Finding efficient trade-offs in multi-fidelity response surface modelling

Rijn, S.J. van; Schmitt, S.; Leeuwen, M. van; Bäck, T.H.W.

Engineering Optimization

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/geno20

Finding efficient trade-offs in multi-fidelity response surface modelling

Sander van Rijn, Sebastian Schmitt, Matthijs van Leeuwen & Thomas Bäck

To cite this article: Sander van Rijn, Sebastian Schmitt, Matthijs van Leeuwen & Thomas Bäck (2022): Finding efficient trade-offs in multi-fidelity response surface modelling, Engineering Optimization, DOI: 10.1080/0305215X.2022.2052286

To link to this article: https://doi.org/10.1080/0305215X.2022.2052286

Published online: 16 May 2022.

ENGINEERING OPTIMIZATION
https://doi.org/10.1080/0305215X.2022.2052286

Taylor & Francis
Taylor & Francis Group

Finding efficient trade-offs in multi-fidelity response surface modelling

Sander van Rijn [a], Sebastian Schmitt [b], Matthijs van Leeuwen [a] and Thomas Bäck [a]

[a]LIACS, Leiden University, Leiden, The Netherlands; [b]Honda Research Institute Europe GmbH, Offenbach am Main, Germany

ABSTRACT

In optimization approaches to engineering applications, time-consuming simulations are often utilized that can be configured to deliver solutions for various fidelity (accuracy) levels. It is common practice to train hierarchical surrogate models on objective functions in order to speed up the optimization process. These operate under the assumption that there is a correlation between the different fidelities that can be exploited to gain information cheaply. However, limited guidelines are available to help divide the available computational budget between multiple fidelities in practice. This article evaluates a range of different choices for a two-fidelity setup that provide helpful intuitions about this trade-off. An heuristic method is presented based on subsampling from an initial Design of Experiments to find a suitable division of the computational budget between the fidelity levels. This enables the setting up of multi-fidelity optimizations that utilize the available computational budget efficiently, independently of the multi-fidelity model used.

ARTICLE HISTORY
Received 28 May 2021
Accepted 19 January 2022

KEYWORDS
Multi-fidelity simulation problems; design of experiments; error grids; hierarchical surrogate models; co-kriging

1. Introduction

When dealing with simulation based optimization problems in engineering applications, the runtime cost of each evaluation is typically the most restrictive aspect of a successful approach. Surrogate models are often used to reduce the total computational load by learning trends from previous evaluations. But the computational cost for single evaluations have grown too high in many modern problems for such approaches to obtain enough information necessary to train an accurate model in a reasonable time.

Many such problems offer tunable accuracy and can therefore be classified as either arbitrarily tunable *variable-fidelity*, or discretely tunable *multi-fidelity*, problems. This article focuses on multi-fidelity problems specifically. Supplementing accurate high-fidelity information with cheaper low-fidelity information is regularly done by incorporating *hierarchical* (co-)surrogate models based on work by Kennedy and O'Hagan (2000), such as co-kriging (Forrester, Sóbester, and Keane 2007) and co-Radial Basis Functions (RBFs) (Durantin *et al.* 2017). These have been successfully applied in, for example, the design of ships (Pellegrini *et al.* 2016), airfoils (Liu *et al.* 2018), satellites (Shi *et al.* 2020), additive manufacturing (Zhou, Hsieh, and Wang 2019) and fire start determination (Li *et al.* 2019).

However, it remains unclear under which conditions the inclusion of low-fidelity information in hierarchical surrogate models is actually beneficial. While previous research has shown that the

CONTACT Sander van Rijn s.j.van.rijn@liacs.leidenuniv.nl

correlation between high- and low-fidelity response surfaces should be fairly high—*i.e.* a sample correlation coefficient *correlation* > 0.9 (Toal 2015; Fernández-Godino *et al.* 2016)—a high correlation by itself is no guarantee of achieving added benefit for multi-fidelity models. That is, regardless of correlation, individual response surface landscapes still have a substantial impact on the final accuracy of the trained models.

Furthermore, even if a model is beneficial, how best to distribute the available computational budget between the fidelity levels is still an open question. Prior work has included experiments where models based on multiple sample sizes were compared, but most presented only a limited selection of combinations, as can be seen in, for example, the overviews by Fernández-Godino *et al.* (2016, 2019). Common heuristics for deciding on this division rely either on the cost ratio between fidelities or otherwise use expected information gains (Guo *et al.* 2020; Moss, Leslie, and Rayson 2020; Huang *et al.* 2006; Ryou, Tal, and Karaman 2020; Belakaria, Deshwal, and Doppa 2020). Of these, the former do not use any function information, while the latter are designed to be used in iterative optimization, not to gain general understanding.

This work explores empirically how to distribute additional computational budget over two fidelities. The focus is on one-shot Design of Experiments (DoEs), by enumerating all possible combinations of a wide range of low- and high-fidelity samples, fitting a hierarchical model and measuring its accuracy for each setup. This approach is similar to a study by Durantin *et al.* (2017) but with a much finer granularity, more sample combinations and many more benchmark functions. The aim is to provide general insight into the behaviour of this trade-off by analysing model accuracy as a function of the DoE sizes and for various benchmark functions. Recognizing that using an enumeration procedure to obtain this information is far too computationally expensive in terms of problem evaluations for practical settings, this article presents a method using subsampling that draws smaller DoEs from an initial DoE to avoid performing any new evaluations. The accuracy trend results are approximated using these subsampled DoEs and show a high correlation between these results and those from the original full enumeration.

An heuristic is presented that utilizes the information gained from the subsampling approach to predict a beneficial split of the number of high- and low-fidelity samples for a given computational budget. This allows for an efficient use of the available computational resources for the multi-fidelity modelling approach to optimization.

All files for this work are archived on Zenodo (van Rijn 2021; van Rijn *et al.* "Generated Data Files and Figures" 2021). Links are given in notes to the captions for each figure to the source code on GitHub (van Rijn 2020) and full versions on FigShare (van Rijn *et al.* "Figures from Paper" 2021), respectively. Appendices 1 and 2 appear after the references list.

2. Background

This section defines some terms and methods that are used in the remainder of this article.

2.1. Multi-fidelity problems

A multi-fidelity problem is an optimization/simulation problem that is available in multiple *fidelities*, *i.e.* accuracy levels. In real-world Computational Fluid Dynamics (CFD) simulations of, for example, airfoils, these fidelities could correspond to different mesh sizes or simulation types. A *low*-fidelity simulation would use a coarse mesh or potential flow solver, and thus give lower accuracy, but be faster to calculate, while a *high*-fidelity simulation would use a finer mesh or Reynolds-Averaged Navier–Stokes (RANS) simulation and therefore be more accurate while taking longer to calculate.

In the following, $f_h : \mathcal{X} \rightarrow \mathcal{Y}$ and $f_l : \mathcal{X} \rightarrow \mathcal{Y}$ are used to denote the high- and low-fidelity levels of a simulator, abstractly represented by the function f that maps input vectors $\mathbf{x} \in \mathcal{X}$ onto outputs $\mathbf{y} \in \mathcal{Y}$.

2.2. Additive hierarchical surrogate models

To make use of the multiple sources of information in a multi-fidelity problem, an additive model structure has been proposed by Kennedy and O'Hagan (2000):

$$z_h(\mathbf{x}) = \rho z_l(\mathbf{x}) + \delta(\mathbf{x}). \tag{1}$$

Here, $z_h(\mathbf{x})$, and $z_l(\mathbf{x})$ are the high- and low-fidelity surrogate models at point $\mathbf{x}$, respectively, for approximating $f_h(\mathbf{x})$ and $f_l(\mathbf{x})$; ρ is a regression parameter and $\delta(\mathbf{x})$ is the difference model at point $\mathbf{x}$, which improves the low-fidelity prediction by additive correction.

Without loss of generality, simplified additive models are considered where the regression parameter $\rho = 1$, to limit algorithmic complexity. While this may reduce the achievable accuracy of the models, it is of no relevance to the introduced concepts. Learning this parameter will most likely only further increase the performance of this approach. Independent models for z_l and δ are created, where z_l models the lowest accuracy information source f_l, and a separate model δ predicts the differences between the high- and low-fidelity responses $f_h(\mathbf{x}) - f_l(\mathbf{x})$. Specifically, kriging models using the Matérn kernel were used for this article, although the proposed method does not depend on this particular choice and could use other models.

2.3. Multi-fidelity design of experiments

A standard approach for systematically sampling a set of input parameter configurations in order to create a dataset of input–output pairs of a given function is referred to as a Design of Experiments (DoE) (Montgomery 2019). The goal when choosing such a dataset is to cover the input-space of the function in such a way that the created model is as good as it can be, whether on a local or global scale. How large the search space is and how much computational effort can be expended on this depends on the problem setting. However, a full factorial design (*i.e.* a grid search) is usually out of the question owing to the relatively high dimensionality and high computational cost. In this work, the common Latin Hypercube Sample (LHS) strategy is used. This technique tries to create a sample such that the samples are evenly distributed over the search space, while avoiding the reuse of coordinate values for a dimension.

