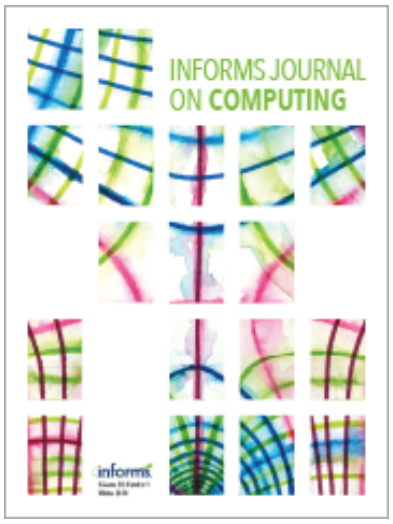



INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

# Bifidelity Surrogate Modelling: Showcasing the Need for New Test Instances

Nicolau Andrés-Thió, Mario Andrés Muñoz, Kate Smith-Miles

To cite this article:
Nicolau Andrés-Thió, Mario Andrés Muñoz, Kate Smith-Miles (2022) Bifidelity Surrogate Modelling: Showcasing the Need for New Test Instances. INFORMS Journal on Computing 34(6):3007-3022. https://doi.org/10.1287/ijoc.2022.1217

https://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Vol. 34, No. 6, November–December 2022, pp. 3007–3022
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

# Bifidelity Surrogate Modelling: Showcasing the Need for New Test Instances

**Nicolau Andrés-Thió,[a,*] Mario Andrés Muñoz,[a] Kate Smith-Miles[a]**

[a]School of Mathematics and Statistics, The University of Melbourne, Parkville, Victoria 3010 Australia
*Corresponding author

**Contact:** nandresthio@unimelb.edu.au, https://orcid.org/0000-0002-5417-8571 (NA-T); munoz.m@unimelb.edu.au, https://orcid.org/0000-0002-7254-2808 (MAM); smith-miles@unimelb.edu.au, https://orcid.org/0000-0003-2718-7680 (KS-M)

**Received:** October 20, 2021
**Revised:** February 21, 2022; May 24, 2022; May 30, 2022
**Accepted:** June 3, 2022
**Published Online in Articles in Advance:** August 9, 2022

**https://doi.org/10.1287/ijoc.2022.1217**



**Abstract.** In recent years, multifidelity expensive black-box (Mf-EBB) methods have received increasing attention due to their strong applicability to industrial design problems. The challenge, however, is that knowledge of the relationship between decisions and objective values is limited to a small set of sample observations of variable quality. In the field of Mf-EBB, a problem instance consists of an expensive yet accurate source of information, and one or more cheap yet less accurate sources of information. The field aims to provide techniques either to accurately explain how decisions affect design outcome, or to find the best decisions to optimise design outcomes. Many techniques that use surrogate models have been developed to provide solutions to both aims. Only in recent years, however, have researchers begun to explore the conditions under which these new techniques are reliable, often focusing on problems with a single low-fidelity function, known as bifidelity expensive black-box (Bf-EBB) problems. This study extends the existing Bf-EBB test instances found in the literature, as well as the features used to determine when the low-fidelity information source should be used. A literature test suite is constructed and augmented with new instances to demonstrate the potentially misleading results that could be reached using only the instances currently found in the literature, and to expose the criticality of a more heterogeneous test suite for algorithm assessment. Addressing the shortcomings of the existing literature, a new set of features is presented, as well as a new instance creation procedure, and a study of their impact on algorithm assessment is conducted. The low-fidelity information source is shown to be valuable if it is often locally accurate, even when its overall accuracy is relatively low. This contradicts the existing literature guidelines, which indicate the low-fidelity information is only useful if it has a high overall accuracy.

**History:** Accepted by Antonio Frangioni, Area Editor for Design & Analysis of Algorithms – Continuous.

**Funding:** This work was supported by Australian Research Council [Grant IC200100009] for the ARC Training Centre in Optimisation Technologies, Integrated Methodologies and Applications (OPTIMA), and the University of Melbourne Research Computing Services and Petascale Campus Initiative. N. Andrés-Thió is also supported by a Research Training Program scholarship from the University of Melbourne.

**Supplemental Material:** The software that supports the findings of this study is available within the paper and its Supplementary Information [https://pubsonline.informs.org/doi/suppl/10.1287/ijoc.2022.1217] or is available from the IJOC GitHub software repository (https://github.com/INFORMSJoC) at [http://dx.doi.org/10.5281/zenodo.6578060].

**Keywords:** surrogate modelling • expensive black-box • Co-Kriging • Kriging • multifidelity

## 1. Introduction

Design problems within a wide range of industries often involve measurable design outcomes for which no analytical expression exists. Design outcomes of this type are known as *black-box* functions. This term denotes the fact that the exact relationship between the decision variables and the design outcome is unknown, and the only way to evaluate the outcome for a new decision point is through the use of a deterministic procedure. It is often the case that these functions are also *expensive*, meaning that sampling the function has a high cost, measured in terms that are either computational, monetary, or temporal. The lack of analytical expression and high sampling cost can be seen when designing and producing a prototype such as a battery or a plane (Forrester 2010), or performing a lengthy software simulation such as a weather or medical model (Aleman et al. 2009).

Problems of this type are known as expensive black-box (EBB). Algorithms developed for these problems often aim to better understand the relationship between the design variables and outcome in order to accurately predict the outcome value in regions not yet sampled. Surrogate models have been developed in the past and successfully used in EBB problems, either for the purpose of uncertainty quantification and reduction, or when performing design optimisation. Perhaps the two best-known surrogate modelling methods are Kriging (Krige 1951, Matheron 1963, Jones 2001) and radial basis functions (RBFs) (Duchon 1977, Gutmann 2001, Regis and Shoemaker 2007, Wild et al. 2008, Müller and Shoemaker 2014). Such methods consist of training a surrogate model on the sparse available data in order to guide the sampling strategy of the algorithm. This strategy can be a global exploration strategy when training a surrogate model (and thus sampling in regions where the model is uncertain of its accuracy), or a balance between global exploration and local exploitation when optimising an objective function. The latter consists of balancing the need for exploration of yet unexplored areas in the sample space for further model training, and the opportunity for exploitation of promising regions via further sampling to reveal optimal solutions.

In many application domains, it is often the case that multiple sources of information exist for the objective function, with varying degrees of cost and accuracy. A survey by Fernández-Godino et al. (2019) shows many potential reasons for different levels of cost and accuracy, including the simplification of a mathematical model, an increase in model coarseness, and the difference between simulations and experiments. For instance, in plane design it might be possible to use a cheap but potentially inaccurate physics simulation engine to assess the performance of a candidate design, whilst the true performance can only be assessed via the more costly construction of a prototype and wind tunnel testing (Forrester 2010). These types of problems are known as multifidelity expensive black-box (Mf-EBB) problems, with its simplest variant known as bifidelity expensive black-box (Bf-EBB) problems. An instance of this class of problems is defined by its two sources of information, namely $f_h$ and $f_l$, defined as

$$f_h : \Omega \to \mathbb{R}$$
$$f_l : \Omega \to \mathbb{R}$$

where $\Omega$ is the sample space and is normally defined as a hypercube

$$\Omega = [x_1^l, x_1^u] \times \cdots \times [x_d^l, x_d^u].$$

Here, $\mathbf{x}^l = (x_1^l, \ldots, x_d^l)^\top$ and $\mathbf{x}^u = (x_1^u, \ldots, x_d^u)^\top$ are the vectors representing the lower and upper bounds of $\Omega$, and $d \in \mathbb{N}$ is the dimension (i.e., number of variables) of the problem. The high-fidelity function $f_h$ represents an accurate yet expensive source of information, and the low-fidelity function $f_l$ represents a cheaper but less accurate source of information. It is worth noting that in the literature $f_h$ is sometimes assumed to be a noisy (i.e., stochastic) function. This is however beyond the scope of this study, and as such $f_h$ is assumed to be deterministic for the remainder of the paper. Furthermore, despite being a less accurate representation of $f_h$, the low-fidelity function $f_l$ is also assumed to be deterministic.

As $f_h$ is considered to be expensive, the cost to sample the function is assumed to dominate the computational time required by any algorithm solving this type of problem. Therefore, the computational time of algorithms is not typically taken into consideration when assessing their performance. Instead, a sampling budget $B$ is specified, determining the maximum amount of sampling allowed by any given algorithm. The function $f_l$ is considered to be a cheaper (but still relatively expensive) function, and as such, a constant $0 < C_r < 1$ is given that represents the cost of $f_l$ relative to the cost of $f_h$. For instance, a value $C_r = 0.1$ implies the cost of a single evaluation of $f_h$ is the same as evaluating $f_l$ a total of 10 times. If a given algorithm has sampled $f_h$ and $f_l$ a total of $n_h$ and $n_l$ times, respectively, the total budget used is given by $n_h + C_r n_l$.

