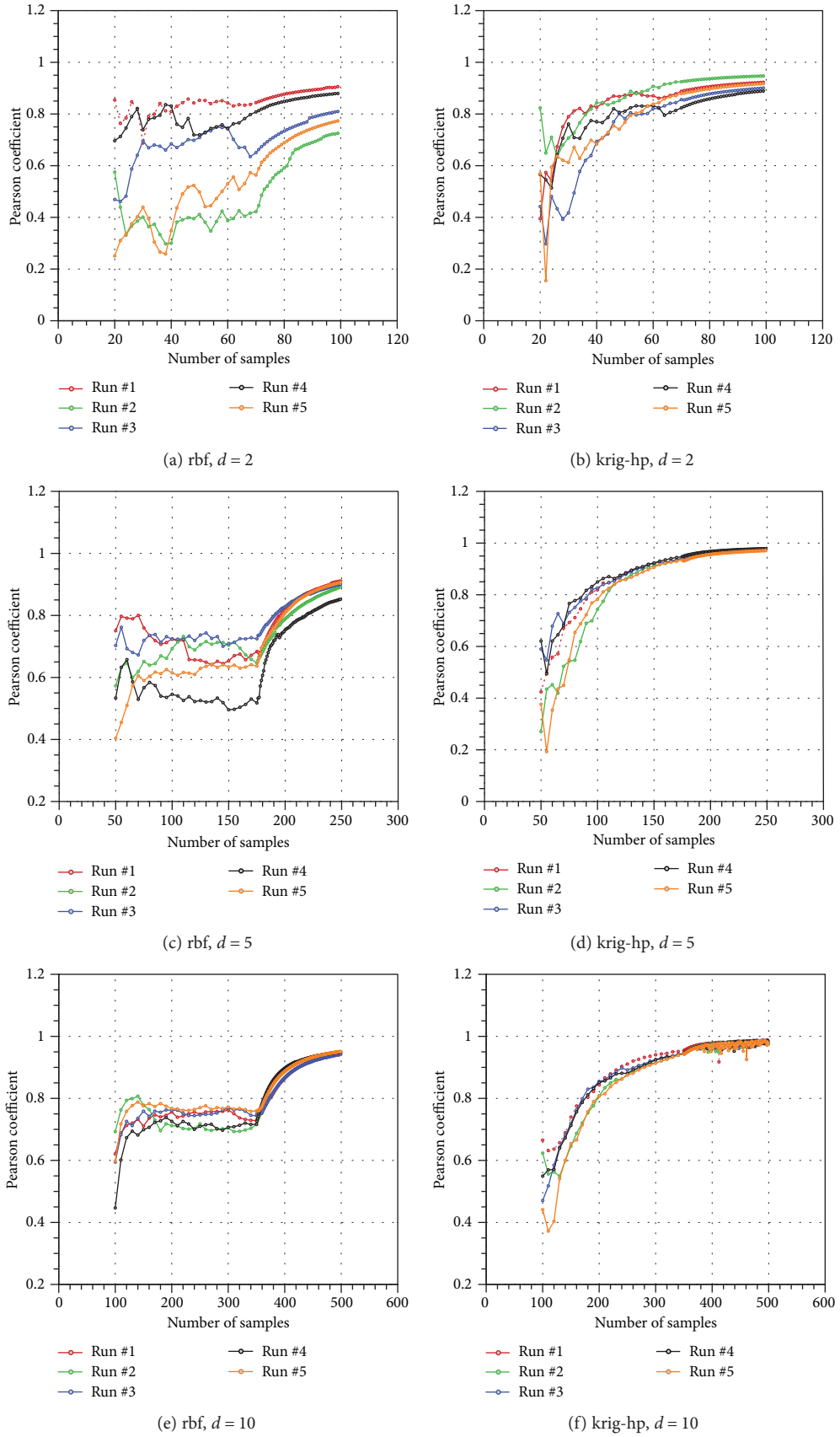
$$f(\mathbf{x}) = f(x_1, \cdots, x_d) = 418.9829d - \sum_{i=1}^{d} x_i \sin\left(\sqrt{|x_i|}\right). \quad (41)$$

The function is continuous, nonconvex, multimodal, and scalable on the d-dimensional space. Many local minima are irregularly distributed in the input space, and one global minimum is present. Moreover, with respect to the global minimum, the nearest local minimum in the objective space is the farthest in the input space. These features make the Schwefel function very hard to solve by an approximated approach. The function is evaluated in the hypercube $x_i \in [-500, 500]$ for all $i = 1, \cdots, d$, and the global minimum is $f(\mathbf{x}_{\text{target}}) = 0$ at $\mathbf{x}_{\text{target}} = (420.9687, \cdots, 420.9687)$.

5.1.5. Details of Surrogate Models under Testing. Despite the fact that the surrogate models are two (RBFN and Kriging model), some differences may arise according to the method chosen for training, i.e., for selecting good values for the hyperparameters. Table 2 shows four types of surrogates that have been used for testing purposes. The search for the hyperparameters must have a global character as often the associated cost function may have more than one extremum and being trapped in a local minimum/maximum may significantly impact the model accuracy. The hyperparameter bounds are fixed as follows:

(i) For RBFN models: $(\theta, \lambda) \in [0.01r_{\min}, 1.5r_{\max}] \times [10^{-5}, 10]$ with $r_{\min}$ and $r_{\max}$ being the minimum and maximum Euclidean distance, respectively, between training samples

(ii) For Kriging models: $(\theta_p, \sigma_k, \lambda) \in [0.001r_{\text{mean}}, 1000r_{\text{mean}}]^d \times [10^{-5}, 10^5] \times [10^{-10}, 10]$, with r_{mean}



(a) rbf, $d = 2$

(b) krig-hp, $d = 2$

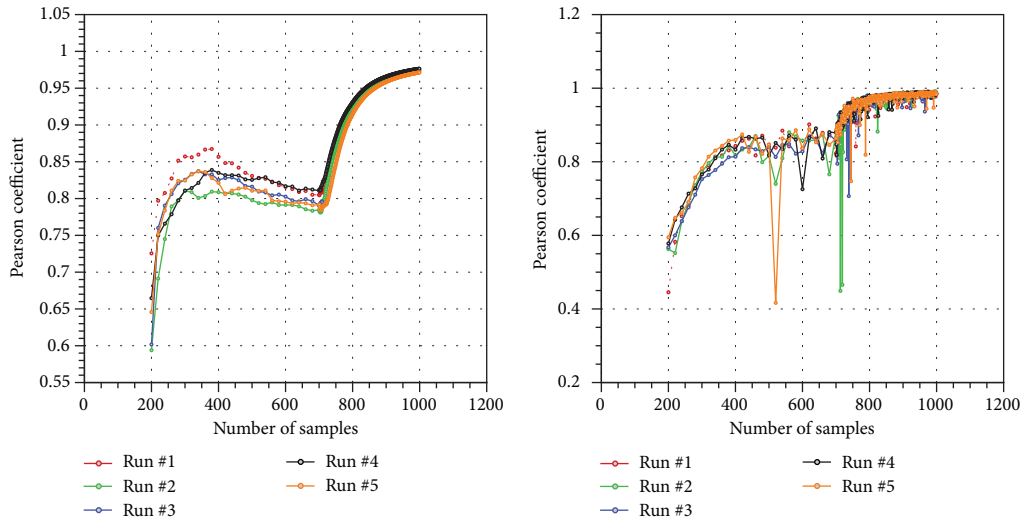
(c) rbf, $d = 5$

(d) krig-hp, $d = 5$

(e) rbf, $d = 10$

(f) krig-hp, $d = 10$

Figure 14: Continued.



(g) rbf, $d = 20$ (h) krig-hp, $d = 20$

Figure 14: Rastrigin function, progress of the Pearson correlation coefficient.

being the average Euclidean distance between training samples

5.2. Results and Discussion. As the surrogate-based optimization package discussed so far offers several choices, this section is dedicated to the investigation of the effect of selecting among the available alternatives. In particular, the influence of the type of surrogate model, infill criteria, and computational budget allocation will be studied and discussed.