In a multi-fidelity setting, where a difference model between high- and low-fidelity functions is trained, overlapping DoEs for low- and high-fidelity models where all high-fidelity samples are also included in the low-fidelity DoE are preferred. Additionally, each DoE should still cover the search space efficiently and therefore should be an approximate LHS itself.

To achieve this, the procedure from Le Gratiet (2013) is used to generate DoEs for the hierarchical models, as outlined in Algorithm 1. In lines 1 and 2, two separate LHSs are generated. Then, for

Algorithm 1 Multi-fidelity LHS Le Gratiet (2013)

Require: $n_l \geq n_h + 1$

1: $H \leftarrow \text{LHS}(n_h)$ ▷ Independent samples per fidelity
2: $L \leftarrow \text{LHS}(n_l)$
3: $L', H' \leftarrow \emptyset$
4: **while** H not empty **do**
5: $\quad \mathbf{h}, \mathbf{l} \leftarrow \arg\min_{\mathbf{h}\in H, \mathbf{l}\in L} \|\mathbf{h} - \mathbf{l}\|$ ▷ Find closest pair
6: $\quad H' \leftarrow H' \bigcup \mathbf{h}$
7: $\quad L' \leftarrow L' \bigcup \mathbf{h}$ ▷ Effectively adjust $\mathbf{l}$ to $\mathbf{h}$
8: $\quad$ Remove $\mathbf{h}, \mathbf{l}$ from H, L
9: **end while**
10: $L' \leftarrow L' \bigcup L$ ▷ Add remaining low-fidelity points
11: **return** H', L'

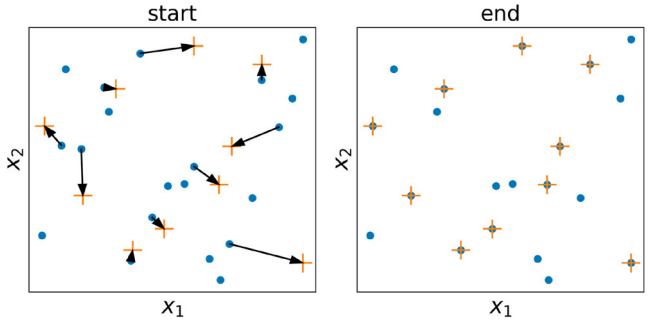


Figure 1. Illustration of Algorithm 1. A two-dimensional DoE with $n_h = |H| = 10$ (shown as pluses) and $n_l = |L| = 20$ (shown as dots).[1,2]

each high-fidelity point, the closest low-fidelity point is replaced by that high-fidelity point (the while loop in lines 4–9). Given the desired sizes, this method returns both a high-fidelity LHS H, and a more spread-out low-fidelity sample L that is still roughly an LHS itself. The final outcome is a DoE that is the union of the sets L and $H \subset L$ of low- and high-fidelity samples, respectively, denoted as $\text{DoE}(H, L)$ and where the exact differences $f_h(\mathbf{x}) - f_l(\mathbf{x})$ can be computed for all $\mathbf{x} \in H$ and used as a training set for the difference model δ. Figure 1 illustrates this procedure.

3. Problem statement

The fundamental question addressed in this work is: how should a given additional computational budget be distributed among multiple fidelities? Especially, in the context of computationally expensive simulation problems where the overall evaluation budget is constrained, this is a highly relevant question. Addressing the question of how to split the high- and low-fidelity samples for the $\text{DoE}(H, L)$ is chosen as the starting point for this article.

The answer to this question depends on which fidelity provides the most information for its computational cost. An additional high-fidelity sample in a so-far unexplored area will definitely improve the model's accuracy. But if some number of low-fidelity samples can improve the model more with equal or lower computational cost, that might be a better choice. How much information is gained by adding another sample for a specific fidelity depends heavily on the number of samples of that fidelity already present, and on the chosen model's capacity to capture the problem's response surface.

An important quantity in determining the split between the number of high- and low-fidelity evaluations is given by the *cost ratio* $\phi = c_l/c_h \in (0, 1)$, where c_h and c_l are typical computation times associated with high- and low-fidelity evaluations, respectively. The problem can thus be stated as follows: given a fixed evaluation budget b (which is measured in high-fidelity evaluation times) and cost ratio ϕ, what are the optimal numbers of high- and low-fidelity evaluations, *i.e.* the optimal division ratio n_h/n_l, that minimizes the model error and which respects the constraint that the budget is not exceeded, *i.e.* $n_h + \phi n_l \leq b$? The Mean Squared Error (MSE) of the response surface model z compared with the true function value of the highest fidelity level f_h is taken as the error model, evaluated on a given test set $x \in T$:

$$\text{MSE}(z, T) = \sum_{\mathbf{x} \in T} \frac{(z(\mathbf{x}) - f_h(\mathbf{x}))^2}{|T|}. \tag{2}$$

The model error is expected to have a non-trivial and in general nonlinear behaviour as a function of the division ratio. For the case consisting only of high-fidelity evaluations, the low number of overall samples probably leads to a rather large error. On the other extreme, using only low-fidelity evaluations will also not produce an accurate model as no actual information about the true function is

used. In the intermediate region, with some low-fidelity samples at the expense of a few high-fidelity evaluations, it is expected that a reduced model error will be obtained. The details of this trade-off strongly depend on various aspects of the problem, like the structure of the high-fidelity function and the similarity between the low- and the high-fidelity functions. However, in order to learn the dominant behaviour of the error as a function of the number of high- and low-fidelity samples, a simple fit to the error is employed. This enables the extraction of the global trend and allows for formulating an heuristic that guides the distribution of additional computational budget.

4. Method

4.1. Enumeration of multi-fidelity DoE sizes

To examine the trade-off between the number of high- and low-fidelity samples fully, model accuracy information needs to be obtained for many possible combinations. Such information is gathered by empirically performing a full enumeration of all possible combinations (n_l, n_h) for $2 < n_h < n_h^{\max}$ and $n_h + 1 < n_l < n_l^{\max}$. For each pair (n_l, n_h) multiple $(I = 50)$ hierarchical multi-fidelity models were trained to collect some statistics and evaluate the errors on an independent test set T. The tables of errors for the complete enumeration DoEs are from here on referred to as *error grids*.

Algorithm 2 Full enumeration error grid

Require: N-dimensional multi-fidelity problem (f_h, f_l)
Require: $n_h^{\max}, n_l^{\max}$ ▷ Maximum DoE size
Require: I ▷ Number of iterations
1: $E \leftarrow \emptyset$ ▷ Error grid storage
2: $T \leftarrow \text{LHS}(500{\cdot}N)$ ▷ Independent test set
3: **for** $n_h = 2 \ldots n_h^{\max}$ **do**
4: **for** $n_l = (n_h + 1) \ldots n_l^{\max}$ **do**
5: **for** $i = 1 \ldots I$ **do**
6: $H, L \leftarrow \text{MF} - \text{LHS}(n_h, n_l)$ ▷ Algorithm 1
7: $Y_h, Y_l \leftarrow f_h(H), f_l(L)$ ▷ Evaluate
8: Train z_h using H, L, Y_h, Y_l
9: $E[n_h, n - l, i] \leftarrow \text{MSE}(z_h, T)$ ▷ Equation (2)
10: **end for**
11: **end for**
12: **end for**
13: **return** E

Algorithm 2 lists a pseudocode representation of the procedure by which the error grids are obtained. The size of the test set T is set to $|T| = 500{\cdot}N$, where N is the dimensionality of the search space (line 2). For each combination (n_h, n_l), $I = 50$ independent DoEs are sampled and the errors for the multi-fidelity model based on each DoE are evaluated and stored (lines 5–10).

The resulting error between the surrogate model and the true high-fidelity function can be visualized in heatmaps of the error grids as shown in Figure 2. The median error over the I independent realizations of the DoEs are shown as 2D heatmaps and as a function of H and L. Experiments showed that the distribution of the MSEs is exponential, so $\log_{10}(\text{MSE})$ is used to account for the different error scales better. These error grids serve as the basis for the analysis that allow:

- examining the dependence of the model error as a function of the division ratio between the number of high- and low-fidelity evaluations;
- examining how this dependency varies between multi-fidelity problems; and
- identifying the optimal division ratio for a given budget and problem.