Surrogate model methods have been adapted to Bf-EBB problems, including Co-Kriging (Kennedy and O'Hagan 2000, Forrester et al. 2007) and RBFs methods (March and Willcox 2012, Durantin et al. 2017, Müller 2020). Due to the importance of the choice of surrogate model in these methods, many studies are devoted to studying the comparative accuracy of different models given a fixed sample of $f_h$ and $f_l$ (Dong et al. 2015, Toal 2015, Park et al. 2017, Liu et al. 2018c, Shi et al. 2020), as well as the impact of the chosen training procedure of a scaling parameter on overall model quality (Park et al. 2018). It is worth noting that any (single-source) EBB algorithm can be also applied to a Bf-EBB problem by simply working with the function $f_h$ and ignoring the extra information provided by $f_l$. Whilst a variety of algorithms exist in the literature, Co-Kriging in particular has taken a prominent position in the field. This can likely be attributed to its strong theoretical backing, and the fact that Co-Kriging provides specific methods for a variety of aims. These include surrogate model fitting via the training of hyperparameters, exploration via the use of a measure of model uncertainty, and an exploration/exploitation balance measure that maximises the expected function improvement of further sampling.

Underlying most of the literature is the assumption that bifidelity methods should always be applied to Bf-EBB problems, thus assuming a minimum threshold in the quality of $f_l$. In other words, if a low-fidelity

function is available, it should be used, be it to train a surrogate model or when using optimisation algorithms. David Toal (2015, p. 1223) was one of the first to challenge this assumption by asking "when exactly should a designer select a multifidelity approach over a single fidelity approach and vice versa?" In his study, he compared the accuracy of (single-source) Kriging models with the accuracy of (two-source) Co-Kriging models for a variety of instances (i.e., pairs of functions $(f_l, f_h)$) given a fixed sample, and showed that the correlation between $f_h$ and $f_l$ can have a significant impact on the quality of a Co-Kriging model. In fact, he showed that in some cases, it would be better to ignore the low-fidelity function and train a Kriging model instead. Toal established some guidelines on when a Co-Kriging model should be trained, including the requirement that $f_l$ and $f_h$ are highly correlated. If this requirement is not satisfied, it is possible that a Co-Kriging model will be less accurate than a Kriging model based only on $f_h$.

Toal's work highlights two important and interrelated questions pertaining to Bf-EBB problems:

i. What are the properties required of a low-fidelity function to ensure a bifidelity model performs no worse than a single-source model?

ii. How can we develop an algorithm or rules to choose when to use a bifidelity or single-source model?

Despite Toal's work bringing attention to these important questions, analysis of the performance of algorithms within the context of variations in low-fidelity function quality has only gained traction in very recent years. A survey by Fernández-Godino et al. (2019), which focused on the topic of building multifidelity surrogate models for optimisation, did not discuss the impact of function quality; nor did another survey by Liu et al. (2018a), which focused on sampling strategies leading to accurate multifidelity surrogate models. Fernández-Godino et al. (2019, p. 2042) stated that "more research into the choice between high-fidelity alone surrogate and multifidelity surrogate is clearly called for." Whilst most studies properly justify the choice of test instances, an analysis of their diversity is often missing (Forrester et al. 2007; Rajnarayan et al. 2008; March and Willcox 2012; Liu et al. 2016, 2018b, c; Ruan et al. 2020; Wu et al. 2020; Zhou et al. 2020, 2021). This can lead to a choice of instances for which the multifidelity algorithm performance is unsurprising and consistent with Toal's findings (Toal 2015): either performing poorly when $f_l$ is chosen as an unrelated function (March and Willcox 2012), or performing better since $f_l$ functions are frequently either chosen as highly correlated industry test problems (Shahpar et al. 2011) or created via a small perturbation of $f_h$ (Rajnarayan et al. 2008).

Some very recent studies, however, extend the work of Toal when performing algorithm analysis. These studies analyse both the qualities of their chosen test instances by measuring the correlation between $f_h$ and $f_l$, and the overall discrepancies between the two, in order to assess whether a wide set of instances have been chosen to conduct a thorough algorithm analysis (Wang et al. 2017, Song et al. 2019, Müller 2020, Shi et al. 2020, Lv et al. 2021, van Rijn et al. 2022). The work of both Müller (2020) and van Rijn et al. (2022) is of particular interest as they presented two of the very few adaptive techniques (the first in surrogate model training with guided sampling, the second in function optimisation) that choose to sample $f_l$ only if it seems beneficial to do so, rather than assuming $f_l$ should always be used. The need for such techniques arises from the large impact the quality of $f_l$ can have on algorithm performance. This further indicates the need for accurate measures of the quality of $f_l$ relative to $f_h$, and of instances that can accurately assess the performance of these algorithms. The need for more instances in the field is highlighted by the work of Wang et al. (2017), which proposes a creation procedure of industry-like Mf-EBB instances with a single parameter that specifies the quality of $f_l$.

This study argues the need for further work in both instance creation and instance characterisation in the field of Bf-EBB. Both are important prerequisites to analysing the types of instances that a two-source algorithm can be used for, and developing prediction techniques that can correctly choose which method to use. As such, their importance will be showcased through the analysis of single-source versus two-source surrogate model accuracy in the form of Kriging versus Co-Kriging, similar to the work of Toal (2015). Performing this analysis twice using different instance test suites will reveal the need for new instances and new instance measures, or features, both of which are put forward in this work.

The remainder of this paper is structured as follows. Section 2 presents the existing types of instances and features in the literature, chooses more than 200 instances to create a literature test suite, and conducts a performance analysis of Kriging and Co-Kriging algorithms using this test suite. Section 3 then proposes new features and instances that are used to create an augmented test suite. This new test suite is used for a second algorithm performance analysis, leading to results in direct disagreement with those found in Section 2. Section 4 presents a discussion of the differences in results obtained based on the chosen test suite and highlights the importance of the presented features and instance creation procedure. Section 5 concludes the study with some closing remarks and direction of future work. To make the results

easily reproducible, both the code and the data used in the research are available on GitHub (Andrés-Thió 2022) and FigShare (Andrés-Thió et al. 2022), respectively. Both the code and the data as well as an appendix giving a formal description of Kriging and Co-Kriging are also available on the IJOC GitHub site (Andrés-Thió et al. 2021).

## 2. Existing Test Suites

Synthetic Bf-EBB problem instances are defined in part by a pair of functions $(f_l, f_h)$. Many such pairs exist in the literature, as studies presenting new techniques often perform their experimental analysis using newly created instances. The standard approach in creating such a pair of functions is the use of a well-known test function such as the Branin or Hartmann functions being assigned to $f_h$, with $f_l$ being defined as a modification of $f_h$ via the addition of some extra terms or the modification of the coefficients of $f_h$. Three types of literature instances, namely *fixed*, *parameter-based*, and *error-based*, are implemented in this study to create a literature test suite.

The term fixed can be used to describe the most common type of instance found in the literature. Each of these instances is the result of some modification being applied to $f_h$ resulting in a unique $(f_l, f_h)$ pair, as opposed to a family of instances as is the case in the other two classes described later. How this modification is added can greatly vary from study to study, from adding a function to $f_h$ to represent a small error term (Rajnarayan et al. 2008) to assigning an unrelated function to $f_l$ (March and Willcox 2012). A total of 43 distinct functions of this type with varying dimension ($d \in \{1,2,3,4,5,6,8,10,20\}$) from 10 different studies (Rajnarayan et al. 2008; March and Willcox 2012; Xiong et al. 2013; Dong et al. 2015; Liu et al. 2016, 2018c; Park et al. 2017; Shi et al. 2020; Surjanovic and Bingham 2020; Wu et al. 2020) are implemented here.

The set of parameter-based instances are of the kind first presented by Toal (2015), which is followed in the work of Song et al. (2019). This type of instances are defined as a family of instances by defining a fixed $f_h$ function and a parametrised $f_l$ function. By varying the single parameter $A$ of $f_l$, new $(f_l, f_h)$ pairs are created with varying $f_l$ quality. An example is Toal's Branin function, defined for $\Omega = [-5,10] \times [0,15]$:

$$f_h(\mathbf{x}) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10,$$

$$f_l(\mathbf{x}) = f_h(\mathbf{x}) - (A + 0.5)\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 \quad A \in [0,1].$$

By varying the parameter $A$ with the values $\{0, 0.1, \ldots, 0.9, 1.0\}$, 11 different instances can be generated from a single definition. A further 77 such instances with different dimensions ($d \in \{1,2,4,10\}$) are added to the literature test suite (seven families of instances with 11 different $A$ values each).