5.2.1. Influence of Surrogate Model Selection. As mentioned in Section 5.1.5, four types of surrogates can be trained according to the mathematical nature (Gaussian or RBF) and optimal hyperparameter selection. All models have undergone a first experimental campaign to underline differences and draw up a ranking. The EI, EI-like, and WLOOE criteria are activated with probabilities of 50%, 30%, and 20%, respectively.

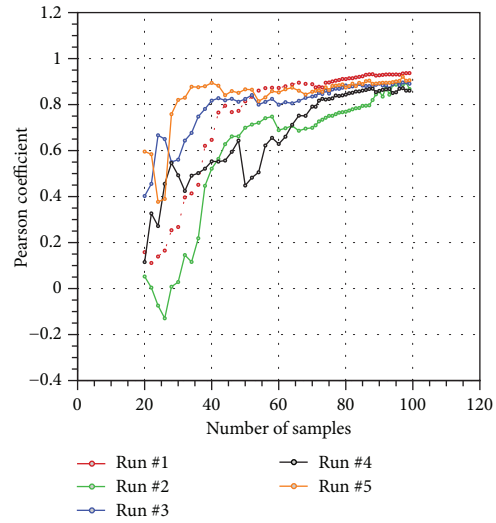
Figures 8–11 show the boxplot distribution of $\Delta f_{\min}$ over 5 repetitions for each surrogate and dimension. Outliers are highlighted by grey-filled circles, and all repeated points are plotted as small grey circles. The Gaussian-based models clearly outperform in average the RBFN except for the Ackley function. Table 3 reports a summary of optimization results, highlighting in bold character the best performing model for each case. By taking into account also the standard deviation of $\Delta f_{\min}$ for ranking purposes, Gaussian-based approaches seem to be more robust in that they offer better results with less variability. For RBFN, optimizing the hyperparameters instead of grid searching is by far the preferred approach. For Kriging, there is no clear evidence about the best training strategy as krig-hp and krig-all show similar performances and trends.

In order to provide a more comprehensive insight into the optimization data, Figures 12–15 depict the evolution of the Pearson correlation coefficient R for all cases and rbf and krig-hp models. In the present case, it is used to compute the correlation between n samples of the "true" objective function $(f_i,\ i = 1, \cdots, n)$ and the leave-one-out prediction by the surrogate $(\widehat{f}_{-1},\ i = 1, \cdots, n)$. It is defined as

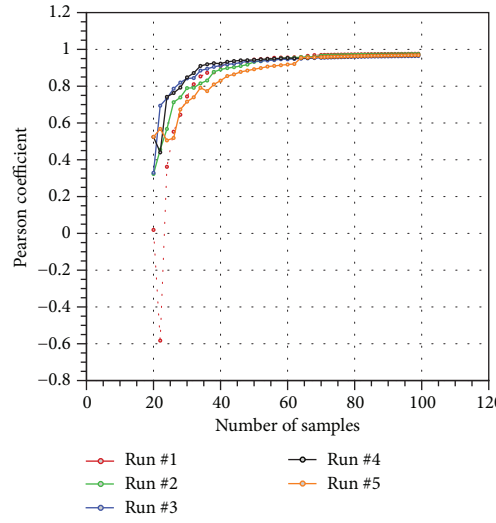
$$R = \frac{n\Sigma f_i \widehat{f}_{-i} - \Sigma f_i \Sigma \widehat{f}_{-i}}{\sqrt{n\Sigma f_i^2 - (\Sigma f_i)^2}\sqrt{n\Sigma \widehat{f}_{-i}^2 - \left(\Sigma \widehat{f}_{-i}\right)^2}}. \tag{42}$$

Each plot does not start from zero because the first training of the surrogate is done after those n_{apr} samples have been computed. It can be observed that all simulations start with low correlation between data and surrogate prediction. This is much more evident for high-dimensional function cases, where in some cases negative correlation is even found. As a general consideration, all methods achieve to reach very high correlation at the end of the optimization process, but with different paths. In particular, Gaussian-based approaches present a constant and continuous increase of R throughout both the adaptive sampling phase and the sequential optimization process. On the other hand, RBF methods tend to stagnate during the infill phase while the correlation gets better with the addition of the last n_{opt} points. This would seemingly suggest that infill criteria are not effective in reducing the prediction error of RBF methods. However, it should be also considered that trends are quite the opposite if considering only the Ackley function; hence, the test function characteristics play a major role. Hence, it derives that a more extended suite of test functions should be considered for further investigation.

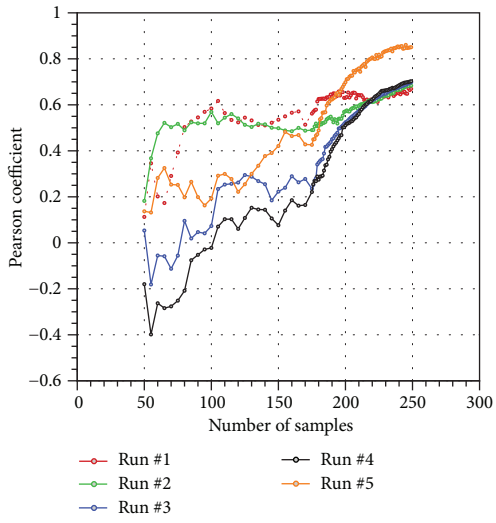
5.2.2. Influence of the Infill Criterion. This section is devoted to testing different infill criteria and combination of them with the aim of assessing which infill solution is more capable