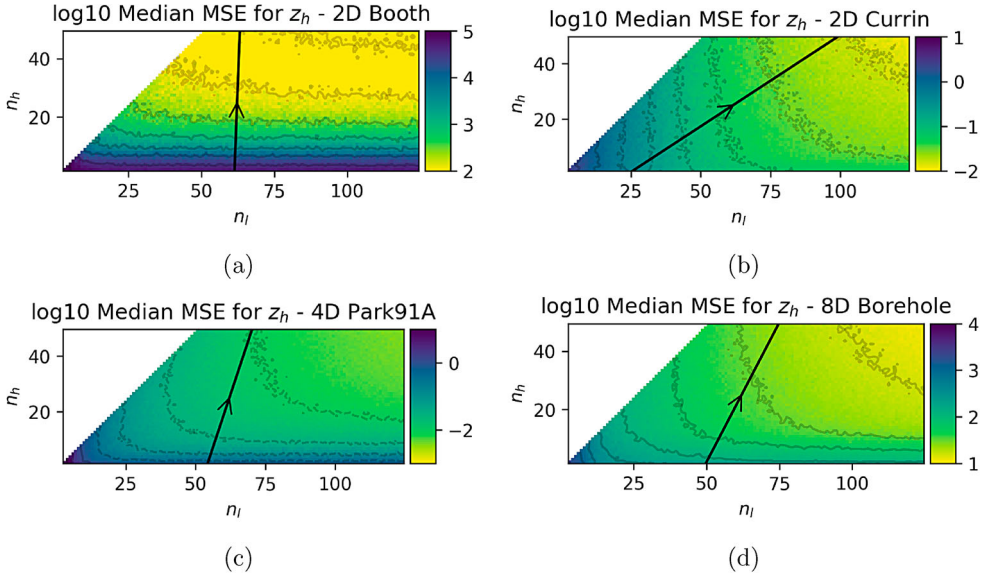


Figure 2. Error grids. Heatmaps of $\log_{10}$ of the hierarchical model MSE for varying DoE sizes, shown as the median over $I = 50$ iterations, on four benchmark problems. The black arrow shows the global gradient direction as described in Section 4.3. Overall, it is clear that adding more samples improves model accuracy, either additional high-fidelity samples (vertical) or low-fidelity samples (horizontal).[3,4]

However, the number of DoEs that need to be evaluated in this full enumeration procedure is $N_{\text{DoE}} = I\, {n_h}^{\max}\, ({n_l}^{\max} - {n_h}^{\max}/2)$, which for $I = 50$, ${n_h}^{\max} = 50$, ${n_l}^{\max} = 125$ is a total of $N_{\text{DoE}} =$ 250,000 DoEs to sample and models to train. The number of high-fidelity function evaluations is consequently in the millions, which might be feasible for trivially computable benchmark problems, but will be prohibitively unfeasible for real-world problems with higher computational demand.

4.2. Cross-validated subsampling of DoE sizes

This section describes how to approximate error grids using only one fixed initial multi-fidelity DoE. Given one DoE(H, L) with $n_h = |H|$ and $n_l = |L|$ samples, a full error grid can be created by reusing these evaluated samples and creating a set of smaller subsampled DoEs. Concretely, $H' \subset H, L' \subset L$ of size $({n_h}', {n_l}')$ is subsampled such that the set of high-fidelity samples H' is still a true subset of the set of low-fidelity samples L', *i.e.* $H' \subset L'$. For this, H' is first drawn uniformly at random without replacement from the available H. Then, those samples are taken as a start for L', and randomly chosen low-fidelity points are added until the desired size is reached (see Algorithm 3).

Algorithm 3 Subsampling MF-DoE

Require: Initial multi-fidelity DoE(H, L)
Require: Desired DoE size ${n_h}', {n_l}'$
1: $H' \leftarrow$ uniform randomly choose ${n_h}'$ samples from H
2: $L' \leftarrow$ uniform randomly choose remaining ${n_l}' - {n_h}'$ samples from $(L \setminus H')$
3: $L' \leftarrow L' \bigcup H'$
4: **return** H', L'

Since the chosen high-fidelity DoE H' is a strict subset of the original DoE H, some samples left out of the subsampled DoE H' can serve as test set $H^{\text{test}} = H \setminus H'$ and the error of the surrogate models

for each DoE similar to cross-validation can be calculated. The complete subsampling approach is summarized in the pseudocode shown in Algorithm 4.

Algorithm 4 Subsampling error grid procedure

Require: N-dimensional multi-fidelity problem (f_h, f_l)
Require: I ▷ Number of iterations
$E \leftarrow \emptyset$ ▷ Error grid storage
$H, L \leftarrow \mathrm{MF} - \mathrm{LHS}({n_h}^{\max}, {n_l}^{\max})$ ▷ Algorithm 1
$Y_h, Y_l \leftarrow f_h(H), f_l(L)$ ▷ Evaluate once
for $n_h = 2 \ldots {n_h}^{\max} - 1$ **do**
  for $n_l = (n_h + 1) \ldots {n_l}^{\max}$ **do**
    for $i = 1 \ldots I$ **do**
      $H', L' \leftarrow$ Subsample H, L ▷ Algorithm 3
      $Y_h', Y_l' \leftarrow$ values from Y_h, Y_l for H', L'
      Train z_h using H', L', Y_h', Y_l'
      $E[n_h, n_l, i] \leftarrow \mathrm{MSE}(z_h, H^{\mathrm{tst}})$ ▷ $H^{\mathrm{tst}} : H \setminus H'$
    end for
  end for
end for
return E

A comparison of the subsampling and full enumeration procedure is made in Appendix 2.

4.3. Angle of gradient quantification

The error grids provide intuitive information about the trade-off between the numbers of high- and low-fidelity samples. The contour lines give clear visual guidance about in which direction of the (n_l, n_h)-plane the accuracy of the surrogate models increases. To evaluate this behaviour quantitatively, the gradient of the error with respect to the number of samples is used. If this gradient direction is predominantly along the n_h direction, *i.e.* it has an angle close to 90° as measured from the horizontal n_l axis (see the example in Figure 2), this indicates that improvements in model quality mostly depend on additional high-fidelity information. However, if the error gradient angle is more horizontal, the benefit of adding low-fidelity information is larger. It is important to note that, even when the angle is mostly vertical, *e.g.* 80°, adding low-fidelity information can still be beneficial if it is computationally much cheaper.

In the following, the direction of the error gradient is used to estimate the best split between high- and low-fidelity samples in order to reduce the modelling error. Even though the gradient is not consistent throughout the error grid, as can be seen by the curved contour lines, the global behaviour can be extracted by fitting a hyperplane through $\log_{10}$ of the MSE data according to

$$\log_{10}(\mathrm{MSE}) = \alpha + \beta_h n_h + \beta_l n_l. \tag{3}$$

Although clearly an approximation, the global direction provided by a simple linear model is sufficient for this purpose. From the linear fit, the global direction of the gradient direction of reducing error can be summarized intuitively by an angle

$$\theta = \arctan\left(\frac{\beta_h}{\beta_l}\right). \tag{4}$$

For the error grids in Figure 2, for example, this results in angles of $\theta_{\mathrm{Booth(2D)}} \approx 88°$, $\theta_{\mathrm{Currin(2D)}} \approx 34°$, $\theta_{\mathrm{Park91A(4D)}} \approx 72°$ and $\theta_{\mathrm{Borehole(8D)}} \approx 63°$, respectively. Confidence intervals of the calculated gradient angles are shown in Figure 3, as determined using the calculations shown in Appendix 1.

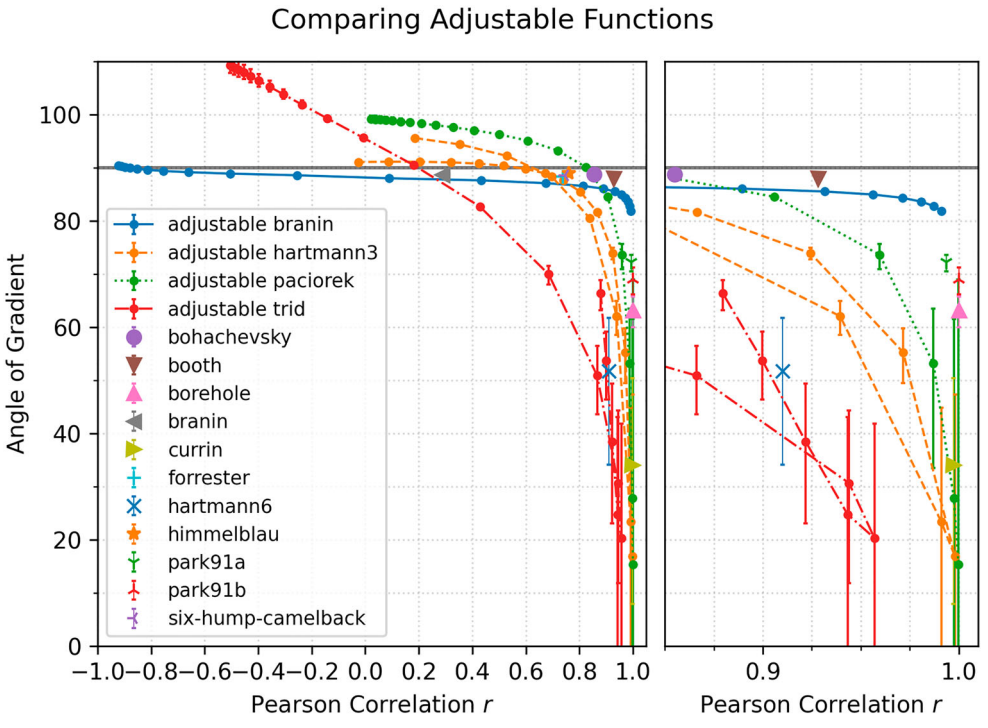


Figure 3. Angle as a function of correlation, illustrated for all functions in the `mf2` (van Rijn and Schmitt 2020) package. The left-hand side shows the complete range $-1 \leq r \leq 1$, while the right-hand side highlights the highly correlated region $0.85 \leq r \leq 1$. Single markers are used for non-adjustable functions such as Booth and Borehole, while the lines with markers show various parameter values for the adjustable functions where the line connects points with adjacent values of A. Error bars show the CI as defined in Equation (A3).[5,6]