Finally, another set of instances defined by Wang et al. (2017) is also added to the literature test suite. While studying an evolutionary algorithm for multifidelity optimisation, the authors argued the need for new test instances for these types of algorithms, and provided a systematic way to create a $(f_l, f_h)$ pair by adding so-called stochastic, instability, or resolution errors to $f_h$. Neither stochastic nor instability errors are implemented in this study, as the former define nondeterministic functions, and the latter define functions which might fail to evaluate. On the other hand, resolution errors are deterministic and therefore highly relevant to this study. Resolution errors are used to create a framework for instance creation based on industry problems where the low-fidelity function $f_l$ is taken to represent a simulation model, with its accuracy and cost able to be varied by changing the resolution of the model. As such, given a function $f_h$, $f_l$ is created by adding one of four resolution errors with parameter $\phi \in [0, 10000]$. The authors showcase their method with the Rastrigin function, for which a desired function degree $d$ can be specified. Setting the dimension to 5 and 10 in combination with all four resolution errors and for $\phi$ values $\{0, 1000, \ldots, 9000, 10000\}$, another 88 instances are added to the literature test suite, ultimately containing a total of 208 instances already studied in the literature.

### 2.1. Existing Features

The work of Toal (2015) was the first to bring attention to the impact that different instance qualities can have on the performance of a given algorithm. His work presented two measures (in this work denoted as *instance features*) for the analysis of pairs of functions $(f_l, f_h)$, namely correlation coefficient (CC) and root mean squared error (RMSE). Note that these two measures are calculated after heavily sampling both $f_h$ and $f_l$. In this study, a sample of size $1{,}000d$ is chosen via Latin hypercube sampling (LHS). Taking such a large sample cannot be done in practice, however, as the sample budget is limited. The aim here is to study the effect of different instance features on algorithm performance in an artificial setting, where the analytical expressions for both $f_h$ and $f_l$ are known and sampling them can be done at little to no cost. Once the features have been measured, these functions are treated as expensive black-box functions when assessing algorithm performance.

Both features are calculated on a given set of sample points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \Omega$, evaluated both at $f_l$ and $f_h$. Therefore, the two available sets are $\mathbf{Y}_l = \{Y_1^l, \ldots, Y_n^l\}$ and $\mathbf{Y}_h = \{Y_1^h, \ldots, Y_n^h\}$, with $Y_i^l = f_l(\mathbf{x}_i)$ and $Y_i^h = f_h(\mathbf{x}_i)$. The measures are given by

$$RMSE = \left[\frac{\sum_{i=1}^{n}(Y_i^l - Y_i^h)^2}{n}\right]^{1/2},$$

$$CC = \left[\frac{1}{n-1}\left(\frac{\sum_{i=1}^{n}(Y_i^l - \bar{Y}_l)(Y_i^h - \bar{Y}_h)}{s_{Y_l} s_{Y_h}}\right)\right]^2,$$

where $$\bar{Y}_l = \frac{1}{n}\sum_{i=1}^{n} Y_i^l \qquad s_{Y_l} = \left[\frac{\sum_{i=1}^{n}(Y_i^l - \bar{Y}_l)^2}{n-1}\right]^{1/2},$$

$$\bar{Y}_h = \frac{1}{n}\sum_{i=1}^{n} Y_i^h \qquad s_{Y_h} = \left[\frac{\sum_{i=1}^{n}(Y_i^h - \bar{Y}_h)^2}{n-1}\right]^{1/2}.$$

Note that Toal's *CC* feature is defined as the square of the sample correlation of $\mathbf{Y}_l$ and $\mathbf{Y}_h$. As such, $0 \le CC \le 1$, and a high *CC* value indicates $f_l$ is correlated with $f_h$ and therefore behaves similarly. Note also that the *RMSE* feature can vary greatly between instances as it is affected by the range of $f_h$ and $f_l$. As such, in this study the relative *RMSE* (*RRMSE*) feature will be used instead to allow comparison between instances

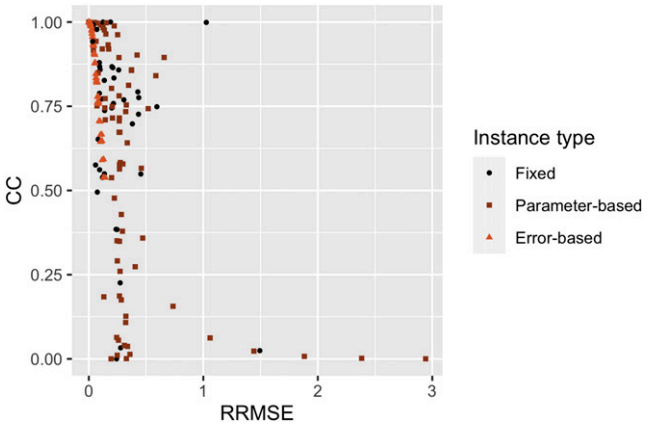
$$RRMSE = \frac{RMSE}{\max\{\mathbf{X}_h\} - \min\{\mathbf{X}_h\}}.$$

Figure 1 shows a plot of all 208 instances of the literature test suite based on their *CC* and *RRMSE* feature values.

## 2.2. Algorithm Performance Analysis on Literature Test Suite

The created literature test suite can be used to assess the comparative performance of Kriging and Co-Kriging (see Andrés-Thió et al. 2021) for the aim of surrogate model fitting, that is, when fitting as accurate a surrogate model as possible to the available data. To do so, an experimental setup similar to that of Toal (2015) is used. For a given instance of dimension $d$, a total sample budget of $5d$ is used for both Kriging and Co-Kriging. In the case of Kriging, the whole budget is used to sample $f_h$; that is, a sample of $f_h$ of size $5d$ is created using LHS and then used to fit the surrogate model. In the case of Co-Kriging, a budget of $d$ samples is used to sample $f_l$, with the remaining $4d$ used to sample $f_h$. These samples are chosen using first LHS, and then a subset $f_h$ sample is chosen using the Morris-Mitchell criterion, in a procedure presented by Forrester et al. (2007). Toal analyses algorithm performance for different cost ratio $C_r$ values; in this study, however, a value $C_r = 0.1$ is fixed, leaving further analysis of the impact of this constant to future work. Therefore, for an instance of dimension 5, a Kriging model is fitted using 25 samples of $f_h$, and a Co-Kriging model is fitted using 20 samples of $f_h$ and 50 samples of $f_l$.

**Figure 1.** (Color online) Instances in the Literature Test Suite Plotted Based on their Correlation Coefficient (*CC*, *y*-axis) and Relative Root Mean Squared Error (*RRMSE*, *x*-axis) Values

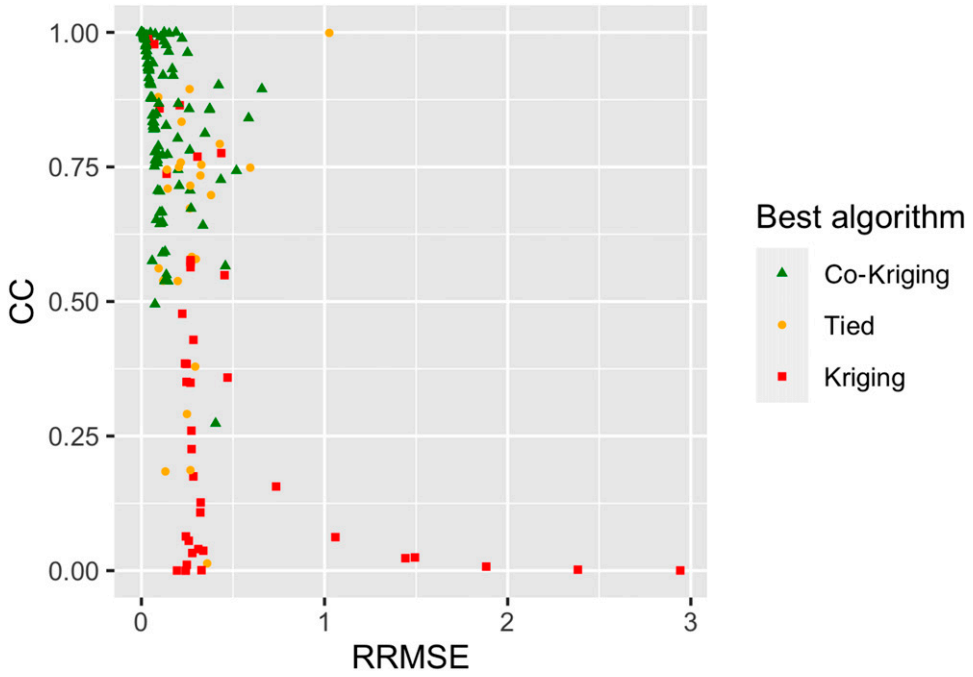



*Note.* Each plot point represents an instance, that is, a pair of functions ($f_l$, $f_h$), and a set of 1,000$d$ points is used to calculate the features of each instance, where $1 \le d \le 20$.