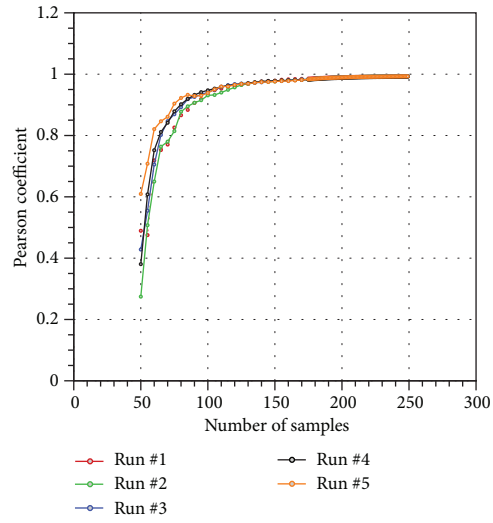
(a) rbf, $d = 2$



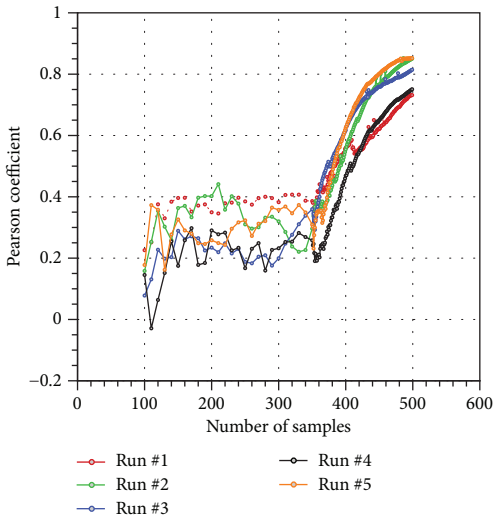
(b) krig-hp, $d = 2$



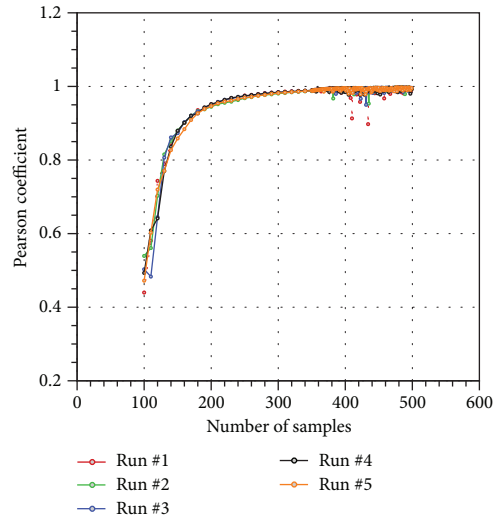
(c) rbf, $d = 5$



(d) krig-hp, $d = 5$

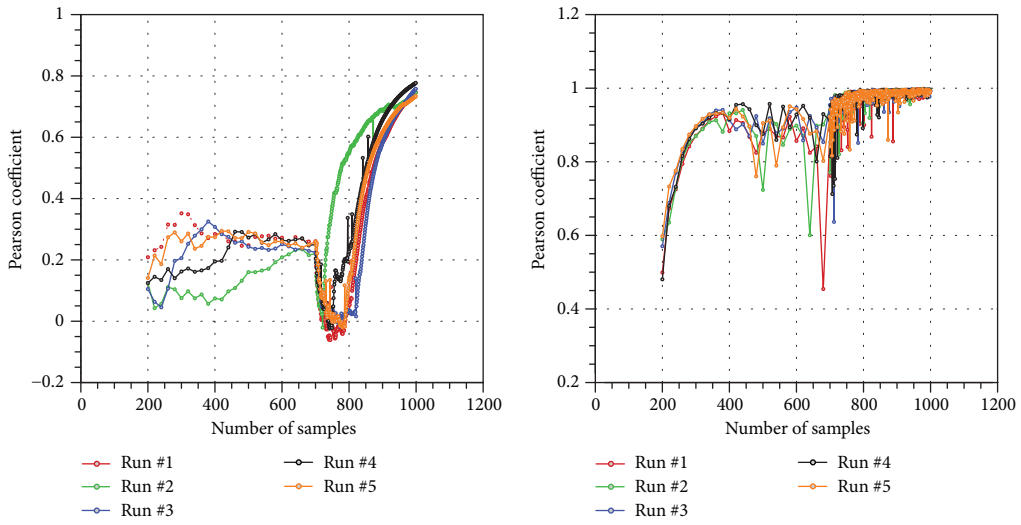


(e) rbf, $d = 10$



(f) krig-hp, $d = 10$

Figure 15: Continued.

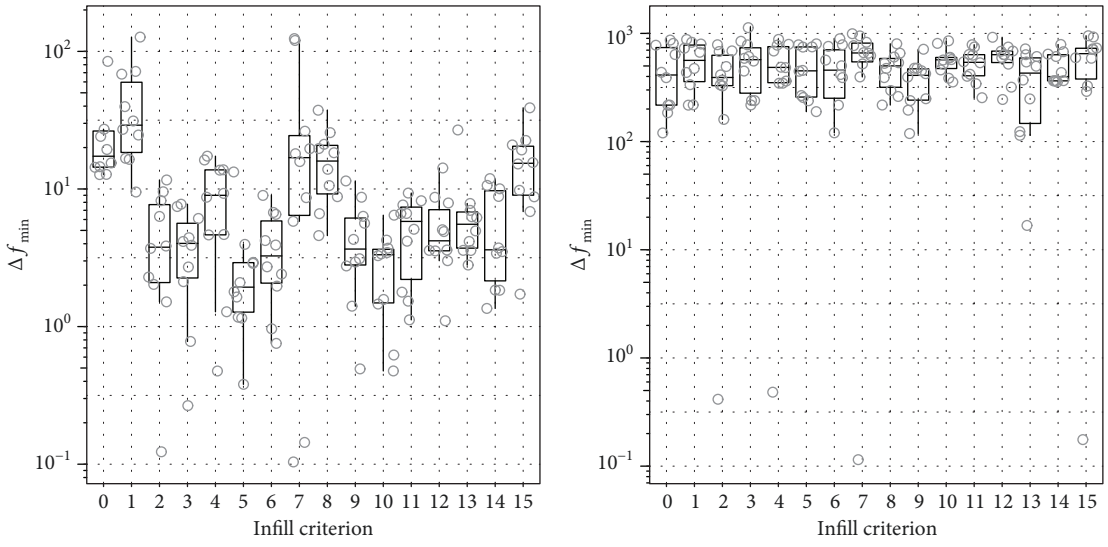


(g) rbf, $d = 20$ (h) krig-hp, $d = 20$

Figure 15: Schwefel function, progress of the Pearson correlation coefficient.

Table 4: Infill strategies.

	Strategy ID															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Criterion	Activation probability															
LC	1.0	—	—	—	—	—	—	—	—	—	—	—	0.3	0.5	—	—
LOOE	—	1.0	—	—	—	—	—	—	—	—	0.3	—	—	—	0.5	—
WLOOE	—	—	1.0	—	—	—	—	—	—	—	—	0.3	—	—	—	—
WD	—	—	—	1.0	—	—	—	—	—	—	—	—	—	—	0.5	—
FMIN	—	—	—	—	1.0	—	—	—	—	—	—	—	—	—	—	0.5
EIL	—	—	—	—	—	1.0	—	—	—	—	—	0.7	0.7	—	—	—
EI	—	—	—	—	—	—	1.0	—	—	0.7	0.7	—	—	0.5	—	—
EIGF	—	—	—	—	—	—	—	1.0	—	0.3	—	—	—	—	—	—
GEIGF	—	—	—	—	—	—	—	—	1.0	—	—	—	—	—	—	0.5



(a) krig-hp model (b) rbf model

Figure 16: Boxplot of $|\Delta f|$ for the Schwefel function ($d = 5$) with infill strategies defined in Table 4.

to provide an improved surrogate function from the optimizer's point of view. In order to ease the understanding of the results, two models (krig-hp and rbf) and a single test function (Schwefel function with $d = 5$) have been used. 16 infill strategies have been defined, and Table 4 summarizes all of them. The first 9 strategies exploit each single criterion without combination. On the other hand, the last 7 strategies have been designed by coupling a factorization-based criterion with an expected improvement-based one. 10 repetitions have been carried out for each infill strategy in order to extract reliable mean values. The optimization setup is identical to that in the previous section. Results are shown in Figure 16 as boxplot distribution and reported in Table 5 in terms of mean and standard deviation of $\Delta f_{\min}$ (over 10 repetitions). By considering both mean values and outliers, the best strategies are 3, 5, 6, 9, 10, and 11 for the Kriging model and 0, 2, 6, 8, 9, and 13 for the RBFN model. Thus, as a general consideration, Gaussian-based models benefit from a blend of expected improvement-based criteria (EI, EI-like, EIGF) and error-driven ones (mainly LOOE and WLOOE). Conversely, RBFN models work well with global fit-oriented criteria (EIGF, GEIGF, and LC) and pure EI search.

5.2.3. Influence of the Computational Budget Allocation. In this section, the influence of the optimization setup is investigated in terms of computational budget allocation to the infill and sequential optimization phases, having fixed the total budget ($n_{\text{run}} = 50d$). Table 6 reports the base setup used so far as setup no. 1, while setup no. 2 refers to the new one to be studied. The main difference is found in a more extended search with adaptive sampling to the detriment of the optimization iterations which are reduced. On the other hand, a more pronounced emphasis is put on infill criteria (EI and EI-like) specifically devoted to global optimization purposes. Results obtained with the new setup are depicted in Figures 17–20. All test functions have been considered, but only the best models (namely, krig-hp and rbf) from previous investigations have been used. The pictures are the analog of Figures 8–11 and must be compared with them in order to draw conclusions. In particular, a general worsening of the algorithm performance is highlighted by employing the new setup. Table 7 shows the new results and compares them with setup 1. The bold character is used to underline improvements with respect to setup 1. Setup 2 is beneficial in reducing the standard deviation in several cases and in average offers meaningful improvement only for the Rastrigin function. This observation confirms that results may vary significantly depending on the selection of the set of test functions as well as on the methodology and setup.

Table 5: Summary of infill screening results. The best 5 results for mean and standard deviation of $\Delta f_{\min}$ are highlighted in bold character.