5. Experiments

5.1. Replication of results and implementation details

All source code of this work is available on GitHub (van Rijn 2020), and archived on Zenodo (van Rijn 2021; van Rijn *et al.* "Generated Data Files and Figures" 2021) together with data files. The analyses are written in Python 3.6+, most notably using the packages `matplotlib` (Hunter 2007), `numpy` (van der Walt, Colbert, and Varoquaux 2011), `scikit-learn` (Pedregosa *et al.* 2011) and `xarray` (Hoyer and Hamman 2017).[7] Reproducibility is guaranteed by using a single fixed random seed for globally used random values such as the test set T and the initial DoE used for subsampling.[8] All experiments use $I = 50$ iterations.

5.2. Benchmark functions

The benchmark functions used from the `mf2` (van Rijn and Schmitt 2020) package range from one dimensional (1D), such as the Forrester function, to 10D, such as the Trid function, with a majority of 2D functions such as Bohachevsky, Currin and Six-Hump Camelback. This collection contains several different problem landscapes and all are previously used in the literature. With the exception of the 2D Branin function, the correlations between between the high- and low-fidelity functions are all above 0.7.

The adjustable benchmark functions previously proposed in Section 3 of Toal (2015) are focused on in particular: the 2D *adjustable* Branin function,[13] the 2D Paciorek, the 3D Hartmann3 and the 10D Trid functions. The low-fidelity functions of these benchmarks include a tuning parameter A,

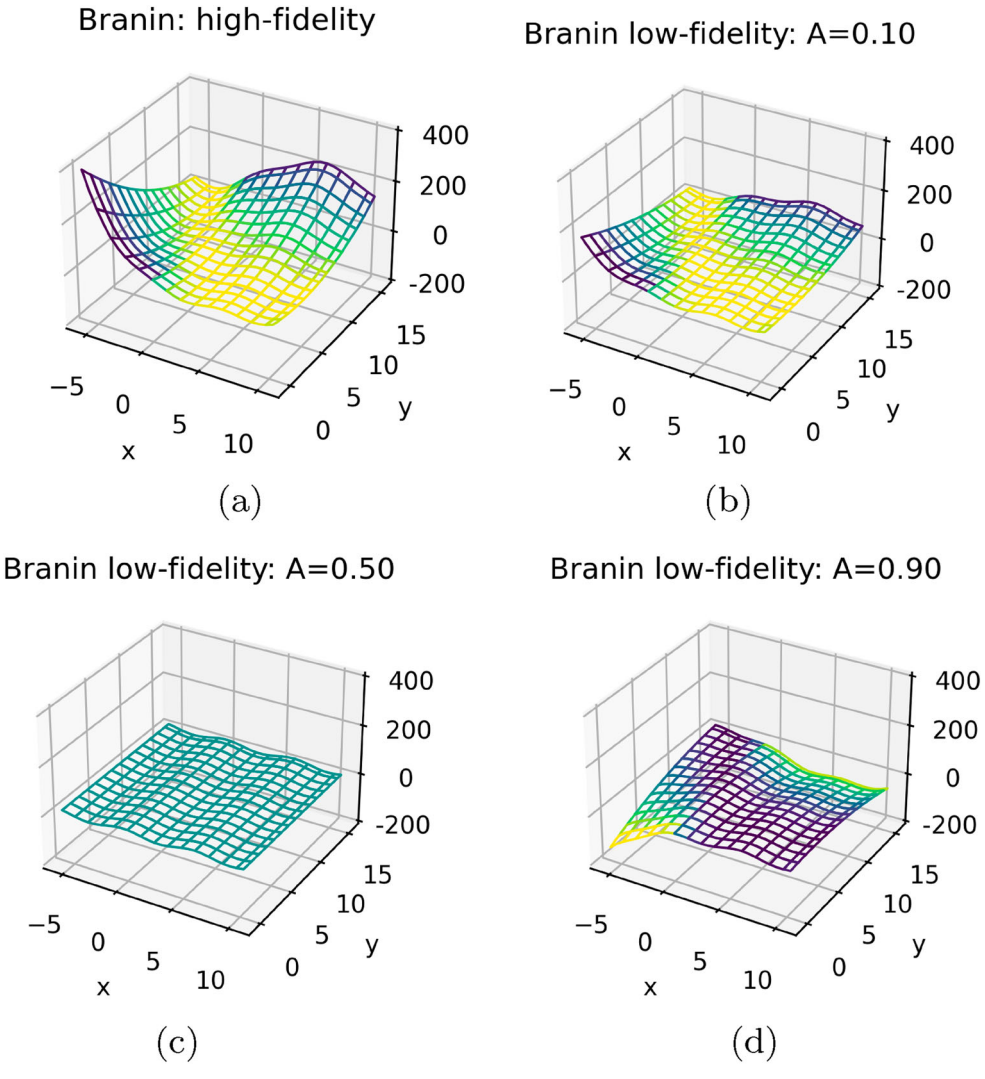


Figure 4. Adjustable multi-fidelity function example showing the adjustable Branin function: (a) high-fidelity; (b), (c) and (d) low-fidelity for $A = 0.1$, 0.5 and 0.9, respectively.[9,10]

which controls the correlation between high- and low-fidelity for these functions. Figure 4 shows an example of this, although the exact influence of A depends on the specific function, and the relationship between A and the correlation for these functions is shown in Figure 5. For all functions, the correlation can be tuned to any value between maximally positive ($r \approx 1$) and absent ($r \approx 0$). Additionally, for the Branin and Trid functions, this range extends to negative correlations ($r \approx -1$). The explicit functional forms can be found in the article by Toal (2015).

5.3. Error gradient angle analysis

The full enumeration procedure described in Section 4.1 was run for all `mf2` functions, using parameter values $A \in [0, 0.05, \ldots, 0.95, 1.0]$[14] for the adjustable functions. For each function, the gradient direction and error gradient angle was estimated using the linear fit procedure described

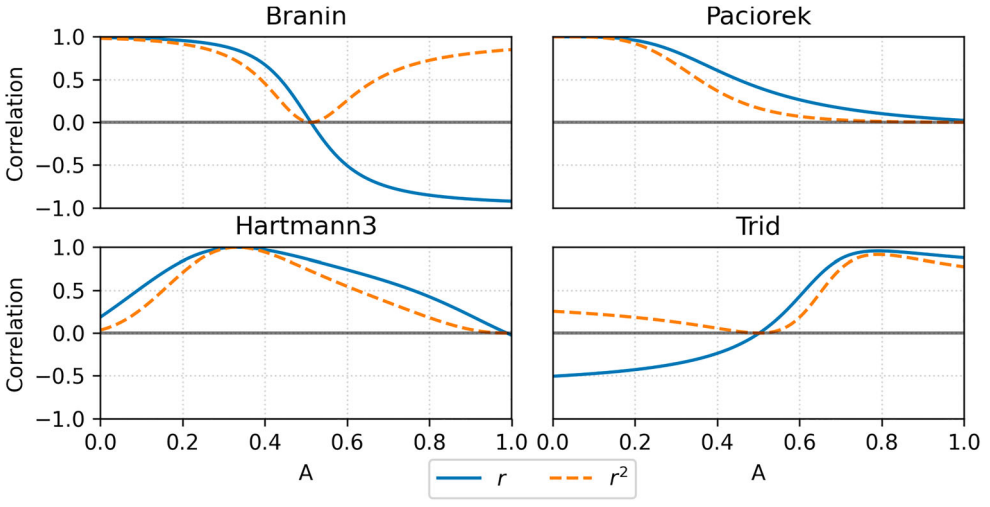


Figure 5. Correlation between high- and low-fidelity for adjustable 2D Branin, 2D Paciorek, 3D Hartmann3 and 10D Trid functions as a function of parameter A.[11,12]

in Section 4.3. The resulting angles, along with the confidence intervals, are shown in Figure 3 as function of the correlation r.