The performance of the two techniques on a given instance is compared by assessing the statistical performance of 20 runs per algorithm. All experiments were run using the Spartan computer cluster (for details, see funding). The performance of an algorithm on a single run is measured in the form of the error between the constructed surrogate model and the function $f_h$. This is measured by taking 1,000$d$ samples of both the surrogate model and $f_h$, and calculating the *RRMSE* value between the two samples. The 20 runs of each algorithm are then compared using a two-tailed Wilcoxon test (Wilcoxon 1992), with the null hypothesis being that the two sets of errors are not statistically significantly different from one another. The two algorithms are said to perform equally well if the null hypothesis cannot be rejected with higher than 95% accuracy. If the null hypothesis is rejected, the algorithm with the smaller median error is said to perform best. A plot representing which algorithm performs best for each instance is shown in Figure 2, which seems to indicate that for $CC \le 0.5$, Co-Kriging is ill-advised. This supports the findings of Toal (2015) regarding the strong impact the feature *CC* has on algorithm performance, despite a discrepancy on the critical *CC* value at which Co-Kriging is discouraged. Toal concluded that Co-Kriging should be used for $CC \ge 0.9$ whereas Figure 2 seems to indicate that Co-Kriging could be used for $CC \ge 0.5$, although this difference may be due to the choice of $C_r$ and budget in this study.

To further evaluate the impact of different features on algorithm performance, a machine learning method is used to infer when each algorithm is expected to perform better than the other. The question being answered is when to ignore the low-fidelity source of information as harmful and use Kriging instead of Co-Kriging. Therefore, the prediction accuracy is assessed in terms of the accuracy predicting for which

**Figure 2.** (Color online) Comparative Performance of Kriging and Co-Kriging on the Literature Test Suite




*Notes.* Each plot point represents an instance, that is, a pair of functions ($f_l, f_h$) and its position is based on the features correlation coefficient (*CC*, *y*-axis) and relative root mean squared error (*RRMSE*, *x*-axis). The colour/shape represents which technique performed best, either Kriging (red square), Co-Kriging (green triangle) or statistically equal performance (yellow circle).

instances Co-Kriging can be used (i.e., Co-Kriging performs no worse than Kriging, shown as green triangles and yellow circles in the plots), and for which it cannot be used (i.e., Kriging outperforms Co-Kriging, red squares in the plots). The literature test suite is randomly divided into a training and a testing set, the former containing 80% and the later 20% of the data. A decision tree is trained, with its accuracy assessed on the testing set. This cross-validation procedure is repeated 100 times for each technique and the average accuracy calculated.

Two basic prediction rules are used as baseline comparisons. The first is the simple rule to always use Co-Kriging. The second is a majority rule; that is, the same algorithm is always used based on the one that has superior accuracy on the training data. For example, if Co-Kriging should only be used in 30% of the instances in the training set, the majority rule chooses to always use Kriging on the testing set. These two baselines are compared with a set of decision trees built using varying sets of features. These are constructed using the rpart package in R, with parameters $cp = 0$ and $minbucket = 0.05$. This allows the constructed trees to keep growing as long as prediction error is reduced, while restricting the leaves of the tree to contain at least 5% of the instances in the training set. Two sets of prediction trees are compared, the first built using only the feature *CC*, and the second using all the literature features, namely *CC*, *RRMSE*, problem dimension and problem budget. The resulting prediction accuracies are shown in Table 1. Interestingly, creating a decision tree with only the *CC* feature or with all literature features results in almost equally accurate trees.

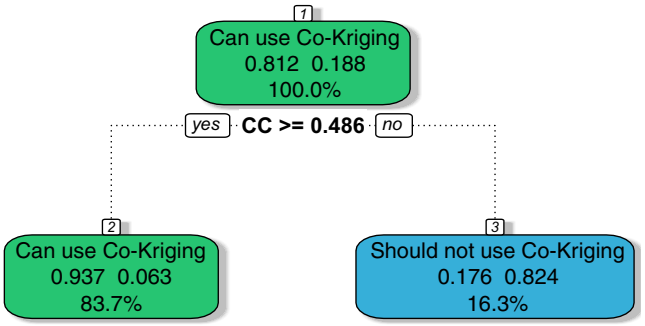
Finally, a decision tree is built using all literature features with the whole literature test suite as the training set, as shown in Figure 3, to further analyse the impact of different features on accuracy. This final tree states that Co-Kriging should be used only when $CC \geq 0.486$ (93.7% true positive rate, 82.4% true negative rate, 91.9% accuracy), which would be an improvement over the choice to always use Co-Kriging (81.5% accuracy). Despite being allowed to grow unrestricted (by setting $cp = 0$), the tree is quite small. It is, however,

**Table 1.** Cross-Validated Accuracies of Different Techniques Predicting When to Use Co-Kriging on the Literature Test Suite

| Prediction technique | Literature test suite |
|---|---|
| Always use Co-Kriging | 81.5% |
| Majority rule | 81.5% |
| Decision tree, using CC only | 90.9% |
| Decision tree, using literature features | 91.0% |

*Note.* The techniques include always using Co-Kriging, a majority rule, and the accuracies of constructing a decision tree using only the *CC* feature, or using all of the literature features, that is *RRMSE*, *CC*, problem dimension, and problem budget.

**Figure 3.** (Color online) Decision Tree Constructed Using the Literature Features and the Performance of Co-Kriging and Kriging on the Literature Test Suite




*Notes.* The label "Can use Co-Kriging" means the tree predicts Co-Kriging performs no worse than Kriging. The label "Should not use Co-Kriging" means the tree predicts Co-Kriging performs worse than Kriging. The percentages indicate the instances present in the node, and the proportions indicate the proportion of the instances in the node for which Co-Kriging can be used (left) and for which it should not be used (right).

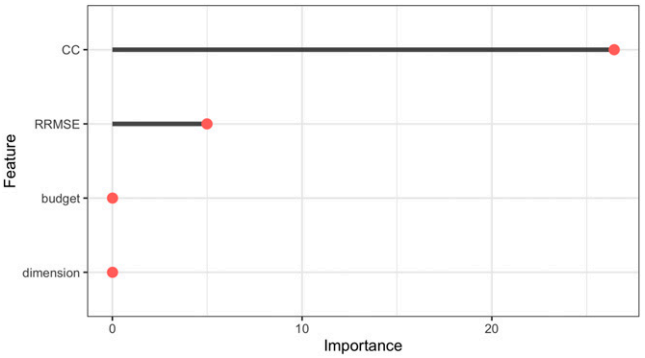
possible that features other than *CC* could have been used to produce a different tree with similar accuracy. To get a sense of the relevance of each of the features, the variable importance value of the decision tree is inspected. This value indicates the importance of each of the features when constructing the tree, as it indicates the reduction in prediction error obtained when splitting nodes in the tree using a particular feature. The variable importance is measured in each of the 100 runs when constructing a decision tree using all literature features, and the average is shown in Figure 4.

The results presented so far might lead the reader to the following natural conclusions:

- The constructed literature test suite appears to be varied enough, as seen in Figure 1, and therefore they present a good test of algorithm performance.
- Co-Kriging is overall a safe technique when applied to Bf-EBB problems, as it performed no worse than Kriging in 81.5% of the instances analysed, as shown in Table 1.
- The feature *CC* is sufficient to accurately predict when Co-Kriging can be used, and the use of other features only results in a marginal improvement on this accuracy, as shown in Table 1. Figure 4 shows that this feature has a much higher importance than the other features when constructing a decision tree.
- For problems with a total budget of $5d$ and a cost ratio $C_r = 0.1$, it is beneficial to use $f_l$ (through Co-Kriging) for $CC \geq 0.486$, as shown in Figure 3.

These conclusions are in agreement with consensus in the literature, except perhaps the last one, as Toal recommends the use of Co-Kriging only for $CC > 0.9$. His work, however, studies the impact of different budgets and cost ratio values, rather than fixing them to a given value. It is likely that for different budgets and cost ratios, the critical value at which the function $f_l$ is deemed to be useful may vary. In the next section, these conclusions are further analysed, through the introduction of new features and instances, to see if they are upheld when an intentionally more diverse suite of test instances is explored.