Model	Strategy ID	Mean (st. deviation)	Ranking
krig-hp	0	37.3834 (44.7018)	15
krig-hp	1	43.2214 (36.1784)	16
krig-hp	2	4.92186 (3.8269)	6
krig-hp	3	**3.9667** (**2.58573**)	4
krig-hp	4	9.0102 (6.11787)	11
krig-hp	5	**3.12821** (3.71955)	2
krig-hp	6	**3.9207** (**2.72528**)	3
krig-hp	7	33.8321 (47.1973)	14
krig-hp	8	16.6604 (9.96748)	13
krig-hp	9	**4.71286** (**3.38458**)	5
krig-hp	10	**2.86443** (**1.84513**)	1
krig-hp	11	5.21631 (**2.96662**)	7
krig-hp	12	5.5616 (3.78726)	8
krig-hp	13	7.32145 (7.05348)	10
krig-hp	14	5.70416 (4.13261)	9
krig-hp	15	15.937 (10.4347)	12
rbf	0	**465.682** (285.574)	5
rbf	1	558.279 (253.342)	11
rbf	2	**425.5** (245.187)	3
rbf	3	565.885 (297.147)	12
rbf	4	516.356 (270.008)	9
rbf	5	493.261 (253.584)	7
rbf	6	**464.977** (295.829)	4
rbf	7	639.904 (289.831)	16
rbf	8	476.606 (**186.693**)	6
rbf	9	**373.97** (**175.277**)	1
rbf	10	571.658 (**161.376**)	13
rbf	11	529.976 (**174.897**)	10
rbf	12	598.873 (198.174)	15
rbf	13	**388.035** (249.365)	2
rbf	14	496.935 (**165.558**)	8
rbf	15	583.905 (298.24)	14

6. Aerodynamic Shape Optimization Problem

In order to test the present approach in a real-world and representative case, a benchmark problem has been selected from those proposed within the AIAA Design Optimization Discussion Group (ADODG), namely, the RAE 2822 airfoil optimization case. This section proposes a critical analysis of the results obtained and represents a sort of prosecution of a previous work [31] which will be taken here as a reference.

6.1. RAE 2822 Airfoil Shape Optimization. The shape optimization problem is formulated as a drag (total drag coefficient C_d) reduction task while keeping the lift level (lift coefficient C_l), trim control of the pitching moment coefficient C_m, and minimum airfoil area constant. It reads as follows:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & C_d(\mathbf{x}) \\ \text{subject to} \quad & C_l(\mathbf{x}) = C_{l,\text{base}} = 0.824 \\ & C_m(\mathbf{x}) \geq C_{m,\text{base}} = -0.092 \\ & A(\mathbf{x}) \geq A_{\text{base}} = 0.7787 m^2. \end{aligned} \tag{43}$$

Table 6: Optimization setup.

Setup ID	n_{apr}	$n_{adpt}(n_{par})$	n_{opt}	Criteria (% act. prob.)
1	$10d$	$25d$ (d)	$15d$	EI (50%) + EI-like (30%) + WLOOE (20%)
2	$10d$	$35d$ ($2d$)	$5d$	EI (30%) + EI-like (70%)

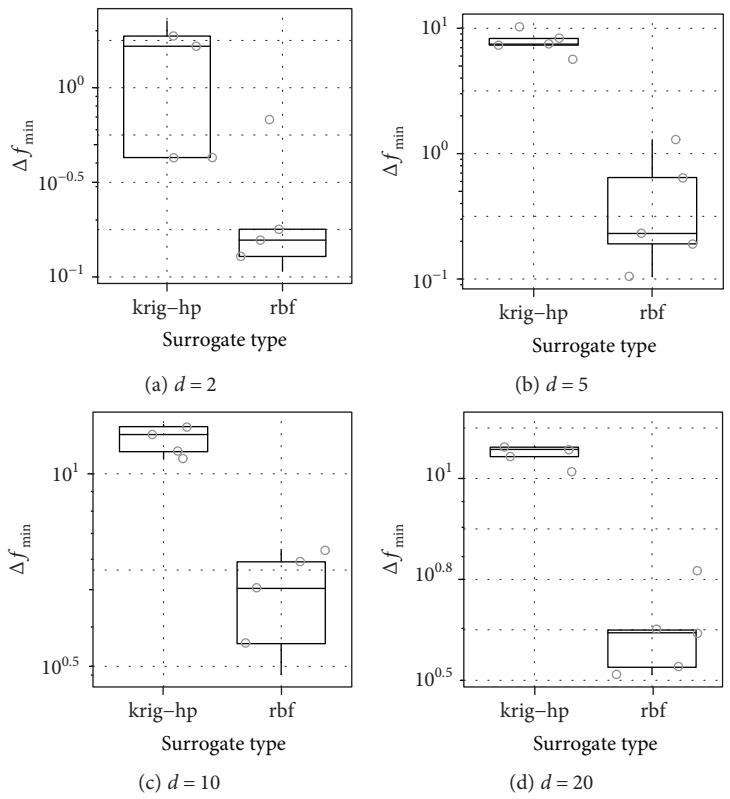


Figure 17: Boxplot of $|\Delta f|$ for the Ackley test function with setup 2.

where $\mathbf{x}$ is the generic design vector representing an airfoil shape, $A(\mathbf{x})$ is the total area enclosed by the airfoil, and A_{base} is the corresponding value for the baseline RAE 2822 airfoil. The lift constraint is explicitly satisfied by performing the flow simulation at fixed lift by varying the angle of attack. The pitching moment and the geometric constraint are treated by using a penalization approach; hence, the objective function is defined as

$$f(\mathbf{x}) = \frac{1}{C_{d,0}}\left[C_d(\mathbf{x}) + 0.2\max\left(0, C_{m,base} - C_m(\mathbf{x})\right) + 0.1\max\left(0, C_{l,base} - C_l(\mathbf{x})\right) + \max\left(0, A_{base} - A(\mathbf{x})\right)\right], \quad (44)$$

with $C_{d,0}$ being the drag coefficient of the RAE 2822 airfoil. According to this position, the baseline airfoil has a unit objective function value ($f(\mathbf{x}_{RAE}) = 1$), while feasible design solutions "better" than the baseline shape will be rewarded with $f(\mathbf{x}) < 1$. A unit airfoil chord is assumed, the pitching moment is evaluated at the quarter-chord, the Mach number is 0.734, and the Reynolds number is 6.5×10^6.

6.1.1. Parameterization. In the reference paper [31], the Class-Shape Transformation (CST) by Kulfan [43] was used to make the RAE 2822 shape parametric using 14 design variables. In the present work, the open-source SU2 code capabilities [44–46] are exploited and the FFD (free-form deformation) approach is used to arbitrarily change the geometry and, consequently, the volume mesh. In particular, 20 FFD control points (CPs) are defined around the RAE 2822 airfoil, as depicted in Figure 21, and the vertical displacements of the control points are employed as design variables. The two CPs on the leading edge and on the trailing edge are constrained to have the same displacement, so the total number of design variables is reduced to 18.

6.1.2. Optimization Studies. Three optimization studies have already been performed employing different methods

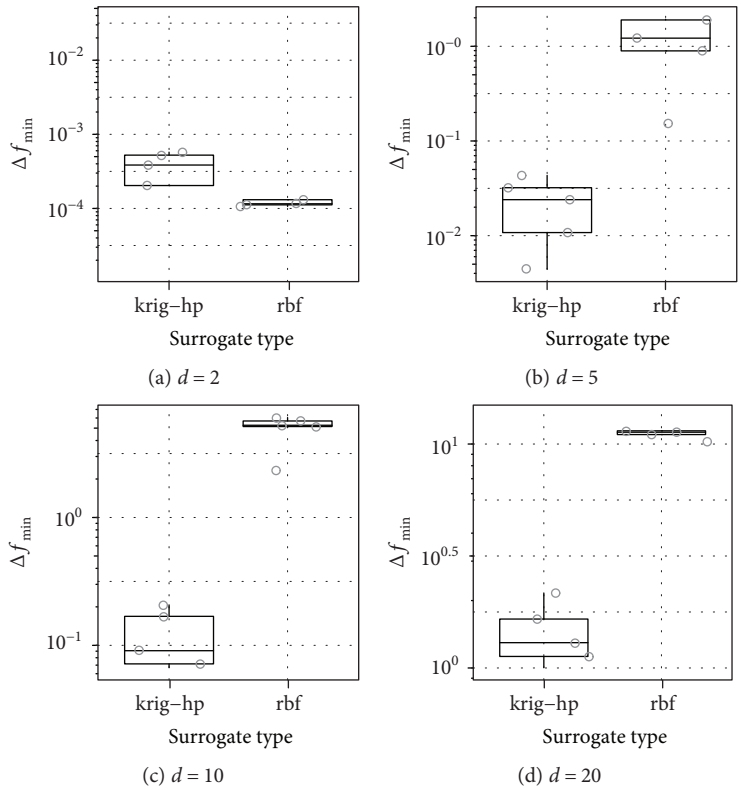


Figure 18: Boxplot of $|\Delta f|$ for the Michalewicz test function with setup 2.