First, it should be noted that the calculated angles cover a very wide range from basically zero up to almost 120°, with the bulk of the values between 30° and 90°. For many functions and in a large range of correlations an angle of around 90° is computed, which indicates that the accuracy only increases when adding high-fidelity samples. Interestingly, angles $\geq 90°$ are also present. These high angles indicate that the added low-fidelity information actually hurts the accuracy of this hierarchical model, making it perform *worse* than a model trained with fewer low-fidelity samples. This adverse effect of increasing the error by adding low-fidelity samples occurs for correlations up to $r \lessapprox 0.8$ and the tendency gets stronger for less correlated or even anti-correlated benchmark functions. It should be noted that this is not an artifact of the linear fit or the way the error gradient angles are calculated, but truly reflects the behaviour of the model error for those functions, as can be seen in Figure 6 for the adjustable Trid function.

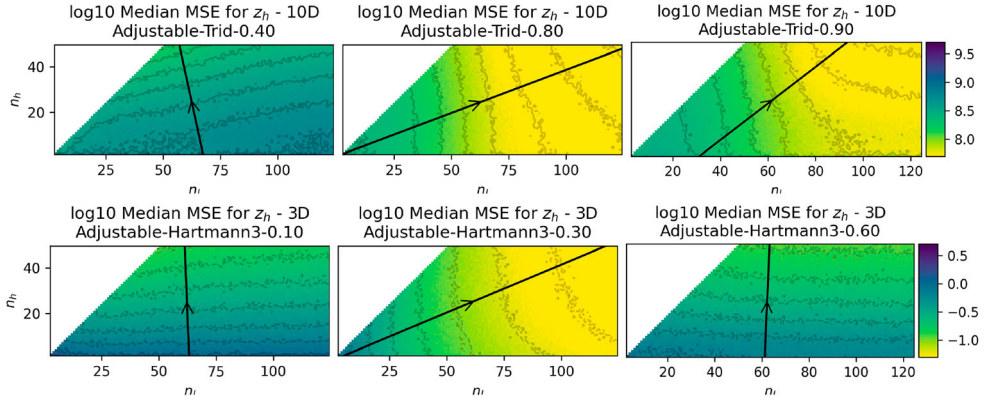


Figure 6. (a) Error grids for adjustable Trid function for $A = 0.4, 0.8, 0.9$, with $r = -0.23$, ;0.96, 0.92, respectively (upper row, from left to right) and (b) the adjustable Hartmann3 function for $A = 0.1, 0.3, 0.6$, with $r = 0.54, 0.99, 0.74$, respectively (lower row, from left to right); the black arrow shows the global gradient direction as described in Section 4.3.[15,16]

Figure 3 clearly shows a strong relationship between correlation coefficients and the error gradient angle. For each function, the observed gradient angle decreases for higher correlation coefficients. However, the exact values and the functional relationship differ vastly. Even for high correlation coefficients, *e.g.* $r \geq 0.9$, a large range of resulting error gradient angles can be observed. This means that a high correlation does not necessarily imply a low error gradient angle in the corresponding error grid.

Furthermore, the Hartmann3 and Trid functions show some other interesting behaviour. Note from Figure 5 that, within A's parameter range (*i.e.* $[0, 1]$), the correlation r for the Hartmann3 and Trid functions is not a bijection. Certain correlation values r are associated with two different gradient angles and vice versa, since different values for A can map to the same correlation r for the Hartmann3 and Trid functions (see Figure 5). The magnified section on the right-hand side of Figure 3 shows this most clearly. To help explain this, recall that the lines in the graph connect data points with adjacent values for A, not r. This behaviour can be visually confirmed by inspecting the error grids directly as shown in Figure 6.

These examples show that, although only a linear fit is used to extract gradient direction, the overall functional dependency of the model error is captured rather well. So, for the purpose of this work, exploring more complicated measures is not necessary. The proposed linear measures are accurate enough to capture the global tendencies, which can already provide insight and the possibility to formulate useful heuristics for practical application (see below).

5.4. Extrapolation

Given that the error grids from subsampling and their subsequent error gradient angles have been shown to match those from full enumeration quite well (see Appendix 2), this information can be used to answer the question posed in Section 3: how should additional computation budget be divided between the two available fidelity levels?

As discussed in Section 4.3, the linear fit slope β_h/β_l is assumed to indicate the direction of most improvement on the error grid. This implies that, if the error grid is extended with additional samples, the lowest model errors will be found in the same direction. So, when selecting n_h high- and n_l low-fidelity samples, the ratio between them should match the previously mentioned slope β_h/β_l to achieve the lowest model error, *i.e.*

$$\frac{n_h}{n_l} = \frac{\beta_h}{\beta_l}. \tag{5}$$

However, in order to apply this method, an initial DoE($n_{h,0}, n_{l,0}$) is required from which to create an error grid and calculate the gradient angle. The number of samples in this initial DoE can be obtained in any way, so does not have to match the ratio β_h/β_l. Since the global gradient angle of the error grid is considered when deciding how to select additional samples, there is no need to select the additional samples ($\Delta n_h, \Delta n_l$) in such as way as to bring the total (n_h, n_l) to match the calculated ratio. Instead, the most improvement is expected by having the additional number of samples ($\Delta n_h, \Delta n_l$) respect the relation

$$\Delta n_h = \frac{\beta_h}{\beta_l} \Delta n_l, \tag{6}$$

where $\Delta n_h = n_h - n_{h,0}$ and $\Delta n_l = n_l - n_{l,0}$ are the additional samples to be simulated. A fixed additional budget b can be split between high- and low-fidelity samples according to the cost ratio ϕ,

$$\Delta n_h + \phi \Delta n_l = b. \tag{7}$$

From Equations (6) and (7), the best number of additional low- and high-fidelity samples can be determined for a given additional computational budget b as

$$\Delta n_l = \frac{b\beta_l}{\beta_h + \phi\beta_l}, \tag{8}$$

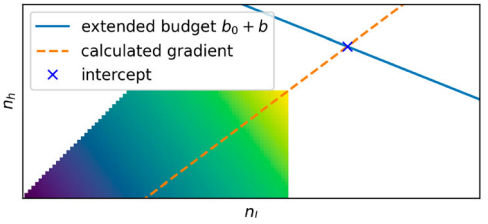


Figure 7. Schematic representation of the proposed method to determine the best split for a given additional budget b by extrapolating along the gradient of the error grid, through the upper rightmost point of the error grid until it intersects. The cost ratio for this example is $\phi = 0.4$.[17,18]

$$\Delta n_h = \frac{b\beta_h}{\beta_h + \phi\beta_l}. \tag{9}$$

Figure 7 shows the schematic representation of the proposed extrapolation method for splitting an additional budget b. The dashed line indicates the extension of the error grid along the direction of the gradient of the error grid. The solid line represents the line where the additional budget is spent according to the cost ratio. The intersection of both lines marks the proposed new samples split for the next DoE. Recall that the initial sample sizes $(n_{h,0}, n_{l,0})$ do not need to respect the cost ratio relation of Equation (7), as the initial DoE might be obtained by any method.

To evaluate this method, consider the example case of starting with a $(30, 75)$ initial DoE that is to be extended with an additional budget $b = 20$, assuming a cost ratio $\phi = 0.4$. First, an error grid as described in Section 4.2 is created, and the gradient is calculated as usual. This gradient predicts the division of additional samples according to Equations (8) and (9), with the size of the resulting DoE falling between $(50, 75)$ for $\Delta n_h = 20$ and $(30, 125)$ for $\Delta n_h = 0$.

For all DoE sizes between $(50, 75)$ and $(30, 125)$, the actual model error data from the full enumeration experiment described in Section 5.3 can be reused. This data is plotted in Figure 8 as a function of the gradient angle $\theta = \arctan(\Delta n_h / \Delta n_l)$, and compared with the predicted gradient angle.

Generally, it can be observed that the MSE values are quite noisy, even though the median MSE values of 50 different runs are plotted. This is due to the fact that the total number of samples is rather low for the investigated functions, leading to a generally large error and also large variations in error. But since the ultimate interest is in real-world problems, where a low number of (high-fidelity) samples is the norm, such noise is expected in these scenarios. The first three examples show that the predicted best angle, *i.e.* the error gradient angle from the subsampled error grid, matches roughly with the minimum measured MSE, regardless of whether this angle is high (90°) or low (0°). However, the last plot shows a case where the predicted angle does not lead to a region of low error but rather high error. This is likely to be an artifact of the linear fit to the error grid. If the error grid has a significantly different gradient in the region of low samples compared to the number of high samples, the linear fit matches the low-sample region and cannot accurately describe the region of a larger number of samples where the extrapolation is done. So the global estimate of the gradient of the error grid does not align with the local direction of decreasing error around the upper right of the error grid. This is the case for the Park91A function shown in Figures 2(c) and 8. The error grid of the initial DoE with $(n_{h,0}, n_{l,0}) = (30, 75)$ is best fitted by a linear function with a rather large 75° error gradient angle and consequently the extrapolation suggests sampling $(\Delta n_{h,0}, \Delta n_{l,0}) = (17, 7)$ additional points corresponding to that angle. However, only looking at the region above and to the right of the size of the initial DoE with $(n_{h,0}, n_{l,0}) = (30, 75)$ in the error grid of Figure 2(c) reveals that the direction of decreasing error is more along the n_l axis, *i.e.* an angle close to zero.