**Figure 4.** (Color online) Average Variable (Feature) Importance of 100 Runs When Constructing a Decision Tree Using All the Literature Features on a Training Data Set Chosen from the Literature Test Suite




## 3. Creation of New Test Suites

The quality of a test suite is based on its ability to provide a clear analysis of algorithm performance. This is achieved by showcasing the strengths and weaknesses of the algorithms being tested: where existing algorithms perform well, and where they do not, and where new algorithms might be needed. This analysis can only be conducted if the test suite is heterogeneous, that is, if the instances it contains are as diverse and varied as possible. As will be seen in this section, new features are needed to further analyse differences between instances, and to guide the creation of new instances and appropriate test suites.

### 3.1. New Features

A potential shortcoming of the instance features currently considered in the literature is their global nature. By only measuring the global macro-level relationship between $f_l$ and $f_h$, it is possible for the resulting analysis of the instance to be insufficiently nuanced. Take, for example, two different instances (i.e., two different $(f_l, f_h)$ pairs) with the same *CC* value. It is possible that in one case the quality of $f_l$ (i.e., correlation with $f_h$) is roughly constant everywhere in the sample space, whereas in the second case, $f_l$ is highly accurate half of the time, and wildly inaccurate the other half. It is quite likely that these two instances will lead to different performance of multifidelity algorithms despite having a similar *CC* value. Additional features are therefore needed to assess the local behaviour of $f_l$.

The concept of weighted sample correlation is adapted in this work to create new features. Given two sets $\mathbf{Y}_l$ and $\mathbf{Y}_h$ of samples of $f_l$ and $f_h$, respectively, at locations $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, as well as a set of weights $\mathbf{w} = \{w_1, \ldots, w_n\}$, the weighted correlation coefficient (WCC) is given by

$$WCC(\mathbf{w}) = \left[\frac{1}{\sum_{i=1}^{n} w_i}\left(\frac{\sum_{i=1}^{n} w_i (Y_i^l - \bar{Y}_l)(Y_i^h - \bar{Y}_h)}{s_{Y_l} s_{Y_h}}\right)\right]^2,$$

$$\text{where} \quad \bar{Y}_h = \frac{\sum_{i=1}^{n} w_i Y_i^h}{\sum_{i=1}^{n} w_i} \qquad s_{Y_h} = \left[\frac{\sum_{i=1}^{n} w_i (Y_i^h - \bar{Y}_h)^2}{\sum_{i=1}^{n} w_i}\right]^{1/2},$$

$$\bar{Y}_l = \frac{\sum_{i=1}^{n} w_i Y_i^l}{\sum_{i=1}^{n} w_i} \qquad s_{Y_l} = \left[\frac{\sum_{i=1}^{n} w_i (Y_i^l - \bar{Y}_l)^2}{\sum_{i=1}^{n} w_i}\right]^{1/2}.$$

Note that if $w_i = 1$ for all $i$, $WCC = CC$. Given the samples $\mathbf{Y}_l$ and $\mathbf{Y}_h$, we define the local correlation coefficient at a point $\mathbf{x}$ with radius $r$ as

$$LCC^r(\mathbf{x}) = WCC(\mathbf{w}),$$

$$\text{where} \quad w_i = \min\left\{0, 1 - \frac{\|\mathbf{x} - \mathbf{x}_i\|}{r\|\mathbf{x}^u - \mathbf{x}^l\|}\right\}.$$

This measure calculates the correlation between $f_l$ and $f_h$ inside the $d$-sphere centred at a point $\mathbf{x}$ with radius $r\|\mathbf{x}^u - \mathbf{x}^l\|$, where the measure gives higher importance to points that are closer to the centre. Given the set of local correlations $\mathcal{L}^r = \{LCC^r(\mathbf{x}_1), \ldots, LCC^r(\mathbf{x}_n)\}$, a new family of features is created which calculates the proportion of local correlations which are above a certain threshold $p$

$$LCC_p^r = \frac{|\mathcal{L}_p^r|}{|\mathcal{L}^r|}$$

Here, $\mathcal{L}_p^r = \{LCC^r(\mathbf{x}) \in \mathcal{L}^r \mid LCC^r(\mathbf{x}) \geq p\}$ is the set of local correlations with a value larger than $p$. The family of features $LCC_p^r$ provides an approximation of the probability that a randomly chosen point in $\Omega$ has a local correlation larger than $p$. The set $\mathcal{L}^r$ can be used to calculate two additional features that represent the variability in local correlation. These are the sample standard deviation $LCC_{sd}^r$ and coefficient of variance $LCC_{coeff}^r$ of $\mathcal{L}^r$ and are given by the formula

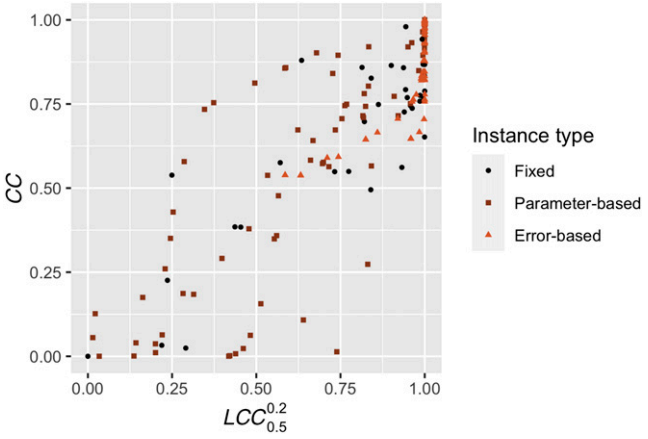
$$LCC_{sd}^r = \sqrt{\frac{\sum_{i=1}^{n} [LCC^r(\mathbf{x}_i) - \overline{LCC}^r]^2}{n-1}}$$

$$LCC_{coeff}^r = \frac{LCC_{sd}^r}{\overline{LCC}^r},$$

$$\text{where } \overline{LCC}^r = \frac{1}{n}\sum_{i=1}^{n} LCC^r(\mathbf{x}_i).$$

In this study, these features are all calculated with $r = 0.2$. This value is chosen to represent a neighbourhood that is small enough to measure local behaviour, but large enough to encompass changes in the relationship between $f_l$ and $f_h$. The values $p = \{0.1, 0.2, \ldots, 0.8, 0.9, 0.95, 0.975\}$ are used to assess which threshold produces the most relevant feature, where $p$ can be thought of as indicating the threshold separating good (i.e., high) and bad (i.e., low) correlation values.

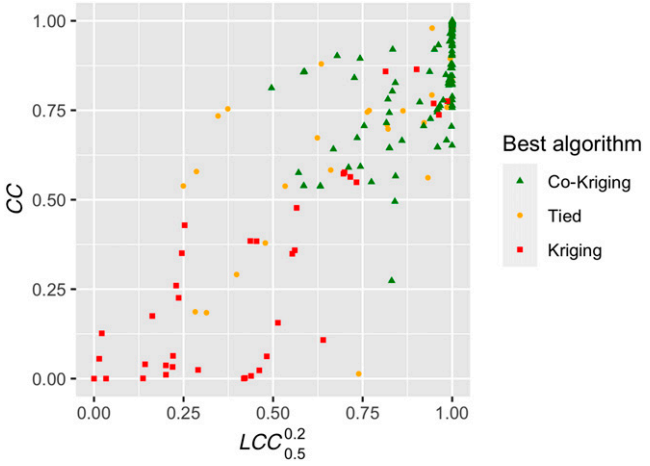
**Figure 5.** (Color online) Literature Instances Plotted Using a Feature from the Literature ($CC$, $y$-axis) and a Proposed New Feature ($LCC_{0.5}^{0.2}$, $x$-axis)




*Notes.* Each plot point represents an instance, that is, a pair of functions ($f_l$, $f_h$). Note that the instances have similar values for both features (the relationship is strongly linear), and the top-left and bottom-right quadrants are relatively empty.