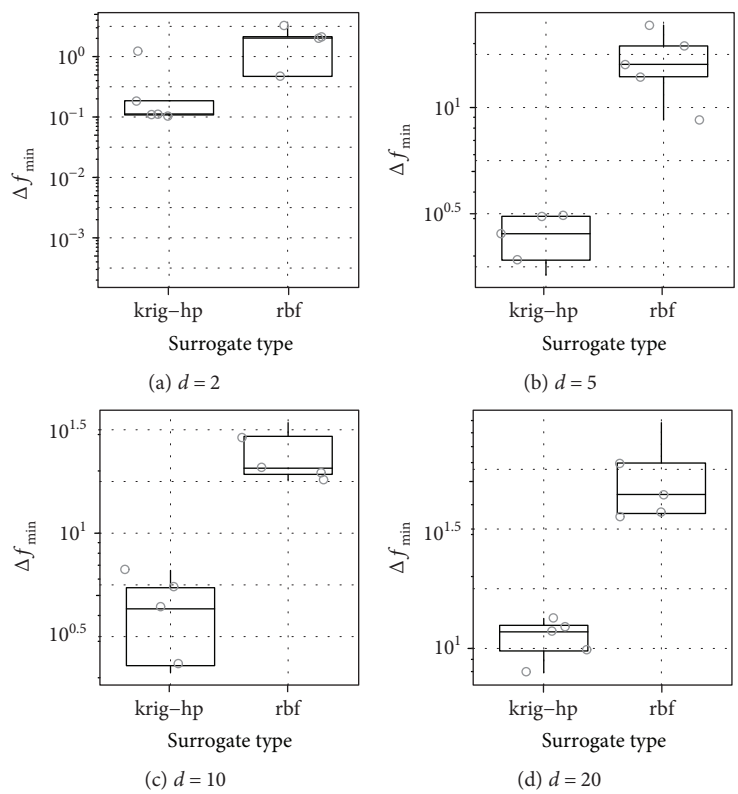


Figure 19: Boxplot of $|\Delta f|$ for the Rastrigin test function with setup 2.

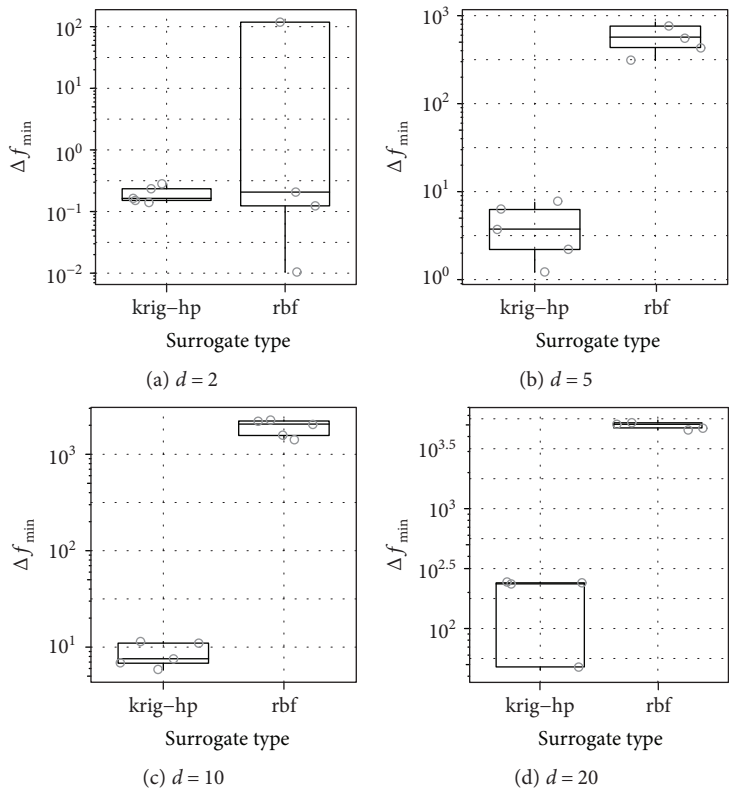


Figure 20: Boxplot of $|\Delta f|$ for the Schwefel test function with setup 2.

and computational loads [31]. Details are reported in Table 8, together with information about the present optimization studies. In all cases, RANS simulations are launched to evaluate the objective function when high fidelity is required, e.g., to validate the samples suggested by infill criteria or by optimizing on the surrogate. Practically, the CFD approach is different: reference cases employed the in-house multiblock structured ZEN code [47], the $k-\omega$ TNT turbulence model, and a fine mesh selected after a detailed mesh convergence analysis; as already mentioned, new cases take advantage of the unstructured SU2 suite running with the Spalart–Allmaras turbulence model on a coarser mesh. Hence, any comparison between cases should take into account such differences.

SBOSA (Surrogate Based Optimization with Sequential Adaptation) was similar to the present one, but considered a lower dimensional space (14 instead of 18 design variables, CST parameterization instead of FFD) and a surrogate model based on proper orthogonal decomposition (POD) and RBF interpolation of coefficients. EGO (Efficient Global Optimization) refers to the classical Kriging-based approach made popular by Jones et al. [1] and implemented within the Dakota package [48]. PGA (Plain Genetic Algorithm) consisted in a pure, intensive evolutionary optimization where no surrogate model was used and all evaluations were performed by using the high-fidelity CFD approach. Here, two further approaches are added for benchmarking: SU2AO and SU2SBO. SU2AO uses the adjoint gradient-based optimization method embedded within the SU2 code which includes the flow solution, continuous adjoint flow solution, geometry modification, mesh deformation, and gradient optimization (SLSQP method). SU2SBO, instead, involves the presented surrogate-based approach and the SU2 tools for flow solution, airfoil shape modification, and mesh deformation. Table 9 reports the setup used for this simulation which reflects the experimental knowledge gained with the test functions in the previous sections.

As for Table 8, the number of CFD evaluations needed by the SU2AO approach comprises the calls to the adjoint flow solver (one for each objective/constraint, 3 in total according to the problem statement in Section 6.1), as each additional adjoint evaluation has approximately the same cost of a single CFD solution. However, the number of gradient evaluations (and, hence, the calls to the adjoint flow solver) is smaller than CFD ones (30 out of 70) as the gradient vector is not updated at each iteration.

Figure 22 shows the progress of the minimum value of the objective function found through the sequential optimizations SU2SBO and SU2AO. A log scale for the x-axis is used to make the picture clearer. The local approach (SU2AO) is much faster to reach feasible and optimal regions of the design space; indeed, the objective function drops down to approximately 30% with respect to the baseline after only 30 design cycles. The global

Table 7: Summary of optimization results with setup 2. The best performance for each case is highlighted in bold character.