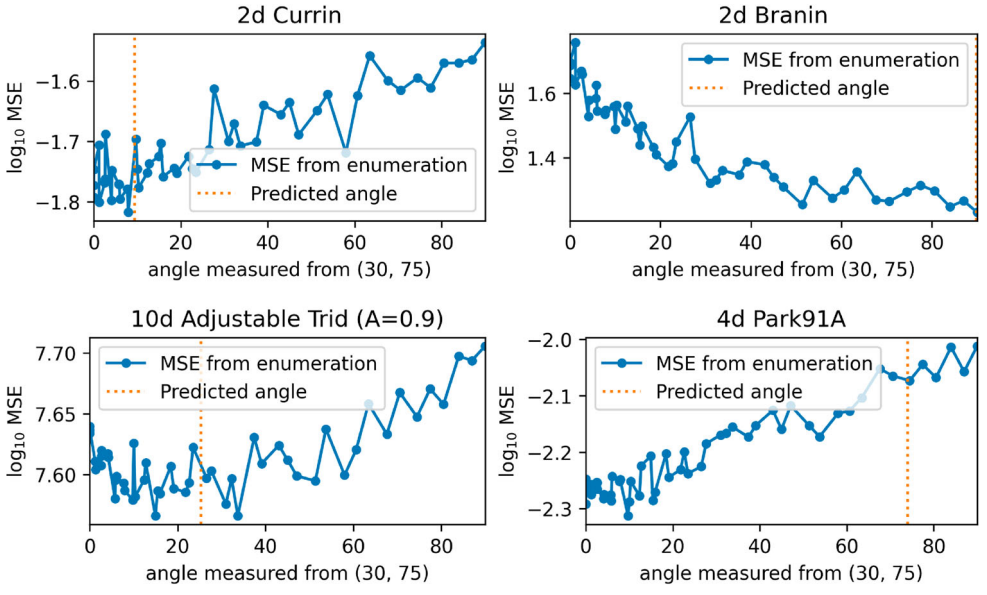


Figure 8. Median $\log_{10}$ MSE of DoE sizes along line $n_h + \phi n_l = 80\ (= b)$ (given $\phi = 0.4$) in the fully enumerated error grid from Section 4.1, shown on the x-axis as the angle measured from the initial sample point $(n_{h,0}, n_{l,0}) = (30, 75)$ for four benchmark functions. The dashed vertical line shows the angle for the proposed new sample split as calculated by Equations (8) and (9).[19,20]

This shortcoming of the extrapolation used to determine the split of an additional budget could be mitigated by limiting the region of the error grid to which the fit is done to a smaller area in the upper right of the error grid. By focusing on that region, the strong effect of small sample sizes is excluded, and the linear fit more accurately models the marginal benefit of adding another sample to the current set.

6. Conclusions

This work empirically examined the trade-off that exists in dividing computational budget between high- and low-fidelity samples in the context of multi-fidelity modelling and optimization problems.

So-called error grids were presented that are given by the modelling error of a hierarchical surrogate model for a DoE with a given number of high- and low-fidelity samples (n_h, n_l). For a complete error grid, the modelling error is evaluated for many DoEs with (n_h', n_l') sample points up to the size of the initial DoE, *i.e.* with $n_h' \in [2, n_h]$ and $n_l' \in (n_h', n_l]$. By this, the structure of the model error is revealed and the behaviour of the modelling error as function of the split between high- and low-fidelity samples can be analysed.

The global trend in the modelling error is captured by fitting a linear hyperplane through $\log_{10}$ of the mean squared errors. The linear fit easily lends itself to extracting the error gradient's global direction, which is used to identify the global direction of reducing error in the n_h–n_l-plane. Error grids were analysed for a multitude of benchmark functions, where some functions have a parameter that allows the tuning of the relation between low- and high-fidelity functions.

The first presented version of the error grid uses an independently sampled DoE for each hierarchical model with a given sample split. As this requires a very large number of independent function evaluations, a simple subsampling method was presented which needs only the available evaluations of an initial DoE in the spirit of cross-validation. It was shown for multiple benchmark functions that the direction of the gradient of the error grid can be estimated from the subsampling error grid reasonably well.

Based on the extracted global direction of the gradient of the modelling error, a simple scheme was proposed that allows an informed decision about how to divide additionally available evaluation budget between the different fidelities. It was shown that the scheme works well on most benchmark functions. Those cases where the predicted split of the additional budget did not extrapolate to a region with smaller model error are characterized by a change of the dominant behaviour of the error grid with the number of high- and low-fidelity samples. This shortcoming of the proposed method could be mitigated by performing the linear fit only to the region with the highest numbers of samples (*i.e.* the upper right part of the error grid) to increase the influence of the region of interest and at the same time reduce the sensitivity to a low number of samples.

Two main applications are envisioned: first, as a means of characterizing the behaviour of a fidelity level with respect to its accuracy—rather than relying on heuristics based on the correlation between fidelity levels, error grids can provide valuable initial insight into the benefit of additional samples from each fidelity level; secondly, as a possible means of online fidelity selection for multi-fidelity optimization use cases. The proposed approach determines the optimal division between the number of high- and low-fidelity samples given a set of samples. This can be utilized at each iteration of an optimization procedure to determine the split between high and low fidelity of the newly generated samples. The marginal benefit of each fidelity level will be reflected in the error grid's gradient direction, thereby steering the fidelity selection for the optimization.

In future work, experiments on additional benchmark functions and also real-world problems will have to be performed in order to confirm the benefits of the error grids for those applications. As only a hierarchical surrogate model based on a simplified additive co-kriging design was considered, other multi-fidelity models should be investigated, since the gradient angle is expected to to change according to the quality of the model fit. Additionally, the benefit of using the error grids and the extrapolation scheme for optimization use cases needs to be explored.

Notes

1. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/scripts/processing/2020-07-29-illustrated-bi-fid-doe.py
2. https://figshare.com/articles/figure/Finding_Efficient_Trade-offs_in_Multi-Fidelity_Response_Surface_Modeling_2020-07-29-illustrated-bi-fid-doe/14060912/2
3. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/scripts/processing/2019-09-19-plot-error-grids.py
4. https://figshare.com/articles/figure/Finding_Efficient_Trade-offs_in_Multi-Fidelity_Response_Surface_Modeling_2019-09-mse-nc/14060957/3
5. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/scripts/processing/2020-02-19-adjustable-gradients.py
6. https://figshare.com/articles/figure/Finding_Efficient_Trade-offs_in_Multi-Fidelity_Response_Surface_Modeling_2020-02-19-adjustable-gradients/14061017/3
7. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/requirements.txt
8. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/scripts/experiments/experiments.py#L35
9. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/scripts/processing/2021-11-25-plot-adjustable-surfaces.py
10. https://figshare.com/articles/figure/Finding_Efficient_Trade-offs_in_Multi-Fidelity_Response_Surface_Modeling_2021-11-25-plot-adjustable-surfaces/19188593/1
11. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/scripts/processing/2019-10-30-correlation-table.py
12. https://figshare.com/articles/figure/Finding_Efficient_Trade-offs_in_Multi-Fidelity_Response_Surface_Modeling_2019-10-correlation-exploration/14061014/3
13. Since Toal's adjustable Branin function differs from the non-adjustable version by Dong *et al.* (2015), they are explicitly differentiated by referring to Toal's version as *adjustable*.
14. For $A = 0$, the high- and low-fidelity versions of the Paciorek function are identical, so it is omitted.
15. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/scripts/processing/2019-10-15-plot-error-grids-adjustables.py
16. https://figshare.com/articles/figure/Finding_Efficient_Trade-offs_in_Multi-Fidelity_Response_Surface_Modeling_2019-10-07-adjustables/14061005/3
17. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/scripts/processing/2020-07-07-intercept-illustration.py

18. https://figshare.com/articles/figure/Finding_Efficient_Trade-offs_in_Multi-Fidelity_Response_Surface_Modeling_2020-07-07-intercept-illustration/14060960/2
19. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/scripts/processing/2020-07-06-extrapolation.py
20. https://figshare.com/articles/figure/Finding_Efficient_Trade-offs_in_Multi-Fidelity_Response_Surface_Modeling_2020-07-06-extrapolation/14061026/3
21. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/scripts/processing/2020-03-09-adjustables-subsampling-comparisons.py
22. https://figshare.com/articles/figure/Finding_Efficient_Trade-offs_in_Multi-Fidelity_Response_Surface_Modeling_2020-03-09-adjustables-subsampling-comparisons/14061020/3
23. https://github.com/sjvrijn/multi-level-co-surrogates/blob/v2/scripts/processing/2020-03-09-subsampling-gradients.py
24. https://figshare.com/articles/figure/Finding_Efficient_Trade-offs_in_Multi-Fidelity_Response_Surface_Modeling_2020-03-09-subsampling-gradients/14061023/2

Data Availability Statement

Source code [van Rijn(2021)] and data files [van Rijn et al.(2021b)] are publicly archived on Zenodo.