The previous section demonstrated that for a budget of $5d$ and cost ratio $C_r = 0.1$, the correlation threshold for which Co-Kriging can be used appears to be $CC > 0.486$. Therefore, the feature $LCC_{0.5}^{0.2}$ is of particular interest as it uses a threshold of 0.5 when assessing whether a local correlation is high enough to be considered good. This feature is used to replot the literature test suite instances, resulting in Figures 5 and 6. As can be seen in both figures, despite containing a large set of instances, the literature test suite only occupies limited regions of the sample space, with most instances having very similar $CC$ and $LCC_{0.5}^{0.2}$ values. Figure 5 shows that error-based instances suffer the most from this, as most of them have very high $CC$ and $LCC_{0.5}^{0.2}$ values. Parameter-based instances provide some variation within the test suite, although most instances still lie in the top-right and bottom-left regions. Figure 6 shows why this could be a problem. If the space is divided into four quadrants, the top-right is mainly filled with instances where Co-Kriging performs well, whereas the bottom-left region is filled with instances where it does not. As the other two regions are relatively empty, however, it is hard to assess whether Co-Kriging performs well for instances with a high $CC$ value or for instances with a high $LCC_{0.5}^{0.2}$, and how it will perform with instances that lie in the empty regions. To analyse the causes for the observed results, new instances that further fill the space are required.

**Figure 6.** (Color online) Comparative Performance of Kriging and Co-Kriging on the Literature Test Suite, Plotted Using a Feature from the Literature (*CC*, *y*-axis) and a Proposed New Feature ($LCC_{0.5}^{0.2}$, *x*-axis)




*Notes.* Each plot point represents an instance, that is, a pair of functions ($f_l$, $f_h$), and the colour/shape represents which technique performed best, either Kriging (red square), Co-Kriging (green triangle), or statistically equal performance (yellow circle). Note that the top-right quadrant is filled with instances where Co-Kriging performs well, the bottom-left is filled with instances where it does not, and the top-left and bottom-right regions are almost empty.

## 3.2. Generating New Instances

A new instance generating procedure is presented here. The aim is two-fold. Firstly, the procedure must generate artificial instances that can represent industry problems. This is needed as algorithms being tested will eventually be applied to real-world problems, and therefore conclusions being drawn from the use of a test suite must be applicable to industry. Secondly, to fill the space shown in Figure 5, instances are needed for which the quality of $f_l$ changes throughout the domain $\Omega$. This differs from existing literature instances, which have a $f_l$ function with a fixed quality throughout the whole domain.

The proposed procedure for constructing the low-fidelity function $f_l$ consists of two steps. A basic disturbance is first defined, which can be added to the high-fidelity function, similar to the error term used by Wang et al. (2017). A set of modifications are then defined, which can be applied to a basic disturbance so that its impact on the function varies throughout $\Omega$. A single basic disturbance is used in this work, namely

$$dist(\mathbf{x},\alpha,\nu) = \alpha(f_{max}-f_{min})\cos\left(2\pi\nu\frac{\|\mathbf{x}-\mathbf{x}_{min}\|}{\|\mathbf{x}_{max}-\mathbf{x}_{min}\|}\right)$$
$$\sin\left(2\pi\nu\left[\frac{\|\mathbf{x}-\mathbf{x}_{min}\|}{\|\mathbf{x}_{max}-\mathbf{x}_{min}\|}\right]^2\right).$$

The disturbance is defined relative to a given high-fidelity function, with $f_h : \Omega \to [f_{min}, f_{max}] \subset \mathbb{R}$ being bounded by the hypercube $\Omega = [x^1_{min}, x^1_{max}] \times [x^2_{min}, x^2_{max}] \times \cdots \times [x^d_{min}, x^d_{max}]$. The constant $\alpha$ modulates the amplitude of the disturbance, and $\nu$ the frequency of the trigonometric functions. Figure 7 shows the creation of a function $f_l$ via the addition of the disturbance to $f_h$. It is important to note that a single basic disturbance is used in this study to showcase how basic disturbances can be modified to generate novel instances that vary greatly from one another. The modifications presented next can be applied to any basic disturbance, including stochastic disturbances such as white noise, if the creation of a stochastic function is desired.

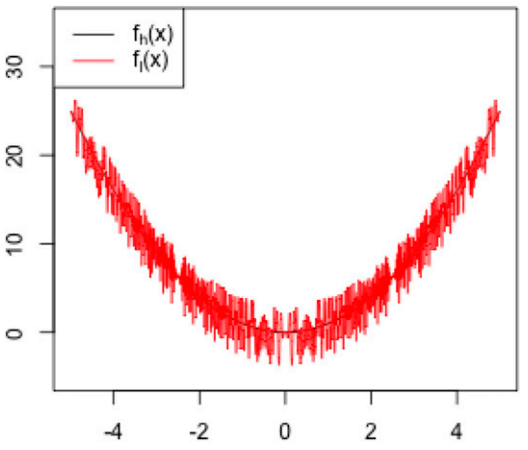
### 3.2.1. Height-Based Disturbance.

Given a basic disturbance *dist*, a modification can be applied to create height-based disturbances. Two such modifications are defined, which modulate the impact of the disturbance based on the objective function value of $f_h$. The first one adds a disturbance around a particular height with a decrease in strength as the function moves away from the given height, with no disturbance being added past a certain point. It is defined as

$$Dist_1^h(f(\mathbf{x}),h,r) = M \cdot dist,$$
$$\text{with } M = \max\left\{0, 1 - \frac{\left|\frac{f(\mathbf{x})-f_{min}}{f_{max}-f_{min}} - h\right|}{r}\right\}.$$

Here, $h \in [0,1]$ specifies the proportional height about which the disturbance is added, and $r \in [0,1]$ specifies the proportional radius. This modification can be used to construct a function $f_l$ representing the readout of a measuring device, for which the accuracy decreases for certain outputs. The second modification removes the disturbance around a particular height instead,

**Figure 7.** (Color online) Basic Disturbance When Added to the Function $f_h(x) = x^2$ with $\alpha = 0.3$ and $\nu = 10$ to Create a $f_l(x) = f_h(x) + dist(x,\alpha,\nu)$ Function



and adds the disturbance past a certain radius. It is defined by

$$Dist_2^h(f(\mathbf{x}),h,r) = M \cdot dist,$$
$$\text{where } M = \max\left\{0, 1 - \frac{1 - \left|\frac{f(\mathbf{x}) - f_{min}}{f_{max} - f_{min}} - h\right|}{1-r}\right\}.$$

Here $r$ and $h$ are similarly defined. This modification can be used to create a function $f_l$ representing a model constructed for a particular output range in which the output is accurate, beyond which it starts to lose accuracy. Both of these modifications can be added to a $f_h$ function in order to create a $f_l$ function as shown in Figure 8.

**3.2.2. Location-Based Disturbance.** Similarly to height-based disturbances, location-based disturbances are a way to modulate the disturbance based on the sample location of $f_h$, with locations defining sources of accuracy or sources of inaccuracy. This can arise in practice when a measuring device has been created for a certain variable range, a model has been trained on data from a particular region, or physical assumptions from a model no longer apply in a particular region. Given a set of locations $L = \{\mathbf{x}_1^s, \ldots, \mathbf{x}_l^s\}$, the first location-based disturbance is therefore defined to add no disturbance around these locations, with the disturbance arising with increasing strength past a certain radius

$$Dist_1^s(\mathbf{x},L,r) = M \cdot dist,$$
$$\text{with } M = \max\left\{0, 1 - \frac{1 - \frac{\min\{\|\mathbf{x} - \mathbf{x}_i^s\|\}}{\|\mathbf{x}_{max} - \mathbf{x}_{min}\|}}{1-r}\right\}.$$

The second location-based disturbance is defined to have a disturbance near the locations, with the disturbance decreasing after a certain radius

$$Dist_2^s(\mathbf{x},L,r) = M \cdot dist,$$
$$\text{with } M = \max\left\{0, 1 - \frac{\min\{\|\mathbf{x} - \mathbf{x}_i^s\|\}}{r\|\mathbf{x}_{max} - \mathbf{x}_{min}\|}\right\}.$$

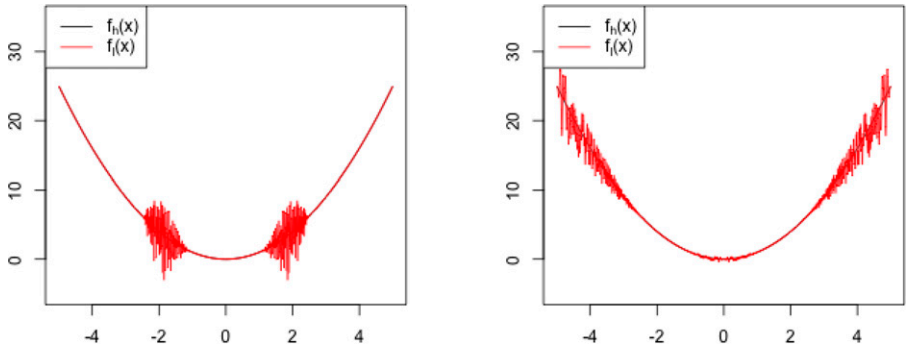
Here, $r \in [0,1]$ again denotes the relative radius of the disturbance. Both of these modifications can also be added to a $f_h$ function to create a $f_l$ function, as shown in Figure 9.