Function	d	Surrogate	Mean $\Delta f_{\min}$ (st. deviation)	Δ wrt setup 1
Ackley	2	rbf	0.249896 (0.241126)	0.133363 (0.215388)
Ackley	2	krig-hp	1.32697 (0.848681)	0.3195 (**-0.766339**)
Ackley	5	rbf	0.492401 (0.492681)	0.081002 (**-0.198316**)
Ackley	5	krig-hp	7.81666 (1.68073)	0.36501 (**-0.53986**)
Ackley	10	rbf	4.78755 (1.44069)	2.61276 (0.581207)
Ackley	10	krig-hp	12.3338 (1.11724)	2.1415 (**-1.76498**)
Ackley	20	rbf	4.96913 (0.991267)	**-0.22585** (**-0.666143**)
Ackley	20	krig-hp	11.4171 (0.906291)	1.4953 (**-0.796279**)
Michalewicz	2	rbf	0.00722741 (0.015901)	**-0.110296** (**-0.246738**)
Michalewicz	2	krig-hp	0.000340437 (0.000231793)	0.000310284 (0.000211552)
Michalewicz	5	rbf	1.21836 (0.74276)	0.482768 (0.246177)
Michalewicz	5	krig-hp	0.0229353 (0.0157477)	**-0.003909** (**-0.0005565**)
Michalewicz	10	rbf	4.89782 (1.46853)	1.85165 (0.632815)
Michalewicz	10	krig-hp	0.121063 (0.0631468)	0.0615937 (0.0395118)
Michalewicz	20	rbf	11.4368 (1.0426)	1.95037 (**-0.49805**)
Michalewicz	20	krig-hp	1.44886 (0.467794)	0.15422 (**-0.208599**)
Rastrigin	2	rbf	1.58798 (1.34542)	**-0.35623** (**-0.0696**)
Rastrigin	2	krig-hp	0.346518 (0.490664)	**-0.253323** (0.031612)
Rastrigin	5	rbf	16.5176 (5.89206)	5.9026 (1.93215)
Rastrigin	5	krig-hp	2.45519 (0.672031)	**-0.18584** (0.181592)
Rastrigin	10	rbf	24.3131 (7.10916)	11.33 (2.74259)
Rastrigin	10	krig-hp	4.15407 (1.95796)	**-0.25307** (0.45149)
Rastrigin	20	rbf	52.845 (21.9848)	**-0.4048** (**-42.2657**)
Rastrigin	20	krig-hp	11.0539 (2.19551)	4.60652 (0.44098)
Schwefel	2	rbf	47.4495 (64.8173)	**-43.4264** (**-27.2863**)
Schwefel	2	krig-hp	0.194305 (0.0612316)	**-0.018209** (**-0.145149**)
Schwefel	5	rbf	584.147 (221.872)	39.448 (**-96.193**)
Schwefel	5	krig-hp	4.19794 (2.67233)	0.46176 (0.87488)
Schwefel	10	rbf	1907.19 (388.003)	539.32 (49.761)
Schwefel	10	krig-hp	8.52275 (2.52773)	5.47446 (1.55199)
Schwefel	20	rbf	5006.46 (374.814)	456.55 (162.955)
Schwefel	20	krig-hp	162.681 (106.106)	88.9398 (81.6433)

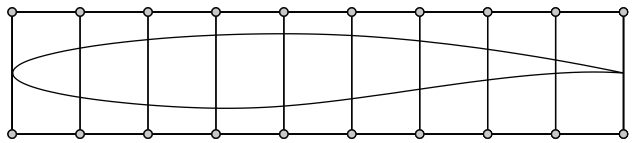


Figure 21: FFD control point definition.

approach (SU2SBO) takes more CFD effort to accurately train the surrogate model: after the initial design space sampling by LHS (180 samples), the objective function has been decreased to just 8%. From that point on, the global approximation starts to predict the objective function landscape and the adaptive sampling provides new insight into unknown regions: their combined effect causes a significant performance improvement at around 300 design solutions. The last 100 solutions are sequentially introduced in the database by optimizing the surrogate with an in-house genetic algorithm [41]. A further drop is observable in the very last stages which pushes the obtained performance beyond the SU2AO best result.

Figure 23 depicts a geometric and aerodynamic comparison of baseline and optimized airfoil shapes. The shock wave intensity reduction and the consequent improvement of the boundary layer behaviour past the flow discontinuity are highlighted by the redesign of the front airfoil region (to increase the flow expansion peak) of the whole pressure side (to recover lift and pitching moment) and of the rear part of the suction side (to control the shock). Similar results have been found in the reference paper. Figure 24 depicts the Mach number contour map and shows how the supersonic region has been greatly reduced and moved upstream. Finally, Table 10

Table 8: Summary of optimization studies for the RAE 2822 case.

Run ID	Reference [31]			Present	
	SBOSA	EGO	PGA	SU2AO	SU2SBO
Opt. method	SBO	EGO	GA	Adjoint gradient-based	SBO
CFD code		ZEN		SU2	
Parameterization		CST		FFD	
Design parameters		14		18	
Fitness eval.	POD	Kriging	CFD	CFD	Kriging
Total CFD eval.	300	224	6400	160 = 70 (CFD) + 3*30 (adjoint)	660

Table 9: SU2SBO optimization setup.

n_{apr}	$n_{adpt}(n_{par})$	n_{opt}	Criteria (% act. prob.)
180	480 (36)	100	EI (30%) + EI-like (70%)

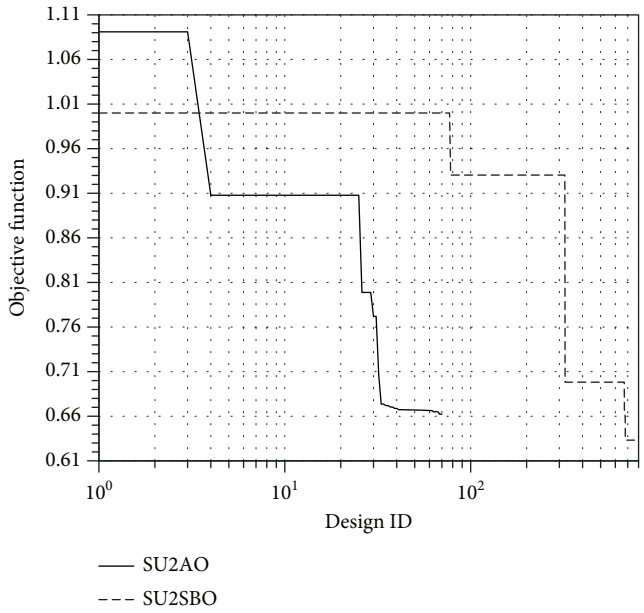
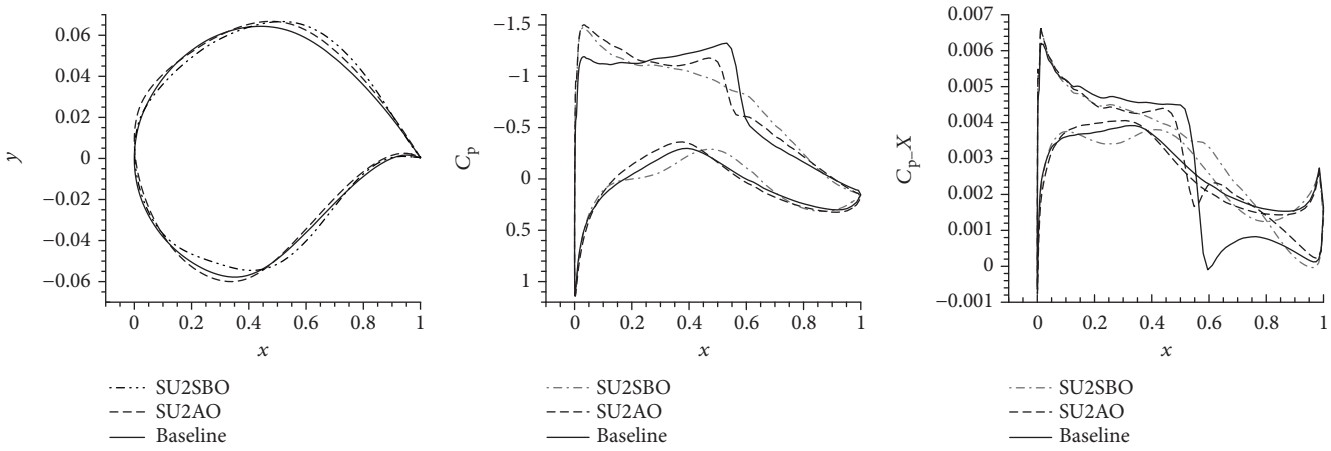


Figure 22: History of minimum objective function values for SU2SBO and SU2AO optimizations.

provides quantitative figures about the optimal solutions from the reference paper and present simulations. The baseline airfoil drag coefficients obtained with both ZEN and SU2 codes have been reported as they represent the reference performance for corresponding cases: indeed, the percentage ΔC_d change is also shown in the rightmost column. The present results are fully in line with previous studies as the optimized airfoils satisfy all constraints and feature drag reductions of the order of 35%–40% with respect to the baseline shape.