Disclosure statement

On behalf of all the authors, the corresponding author states that there is no conflict of interest.

Funding

This work is partly financed by the Netherlands Organisation for Scientific Research (NWO) and is part of the research program DAMIOSO [project number 628.006.002].

ORCID

Sander van Rijn https://orcid.org/0000-0001-6159-041X
Sebastian Schmitt https://orcid.org/0000-0001-7130-5483
Matthijs van Leeuwen https://orcid.org/0000-0002-0510-3549
Thomas Bäck https://orcid.org/0000-0001-6768-1478

References

Belakaria, Syrine, Aryan Deshwal, and Janardhan Rao Doppa. 2020. "Multi-Fidelity Multi-Objective Bayesian Optimization: An Output Space Entropy Search Approach." *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (6): AAAI-20 Technical Tracks 6. doi:10.1609/aaai.v34i06.6560.

Dong, Huachao, Baowei Song, Peng Wang, and Shuai Huang. 2015. "Multi-Fidelity Information Fusion Based on Prediction of Kriging." *Structural and Multidisciplinary Optimization* 51: 1267–1280. doi:10.1007/s00158-014-1213-9.

Durantin, Cédric, Justin Rouxel, Jean-Antoine Désidéri, and Alain Glière. 2017. "Multifidelity Surrogate Modeling Based on Radial Basis Functions." *Structural and Multidisciplinary Optimization* 56: 1061–1075. doi:10.1007/s00158-017-1703-7.

Fernández-Godino, M. Giselle, Sylvain Dubreuil, Nathalie Bartoli, S. Balachandar, and Raphael T. Haftka. 2019. "Linear Regression Based Multi-Fidelity Surrogate for Disturbance Amplification in Multi-Phase Explosion." *Structural and Multidisciplinary Optimization* 60: 220–2220. doi:10.1007/s00158-019-02387-4.

Fernández-Godino, M. Giselle, Chanyoung Park, Nam-Ho Kim, and Raphael T. Haftka. 2016. "Review of Multi-Fidelity Models." arXiv:1609.07196 [statAP]. http://arxiv.org/abs/1609.07196.

Forrester, Alexander I. J., András Sóbester, and Andy J. Keane. 2007. "Multi-Fidelity Optimization Via Surrogate Modelling." *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 463 (2088): 3251–3269. doi:10.1098/rspa.2007.1900.

Guo, Qi, Jiutao Hang, Suian Wang, Wenzhi Hui, and Zonghong Xie. 2020. "Design Optimization of Variable Stiffness Composites by Using Multi-Fidelity Surrogate Models." *Structural and Multidisciplinary Optimization* 63: 439–461. doi:10.1007/s00158-020-02684-3.

Hoyer, Stephan, and Joe Hamman. 2017. "xarray: N-D Labeled Arrays and Datasets in Python." *Journal of Open Research Software* 5 (1): 10. doi:10.5334/jors.148.

Huang, D., T. T. Allen, W. I. Notz, and R. A. Miller. 2006. "Sequential Kriging Optimization Using Multiple-Fidelity Evaluations." *Structural and Multidisciplinary Optimization* 32: 369–382. doi:10.1007/s00158-005-0587-0.

Hunter, John D. 2007. "Matplotlib: A 2D Graphics Environment." *Computing in Science & Engineering* 9 (3): 90–95. doi:10.1109/MCSE.2007.55.

Kennedy, M. C., and A. O'Hagan. 2000. "Predicting the Output From a Complex Computer Code When Fast Approximations Are Available." *Biometrika* 87 (1): 1–13. doi:10.1093/biomet/87.1.1.
Le Gratiet, Loic. 2013. "Multi-Fidelity Gaussian Process Regression for Computer Experiments." PhD diss., Sorbonne Université Paris-Diderot—Paris VII. https://tel.archives-ouvertes.fr/tel-00866770/.
Li, Nan, Eric W. M. Lee, Sherman C. P. Cheung, and Jiyuan Tu. 2019. "Multi-Fidelity Surrogate Algorithm for Fire Origin Determination in Compartment Fires." *Engineering with Computers* 36: 897–914. doi:10.1007/s00366-019-00738-9.
Liu, Yixin, Shishi Chen, Fenggang Wang, and Fenfen Xiong. 2018. "Sequential Optimization Using Multi-Level Co-Kriging and Extended Expected Improvement Criterion." *Structural and Multidisciplinary Optimization* 58: 1155–1173. doi:10.1007/s00158-018-1959-6.
Montgomery, Douglas C. 2019. *Design and Analysis of Experiments*. 10th ed. Wiley. https://www.wiley.com/en-us/Design+and+Analysis+of+Experiments%2C+10th+Edition-p-9781119492443.
Moss, Henry B., David S. Leslie, and Paul Rayson. 2020. "MUMBO: MUlti-Task Max-Value Bayesian Optimization." arXiv:2006.12093 [csLG]. http://arxiv.org/abs/2006.12093.
Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, *et al.* 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12 (85): 2825–2830. http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.
Pellegrini, Riccardo, Umberto Iemma, Cecilia Leotardi, Emilio F. Campana, and Matteo Diez. 2016. "Multi-Fidelity Adaptive Global Metamodel of Expensive Computer Simulations." In *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*. Piscataway, NJ: IEEE. doi:10.1109/CEC.2016.7744355.
Ryou, Gilhyun, Ezra Tal, and Sertac Karaman. 2020. "Multi-Fidelity Black-Box Optimization for Time-Optimal Quadrotor Maneuvers." arXiv:2006.02513 [csRO]. http://arxiv.org/abs/2006.02513.
Shi, Renhe, Li Liu, Teng Long, Yufei Wu, and G. Gary Wang. 2020. "Multi-Fidelity Modeling and Adaptive Co-Kriging-Based Optimization for All-Electric Geostationary Orbit Satellite Systems." *Journal of Mechanical Design* 142 (2): Article ID 021404. doi:10.1115/1.4044321.
Toal, David J. J. 2015. "Some Considerations Regarding the Use of Multi-Fidelity Kriging in the Construction of Surrogate Models." *Structural and Multidisciplinary Optimization* 51: 1223–1245. doi:10.1007/s00158-014-1209-5.
van der Walt, Stefan, S. Chris Colbert, and Gael Varoquaux. 2011. "The NumPy Array: A Structure for Efficient Numerical Computation." *Computing in Science & Engineering* 13 (2): 22–30. doi:10.1109/MCSE.2011.37.
van Rijn, Sander. 2020. "Multi-Level-Co-Surrogates." https://github.com/sjvrijn/multi-level-co-surrogates.
van Rijn, Sander. 2021. "sjvrijn/Multi-Level-Co-Surrogates." doi:10.5281/zenodo.6123254.
van Rijn, Sander, and Sebastian Schmitt. 2020. "MF2: A Collection of Multi-Fidelity Benchmark Functions in Python." *Journal of Open Source Software* 5 (52): Article ID 2049. doi:10.21105/joss.02049.
van Rijn, Sander, Sebastian Schmitt, Matthijs van Leeuwen, and Thomas Bäck. 2021. "Figures from Paper: Finding Efficient Trade-Offs in Multi-Fidelity Response Surface Modeling." doi:10.6084/m9.figshare.c.5311481.
van Rijn, Sander, Sebastian Schmitt, Matthijs van Leeuwen, and Thomas Bäck. 2021. "Generated Data Files and Figures: Finding Efficient Trade-Offs in Multi-Fidelity Response Surface Modeling." doi:10.5281/zenodo.6138077.
Zhou, Xunfei, Sheng-Jen Hsieh, and Jia-Chang Wang. 2019. "Accelerating Extrusion-Based Additive Manufacturing Optimization Processes with Surrogate-Based Multi-Fidelity Models." *The International Journal of Advanced Manufacturing Technology* 103: 4071–4083. doi:10.1007/s00170-019-03813-z.

Appendices

Appendix 1. Gradient angle confidence interval

From the linear fit of Equation (3), the standard errors for the linear fit parameters β_i associated with input feature n_i can also be calculated, *i.e.* the number of high- or low-fidelity samples n_h or n_l, as

$$se_{\beta_i} = \frac{\sqrt{\frac{\text{SSE}}{N_{\text{DoE}}-df}}}{\sqrt{\sum(n_i-\overline{n_i})^2}} = \frac{\sqrt{\frac{\sum(f_h(x)-z_h(x))^2}{N_{\text{DoE}}-df}}}{\sqrt{\sum(n_i-\overline{n_i})^2}}, \tag{A1}$$

where n_i can either be the number of high- or low-fidelity samples, $\overline{n_i}$ is the respective mean, df is the number of degrees of freedom, *i.e.* the number of samples N_{DoE} minus three for the number of parameters from the linear regression equation (α, β_h, β_l), and SSE is the sum of squared errors for the linear fit.