### 3.3. Augmented Test Suite Construction

The newly defined instance generating procedure is used to create a large set of instances from which an augmented test suite can be constructed. The COCO test suite (Hansen et al. 2021), which is composed of 24 generating functions that can be instantiated through translation and rotation, is used for this purpose. Therefore, 75 $f_h$ functions are created by taking the first instantiation from functions $\{1,\ldots,20\}$ with dimensions 1 (except for functions 8, 9, 17, 18, and 19, which are defined for $d \geq 2$), 2, 5, and 10. Each $f_h$ function is then used to construct a set of $(f_l, f_h)$ pairs via the addition of disturbances $Dist_1^h, Dist_2^h, Dist_1^s$, or $Dist_2^s$, using values $h \in \{0, 0.25, 0.5, 0.75, 1.0\}$, $r \in \{0.025,\ 0.05, 0.1, 0.15, 0.2, 0.25\}$, a set $L$ with one, three, six, or nine randomly chosen locations, and a basic disturbance with $\alpha \in \{0.1, 0.5, 1.0, 1.5, 2.0\}$ and $\nu \in \{10, 100\}$. This leads to the construction of 81,000 instances, of which a subset is chosen that possess a variety of $CC$ and $LCC_{0.5}^{0.2}$ values. This is achieved by choosing, when possible, a single instance $i$ of dimension $d$ satisfying $0.05n - 0.05 \leq CC(i) \leq 0.05n$ and $0.05m - 0.05 \leq LCC_{0.5}^{0.2}(i) \leq 0.05m$ for $n, m \in \{1, 2, \ldots, 20\}$ and $d \in \{1, 2, 5, 10\}$. The chosen subset of instances is added to the literature test suite to construct the augmented test suite.

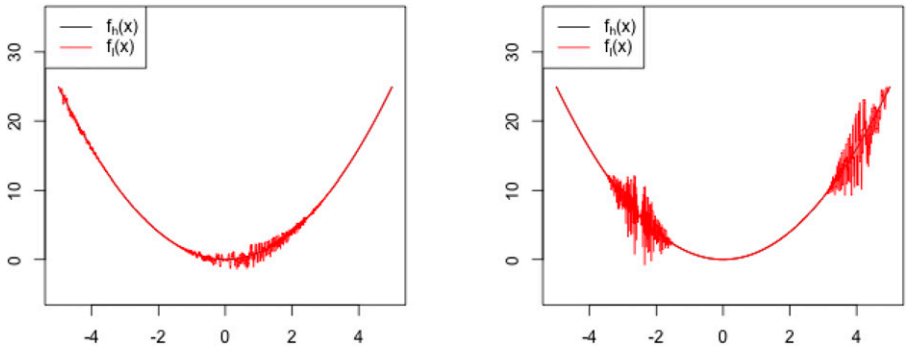
The set of 81,000 instances as well as the augmented test suite are shown in Figure 10. As can be seen on the left plot, despite using a single basic disturbance, the use of the different modifications allows for the

**Figure 8.** (Color online) Height-Based Disturbances Added to the Function $f_h(x) = x^2$ to Create a Function $f_l = f_h(x) + Dist_i^h(f_h(x), h, r)$




*Notes.* The modification $Dist_1^h$ (left) removes the disturbance around a particular height, whereas the modification $Dist_2^h$ (right) adds the disturbance around a particular height. The parameters used are $h = 0.15$ and $r = 0.1$, with a basic disturbance defined with $\alpha = 0.3$ and $\nu = 10$.

**Figure 9.** (Color online) Location-Based Disturbances Added to the Function $f_h(x) = x^2$ to Create a Function $f_l = f_h(x) + Dist_i^h(x, L, r)$




*Notes.* The modification $Dist_1^s$ (left) removes the disturbance around the specified locations, and the modification $Dist_2^h$ (right) adds the disturbance around the locations. The parameters used are $r = 0.1$ and $L = \{(-2.5), (4.1)\}$, with a basic disturbance with $\alpha = 0.3$ and $\nu = 10$.

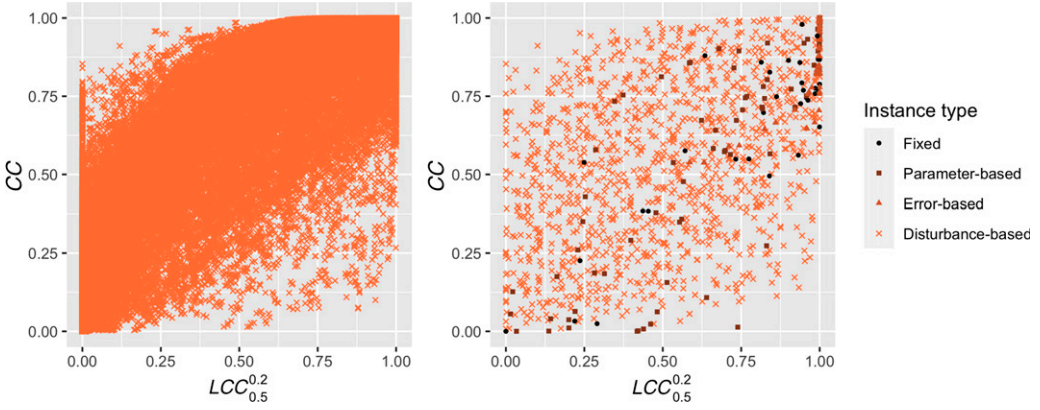
creation of instances with a large variation in *CC* and $LCC_{0.5}^{0.2}$ values, which showcases the procedure's effectiveness in more diverse instance creation. It is also worth noting that despite generating many more instances in the top-left and bottom-right quadrants than the literature test suite, there are still some regions with no instances. This makes sense as they represent extreme regions where instances have either very large *CC* and very low $LCC_{0.5}^{0.2}$ values, or very low *CC* and very large $LCC_{0.5}^{0.2}$ values. Whereas such instances could be created, they are not particularly practical, and we consider the selected augmented test suite to be sufficiently diverse for the purposes of this study.

### 3.4. Updated Algorithm Performance Analysis

The algorithm performance analysis presented in Section 2.2 is repeated with the augmented suite and additional features. Both Kriging and Co-Kriging are again used to construct surrogate models a total of 20 times each for each instance, and their comparative performance is assessed using a two-tailed Wilcoxon test. The top-performing algorithm for each of the instances is shown in Figure 11. Interestingly, it appears at a first glance that the feature *CC* no longer provides an accurate means to decide when Co-Kriging will perform no worse than Kriging as there appears to be no horizontal line that could be drawn to separate the instances shown as red squares from the instances shown as green triangles and yellow circles.

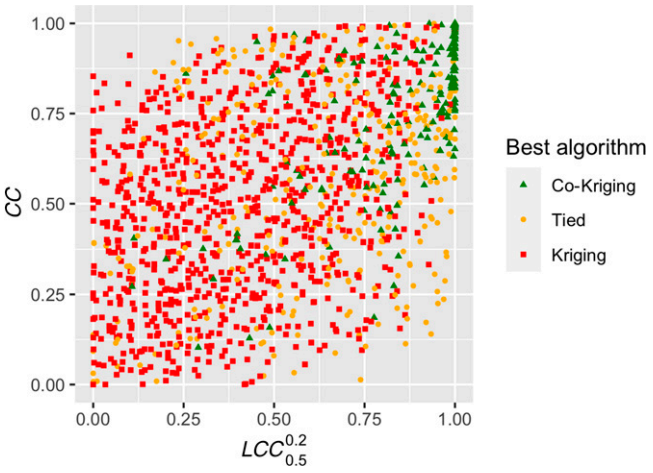
Further analysis is conducted to assess which features have the biggest impact on algorithm performance. The same prediction techniques are used to predict when Co-Kriging will perform no worse than Kriging, including the baselines "always use Co-Kriging" and a majority rule. Three sets of decision trees are built, the first using only the *CC* feature,

**Figure 10.** (Color online) Set of 81,000 Newly Generated Instances (Left) and Selected Augmented Test Suite (Right), Plotted Using a Feature from the Literature (*CC*, *y*-axis) and a Proposed New Feature ($LCC_{0.5}^{0.2}$, *x*-axis)



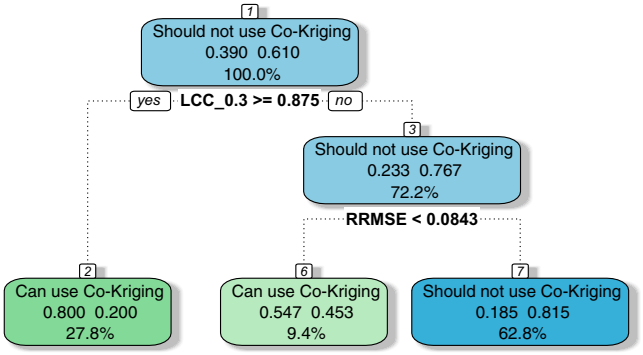

*Notes.* Each plot point represents an instance, that is, a pair of functions ($f_l$, $f_h$). The addition of the chosen subset to create the augmented test suite results in a much more varied test suite.