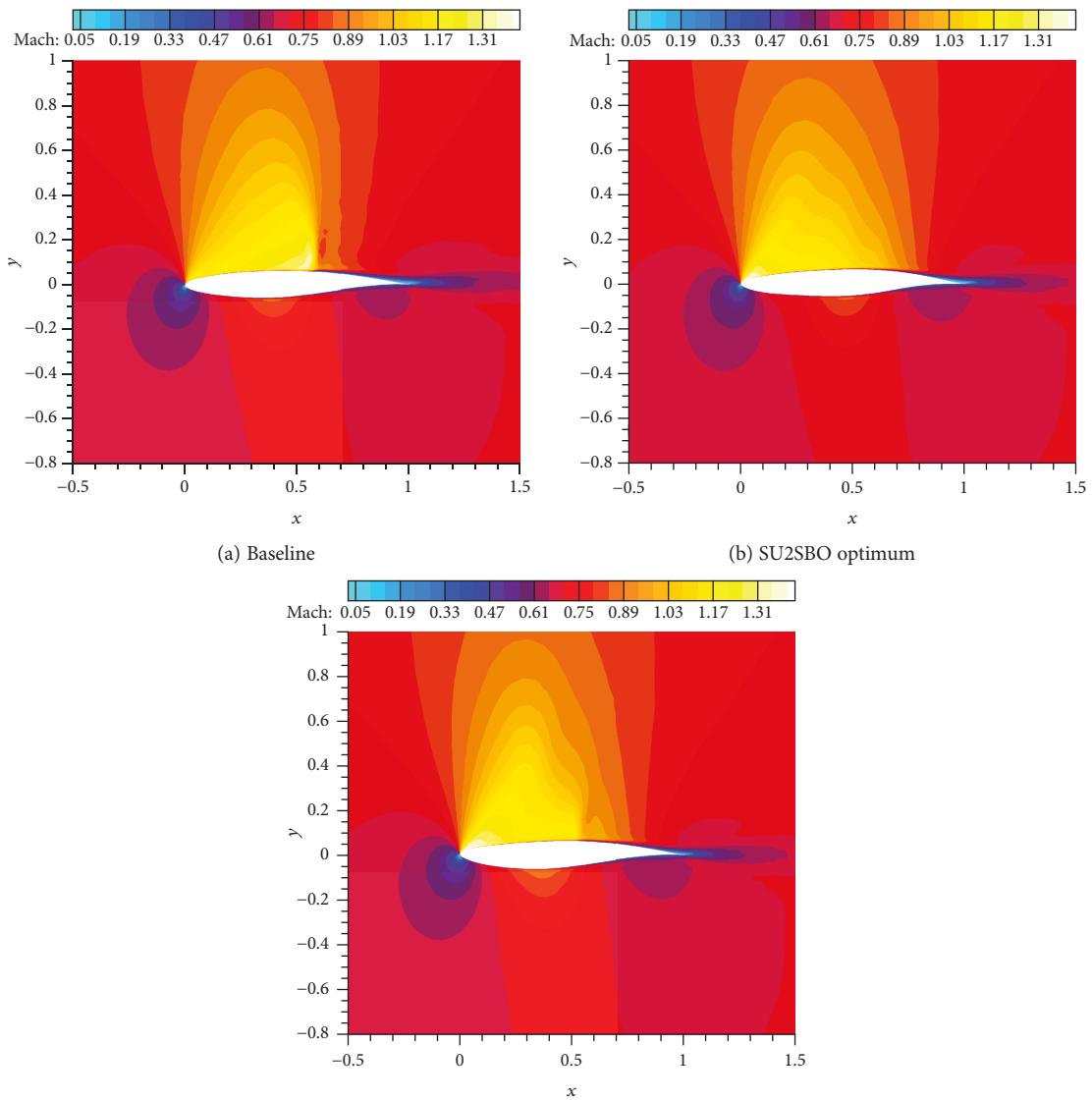
7. Conclusions

A framework for surrogate-assisted optimization has been presented, featuring design exploration, adaptive sampling, and sequential global search. A series of infill criteria have been proposed to adaptively choose new points to be added to the surrogate database: most of them pursue the exploration-exploitation balance and are aimed at providing an improved version of the surrogate to the final optimization stage. A new feature has been added which allows triggering several criteria during the sampling phase according to activation probabilities. Gaussian-based and radial basis function network models are used as surrogates of the objective function. Two experimental test campaigns have been performed: the first has involved analytic test functions with multidimensional and scalable character in order to quickly analyze the performance of the method and the second campaign is a prosecution of a previous work as it deals with the aerodynamic shape optimization of an airfoil profile in transonic viscous flow conditions. The open-source SU2 package has been exploited in most of its functionalities related to aerodesign. Results have been compared to the reference and to an adjoint gradient-based optimization. The work confirmed that approximate solutions, close to the global optima, can be found with the proposed surrogate-assisted optimization. Simulations with test functions highlighted that the proper combination of surrogate and infill criteria can be quite sensible to the function type and suggested that, by having multiple surrogates and criteria available, probably the best option is to exploit them in an ensemble approach. This will be the main topic of a future research study. Concerning the aeroshape optimization case, the achievements are in line with previous investigations as all the basic (geometric and aerodynamic) features of the optimal shape have been captured by the present approach. In the context of global and real-world optimization, the adoption of surrogate models is essential to reduce the computational burden. The proposed example clearly indicates the benefits associated with their usage in terms of cost/performance trade-off. Further application will deal with more complex cases, e.g., three-dimensional benchmark problems defined within the AIAA Aerodynamic Design Optimization Discussion Group and interference drag reduction of wing/pylon/nacelle configuration.



(a) Airfoil shapes (b) Pressure coefficient (c) Skin friction coefficient

Figure 23: Geometry and aerodynamic comparison.



(a) Baseline (b) SU2SBO optimum

(c) SU2AO optimum

Figure 24: Contour map of the Mach number.

Table 10: Comparison of the best solutions.

Run	Total CFD evals.	C_d (ZEN, fine mesh)	C_d (SU2, coarse mesh)	ΔC_d wrt ref.
RAE 2822 baseline	—	0.0194	0.0206	—
PGA best	6400	0.0116	na	-40.2%
SBOSA best	300	0.0118	na	-39.2%
EGO best	224	0.0132	na	-32.0%
SU2AO best	160	na	0.0132	-36.0%
SU2SBO best	760	na	0.0126	-38.8%

Data Availability

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

References

[1] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[2] A. I. J. Forrester and A. J. Keane, "Recent advances in surrogate-based optimization," *Progress in Aerospace Science*, vol. 45, no. 1-3, pp. 50–79, 2009.

[3] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. Kevin Tucker, "Surrogate-based analysis and optimization," *Progress in Aerospace Science*, vol. 41, no. 1, pp. 1–28, 2005.

[4] A. I. J. Forrester, N. W. Bressloff, and A. J. Keane, "Optimization using surrogate models and partially converged computational fluid dynamics simulations," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 462, no. 2071, pp. 2177–2204, 2006.

[5] T. Braconnier, M. Ferrier, J.-C. Jouhaud, M. Montagnac, and P. Sagaut, "Towards an adaptive POD/SVD surrogate model for aeronautic design," *Computers & Fluids*, vol. 40, no. 1, pp. 195–209, 2011.

[6] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo, "Ensemble of surrogates," *Structural and Multidisciplinary Optimization*, vol. 33, no. 3, pp. 199–216, 2007.

[7] D. Huang, T. T. Allen, W. I. Notz, and N. Zeng, "Global optimization of stochastic black-box systems via sequential kriging meta-models," *Journal of Global Optimization*, vol. 34, no. 3, pp. 441–466, 2006.

[8] Y. Jin, "Surrogate-assisted evolutionary computation: recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.

[9] V. Picheny, T. Wagner, and D. Ginsbourger, "A benchmark of kriging-based infill criteria for noisy optimization," *Structural and Multidisciplinary Optimization*, vol. 48, no. 3, pp. 607–626, 2013.

[10] C. Q. Lam, *Sequential adaptive designs in computer experiments for response surface model fit. Ph.D. thesis*, The Ohio State University, Columbus, OH, USA, 2008.