Using these standard errors, a 95% Confidence Interval (CI) for the slope β_h/β_l can be determined, from which the range of the angle can be estimated:

$$\text{CI}\,\frac{\beta_h}{\beta_l} = \frac{\beta_h}{\beta_l} \pm 1.96\sqrt{\left(\frac{se_{\beta_h}}{\beta_h}\right)^2 + \left(\frac{se_{\beta_l}}{\beta_l}\right)^2} \tag{A2}$$

$$\mathrm{CI}\,\theta \approx \left[\tan^{-1}\left(\frac{\beta_h}{\beta_l} - 1.96\sqrt{\left(\frac{se_{\beta_h}}{\beta_h}\right)^2 + \left(\frac{se_{\beta_l}}{\beta_l}\right)^2}\right),\right.$$
$$\left.\tan^{-1}\left(\frac{\beta_h}{\beta_l} + 1.96\sqrt{\left(\frac{se_{\beta_h}}{\beta_h}\right)^2 + \left(\frac{se_{\beta_l}}{\beta_l}\right)^2}\right)\right] \tag{A3}$$

This CI allows for more certainty of the global error gradient angle of the error grid. In any local section of the error grid, the angle can still be significantly different, but the proposed method provides a robust estimate of the average gradient and serves the purpose of discriminating the global behaviour of different benchmark functions.

Appendix 2. Subsample analysis

To validate the proposed method of reducing the number of necessary function evaluations for the error grid analysis described in Section 4.2, the influence of the sizes of training and test sets are explored. Results of the following three setups are compared.

(1) Independent full enumeration of training and test sets as described in Section 4.1.
(2) Subsampled training set and independent test set.
(3) Subsampled training set and left-over test set, *i.e.* full subsampling as described in Section 4.2.

If accurate enough, the third setup is the preferred approach for practical applications, as it uses any computational budget most efficiently by using each available evaluation for either test or training sets, and no further evaluations besides that.

Comparing (1) and (2) shows the dependence of the procedure's results on the initial DoEs used as the training set: does the spread of the subsamples cover the search space well enough to simulate independent DoEs of the subsample size? Comparison between (2) and (3) illustrates how accuracy tests with (much) less information influence the results.

Figure A1 shows example comparisons for the adjustable Branin, Hartmann3 and Trid functions. The ground-truth error grid (left) shows a mostly 90° gradient for $n_l \gg n_h$ with a trend toward 45° near the $n_h = n_l$ diagonal. The subsampling error grids (middle and right panels) are visibly noisier than the ground truth, but show similar shape

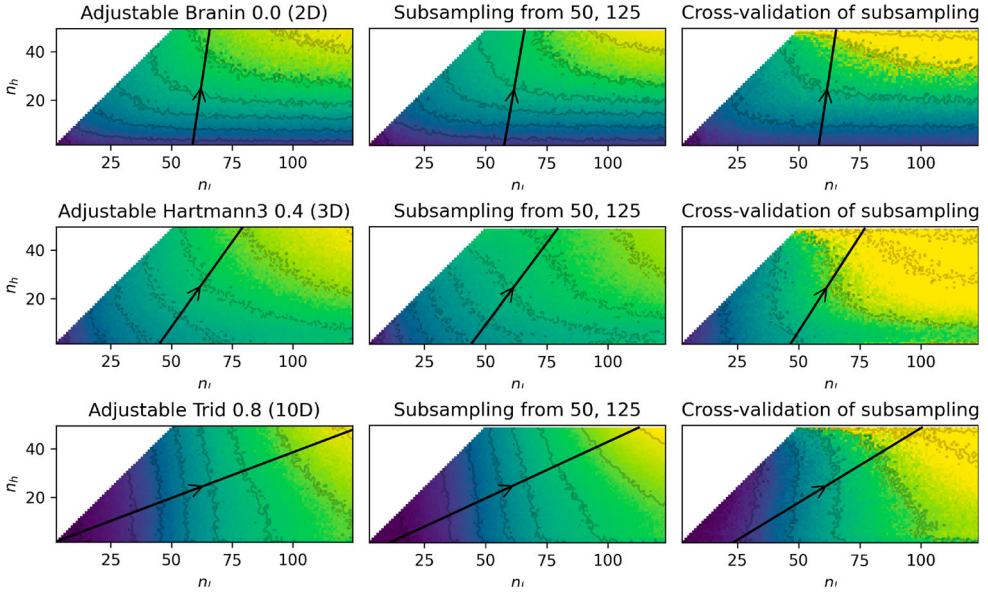


Figure A1. Comparison of error grids for (a) the adjustable Branin ($A = 0.0$, $r = 0.99$); (b) Hartmann3 ($A = 0.4$, $r = 0.97$); and (c) Trid ($A = 0.8$, $r = 0.96$) using different methods. Left: error results using independent training and test set (Section 4.1). The estimated error gradient angles, as illustrated by the black arrows, are $\theta = 81.8°, 55.4°, 20.4°$. Middle: error results using subsampled training set, with independent test set. The estimated error gradient angles are $\theta = 80.3°, 54.0°, 25.5°$. Right: error results using subsampled training set and left-over test set (Section 4.2). The estimated error gradient angles are $\theta = 81.7°, 58.0°, 32.2°$.[21,22]

characteristics. Despite the subtle differences in curvature, the resulting error gradient angles are very similar between 80° and 82°.

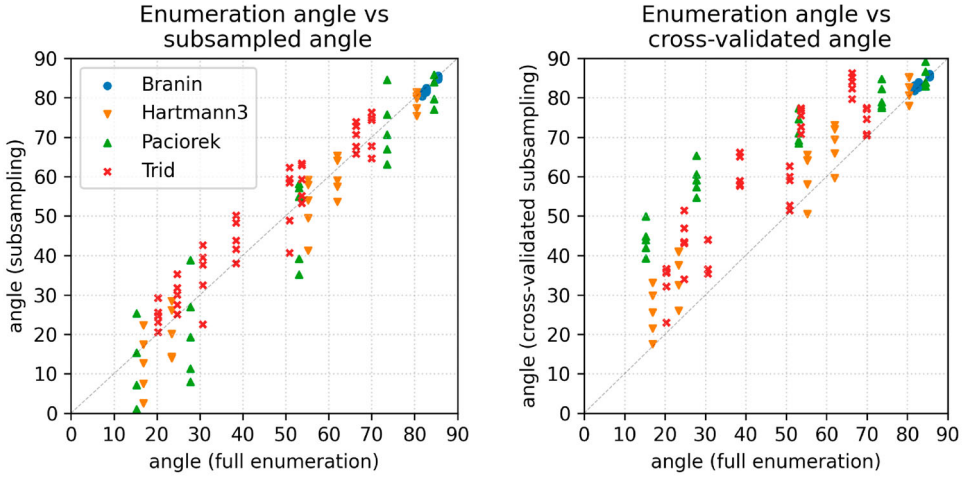
For a more detailed analysis, 21 (function, parameter) combinations as described in Table A1 are selected, covering a wide range of error gradient angles. For all cases, both subsampling procedures are repeated five times using independent initial DoEs in each case, and the error gradient angles are calculated. Figure A2 shows the correlation between the angles from the full enumeration and the subsampling procedures, as evaluated using both the independent large test set (left) and with the full subsampling approach with the left-over test set (right).

First considering Figure A2(a), it can be seen that that the angles from subsampling correspond very well with the ground-truth angles, with a spread of ±5° to ±15°, roughly symmetrical around the diagonal. The magnitude of the spread is visibly larger as the angles become smaller.

Figure A2(b) compares the ground-truth error gradient angles with those calculated using the test set based on cross-validation. Again, the variance in the estimated angle becomes smaller for larger angles. It is interesting to note that the spreads of the angles are also roughly between ±5° to ±15°, but there seems to be a systematic shifting of the angles estimated by full subsampling with a cross-validated test set to *higher* values. The exact shift is dependent on the underlying function, *e.g.* the Paciorek function having a much larger shift than the Hartmann3 function. So while not an exact prediction of the ground-truth error gradient angle, it can be interpreted as a worst-case estimate: the ground-truth error gradient angle is unlikely to be higher than what results from this procedure.

Table A1. Listing of all parameters that are used to compare correlation between error gradient angles from full enumeration and subsampling.

Function	Parameter values used for $A_1, \ldots, A_4$
Branin	0.00, 0.05, 0.25
Paciorek	0.05, 0.10, 0.15, 0.20, 0.25
Hartmann3	0.20, 0.25, 0.30, 0.35, 0.40
Trid	0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.00



(a) Subsampled training set, external test set (b) Subsampled training set, left-over test set

Figure A2. Error gradient angle correlation. The horizontal axis shows the angles as determined using the full enumeration procedure, while the vertical axis shows the angles calculated by the procedure mentioned by the caption below each figure. (a) Subsampled training set, external test set and (b) Subsampled training set, left-over test set.[23,24]