[11] S. Jakobsson, M. Patriksson, J. Rudholm, and A. Wojciechowski, "A method for simulation based optimization using radial basis functions," *Optimization and Engineering*, vol. 11, no. 4, pp. 501–532, 2010.

[12] A. Sóbester, S. J. Leary, and A. J. Keane, "On the design of optimization strategies based on global response surface approximation models," *Journal of Global Optimization*, vol. 33, no. 1, pp. 31–59, 2005.

[13] N. Mai-Duy and T. Tran-Cong, "Approximation of function and its derivatives using radial basis function networks," *Applied Mathematical Modelling*, vol. 27, no. 3, pp. 197–220, 2003.

[14] Y. Xiong, W. Chen, D. Apley, and X. Ding, "A non-stationary covariance-based kriging method for metamodelling in engineering design," *International Journal for Numerical Methods in Engineering*, vol. 71, no. 6, pp. 733–756, 2007.

[15] Y. Tenne and S. W. Armfield, "A framework for memetic optimization using variable global and local surrogate models," *Soft Computing*, vol. 13, no. 8-9, pp. 781–793, 2009.

[16] J. Müller and R. Piché, "Mixture surrogate models based on Dempster-Shafer theory for global optimization problems," *Journal of Global Optimization*, vol. 51, no. 1, pp. 79–104, 2011.

[17] F. A. C. Viana, R. T. Haftka, and L. T. Watson, "Efficient global optimization algorithm assisted by multiple surrogate techniques," *Journal of Global Optimization*, vol. 56, no. 2, pp. 669–689, 2013.

[18] R. G. Regis, "Trust regions in kriging-based optimization with expected improvement," *Engineering Optimization*, vol. 48, no. 6, pp. 1037–1059, 2016.

[19] T. Robinson, K. Willcox, M. Eldred, and R. Haimes, "Multifidelity optimization for variable-complexity design," in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia, September 2006.

[20] M. J. Mifsud, S. T. Shaw, and D. G. MacManus, "A high-fidelity low-cost aerodynamic model using proper orthogonal decomposition," *International Journal for Numerical Methods in Fluids*, vol. 63, no. 4, pp. 468–494, 2009.

[21] R. P. Liem, C. A. Mader, and J. R. R. A. Martins, "Surrogate models and mixtures of experts in aerodynamic performance prediction for aircraft mission analysis," *Aerospace Science and Technology*, vol. 43, pp. 126–151, 2015.

[22] A. J. Booker, J. E. Dennis Jr, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset, "A rigorous framework for optimization of expensive functions by surrogates," *Structural and Multidisciplinary Optimization*, vol. 17, no. 1, pp. 1–13, 1999.

[23] Y. Tenne and S. W. Armfield, "A versatile surrogate-assisted memetic algorithm for optimization of computationally

expensive functions and its engineering applications," in *Success in Evolutionary Computation. Studies in Computational Intelligence*, A. Yang, Y. Shan, and L. T. Bui, Eds., vol. 92, pp. 43–72, Springer, Berlin, Heidelberg, 2008.

[24] S. Koziel and L. Leifsson, "Surrogate-based aerodynamic shape optimization by variable-resolution models," *AIAA Journal*, vol. 51, no. 1, pp. 94–106, 2013.

[25] Z.-H. Han and K.-S. Zhang, "Surrogate-based optimization," in *Real-World Applications of Genetic Algorithms*, pp. 343–362, InTech, 2012.

[26] Y. Mack, T. Goel, W. Shyy, and R. Haftka, *Evolutionary Computation in Dynamic and Uncertain Environments*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[27] J. T. Hwang and J. R. R. A. Martins, "A fast-prediction surrogate model for large datasets," *Aerospace Science and Technology*, vol. 75, pp. 74–87, 2018.

[28] E. Iuliano and D. Quagliarella, "Proper orthogonal decomposition, surrogate modelling and evolutionary optimization in aerodynamic design," *Computers & Fluids*, vol. 84, pp. 327–350, 2013.

[29] E. Iuliano and D. Quagliarella, "Aerodynamic shape optimization via non-intrusive POD-based surrogate modelling," in *2013 IEEE Congress on Evolutionary Computation*, Cancun, Mexico, June 2013.

[30] E. Iuliano and E. A. Pérez, *Application of Surrogate-Based Global Optimization to Aerodynamic Design*, Springer Tracts in Mechanical Engineering, Springer, 2016.

[31] E. Iuliano, "Global optimization of benchmark aerodynamic cases using physics-based surrogate models," *Aerospace Science and Technology*, vol. 67, pp. 273–286, 2017.

[32] J. D. Martin and T. W. Simpson, "Use of kriging models to approximate deterministic computer models," *AIAA Journal*, vol. 43, no. 4, pp. 853–863, 2005.

[33] D. J. J. Toal, N. W. Bressloff, and A. J. Keane, "Kriging hyperparameter tuning strategies," *AIAA Journal*, vol. 46, no. 5, pp. 1240–1252, 2008.

[34] C. H. da Silva Santos, M. S. Goncalves, and H. E. Hernandez-Figueroa, "Designing novel photonic devices by bio-inspired computing," *IEEE Photonics Technology Letters*, vol. 22, no. 15, pp. 1177–1179, 2010.

[35] J. A. Richardson and J. L. Kuester, "Algorithm 454: the complex method for constrained optimization [e4]," *Communications of the ACM*, vol. 16, no. 8, pp. 487–489, 1973.

[36] H. M. Gutmann, "A radial basis function method for global optimization," *Journal of Global Optimization*, vol. 19, no. 3, pp. 201–227, 2001.

[37] G. E. Fasshauer and J. G. Zhang, "On choosing "optimal" shape parameters for RBF approximation," *Numerical Algorithms*, vol. 45, no. 1-4, pp. 345–368, 2007.

[38] S. Rippa, "An algorithm for selecting a good value for the parameter c in radial basis function interpolation," *Advances in Computational Mathematics*, vol. 11, no. 2/3, pp. 193–210, 1999.

[39] M. Schonlau, W. J. Welch, and D. R. Jones, "Global versus local search in constrained optimization of computer models," in *New developments and applications in experimental design*, vol. 34, pp. 11–25, Institute of Mathematical Statistics, Hayward, CA, 1998.

[40] D. Maljovec, B. Wang, A. Kupresanin, G. Johannesson, V. Pascucci, and P. T. Bremer, "Adaptive sampling with topological scores," *International Journal for Uncertainty Quantification*, vol. 3, no. 2, pp. 119–141, 2013.

[41] D. Quagliarella, P. Iannelli, P. L. Vitagliano, and G. Chinnici, "Aerodynamic shape design using hybrid evolutionary computation and fitness approximation," in *AIAA 1st Intelligent Systems Technical Conference*, Chicago, IL, September 2004.

[42] N. Hansen, "The CMA evolution strategy: a comparing review," in *Towards a New Evolutionary Computation. Studies in Fuzziness and Soft Computing*, J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds., vol. 192, pp. 75–102, Springer, Berlin, Heidelberg, 2006.

[43] B. M. Kulfan, "Universal parametric geometry representation method," *Journal of Aircraft*, vol. 45, no. 1, pp. 142–158, 2008.

[44] T. A. Albring, M. Sagebaum, and N. R. Gauger, "Efficient aerodynamic design using the discrete adjoint method in SU2," in *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Washington, D.C., June 2016.

[45] T. D. Economon, D. Mudigere, G. Bansal et al., "Performance optimizations for scalable implicit rans calculations with SU2," *Computers & Fluids*, vol. 129, pp. 146–158, 2016.

[46] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, "SU2: an open-source suite for multiphysics simulation and design," *AIAA Journal*, vol. 54, no. 3, pp. 828–846, 2016.

[47] M. Amato and P. Catalano, "Non linear k e turbulence modeling for industrial applications," in *ICAS 2000 Congress*, IOS Press, Harrogate, UK, 2000.

[48] B. M. Adams, L. E. Bauman, W. J. Bohnhoff et al., "Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 5.4 user's manual. Tech. Rep. SAND2010-2183," Sandia, 2013.