**Figure 11.** (Color online) Comparative Performance of Kriging and Co-Kriging on the Augmented Test Suite




*Notes.* Each plot point represents an instance, that is, a pair of functions ($f_l$, $f_h$) and its position is based on the features correlation coefficient (*CC*, *y*-axis) and local correlation coefficient with threshold 0.5 ($LCC^{0.2}_{0.5}$, *x*-axis). The colour/shape represents which technique performed best, either Kriging (red square), Co-Kriging (green triangle), or statistically equal performance (yellow circle). Note that $LCC^{0.2}_{0.5}$ appears to have a bigger impact on performance than *CC*.

**Figure 12.** (Color online) Decision Tree Constructed Using New and Literature Features and the Performance of Co-Kriging and Kriging on the Augmented Test Suite




*Notes.* The label "Can use Co-Kriging" means the tree predicts Co-Kriging performs no worse than Kriging. The label "Should not use Co-Kriging" means the tree predicts Co-Kriging performs worse than Kriging. The percentages indicate the instances present in the node, and the proportions indicate the proportion of the instances in the node for which Co-Kriging can be used (left) and for which it should not be used (right).

the second using all literature features (*CC*, *RRMSE*, budget, problem dimension), and the third using literature and new features (*CC*, *RRMSE*, budget, problem dimension, $LCC^{0.2}_{sd}, LCC^{0.2}_{coeff}, LCC^{0.2}_{0.1}, \ldots, LCC^{0.2}_{0.9}, LCC^{0.2}_{0.95}, LCC^{0.2}_{0.975}$). The resulting cross-validated accuracies, averaged over 100 runs, are shown in Table 2 for both the literature and the augmented test suites. The second column presents a strong contrast to the conclusions drawn in Section 2.2. Unlike with the literature test suite, with the augmented test suite it is best to choose always to use Kriging (the choice of the majority rule), as choosing to use Co-Kriging chooses the right technique only 38.8% of the time. It also appears that the feature *CC* no longer provides a very accurate indication of which technique will perform best. The decision trees constructed using only this feature performed remarkably worse than the ones constructed using all literature features, and a tree constructed using all features provides a further improvement in accuracy.

The importance of variables in the decision trees is again analysed to examine the impact of each feature on tree accuracy. Figure 12 shows a final decision tree constructed using the whole augmented test suite, and Figure 13 shows the averaged variable importance of trees constructed using all features over the 100 runs using a training set. The decision tree makes use of the newly defined feature $LCC^{0.2}_{0.3}$ and the literature feature *RRMSE*, although interestingly it makes no use of the feature *CC*. Figure 13 further shows the importance of the newly presented features, as the dominant features in terms of importance do not include any of the literature features. The results obtained through the analysis of the augmented test suite lead to the following conclusions:

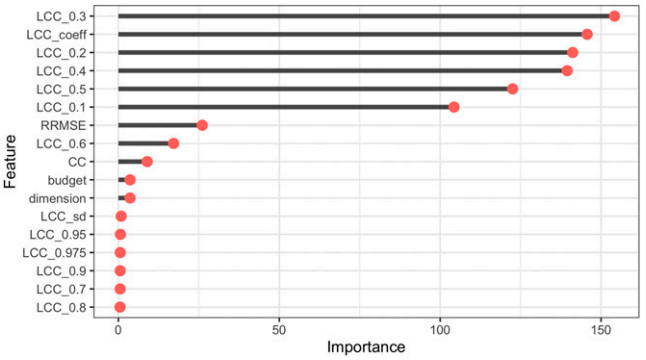
- Although Co-Kriging may be recommended under special conditions, such as a high *CC*, when we consider a more comprehensive set of instances we find

**Table 2.** Accuracies of Different Techniques When Predicting When to Use Co-Kriging on the Literature Test Suite and the Augmented Test Suite

| Classification | Literature test suite | Augmented test suite |
|---|---|---|
| Always use Co-Kriging | 81.5% | 38.8% |
| Majority rule | 81.5% | 61.2% |
| Decision tree, using CC only | 90.9% | 65.3% |
| Decision tree, using literature features | 91.0% | 71.0% |
| Decision tree, using all features | 90.9% | 77.3% |

*Note.* The techniques include always using Co-Kriging, a majority rule, and the accuracies of constructing a decision tree using only the CC feature, using all literature features (*CC*, *RRMSE*, budget, problem dimension), or using literature and newly proposed features (*CC*, *RRMSE*, budget, problem dimension, $LCC^{0.2}_{sd}, LCC^{0.2}_{coeff}, LCC^{0.2}_{0.1}, \ldots, LCC^{0.2}_{0.9}, LCC^{0.2}_{0.95}, LCC^{0.2}_{0.975}$).

**Figure 13.** (Color online) Average Variable (Feature) Importance of 100 Runs When Constructing a Decision Tree Using New and Literature Features on a Training Data Set Chosen from the Augmented Test Suite




*Note.* Notice that the importance of the literature features is quite small relative to the newly defined features.

that Co-Kriging is not likely to perform better than Kriging for many types of instances.

- Choosing to always use Co-Kriging leads to only 38.8% accuracy in predicting which algorithm to use, with the majority rule choosing to always use Kriging instead with an accuracy of 61.2% (Table 2). It is therefore more likely that Co-Kriging will perform worse than Kriging except under some very specific conditions.
- The feature *CC* only gives a rough indication of when Co-Kriging can be used, as the accuracy of a decision tree constructed with this feature presents a small improvement over the majority rule (Table 2) and the feature importance of *CC* is very small relative to other features (Figure 13).
- On the other hand, measures of local correlation (i.e., how $f_l$ behaves locally, and how this behaviour changes throughout $\Omega$) give a good indication of when Co-Kriging can be used (Figure 13).
- For problems with a total budget of $5d$ and a cost ratio $C_r = 0.1$, it is beneficial to use $f_l$ (through the use of Co-Kriging) only if $f_l$ is often somewhat locally correlated to $f_h$ ($LCC_{0.3}^{0.2} \geq 0.875$), or if the error between $f_h$ and $f_l$ is very small ($RRMSE < 0.0843$).

These conclusions represent a significant divergence from the assumptions presented in the literature, and the conclusions drawn in Section 2 based on a more limited set of test instances studied in the existing literature. These findings show that the choice of test instances matters, for reasons we discuss further in the next section.

## 4. Discussion

Sections 2 and 3 presented a comparative analysis of the performance of Kriging and Co-Kriging models through the use of two different instance test suites. Despite the analysis procedure being identical in both cases, the resulting conclusions strongly contradict one another. It is clear from these differences that the choice of test instances can have a significant impact on the resulting algorithm analysis, and therefore on the directions taken in the field of multifidelity surrogate modelling and optimisation.

### 4.1. Impact of New Features

The work of Toal (2015) highlighted the need for some measure of quality of $f_l$ when assessing whether a bifidelity technique such as Co-Kriging can be used when constructing a surrogate model. His work showcased the usefulness of the feature *CC*, which has led to subsequent studies (Wang et al. 2017, Song et al. 2019, Müller 2020, Shi et al. 2020, Lv et al. 2021, van Rijn et al. 2022) to take this feature into account when performing algorithm analysis and instance creation. Despite its usefulness, the work presented here shows the potential shortfalls of this feature, in particular its global nature. Defining new features that measure the change in local correlation between $f_l$ and $f_h$ rather than overall correlation has revealed the potential bias present in existing literature instances. Section 2 presents what seems to be a varied literature test suite, however Section 3 demonstrates that most instances in this suite have in fact very similar *CC* and $LCC_{0.5}^{0.2}$ values, indicating that the overall quality of $f_l$ and its local quality are similar throughout $\Omega$ in most literature instances. Many real-world problems, however, show variation in local $f_l$ quality, due to model assumptions breaking down or device measurements having non-uniform errors, among others.

The dangers incurred by the bias of the literature test suite is demonstrated in the accuracy of the different classification techniques. Table 2 shows that for this test suite, Co-Kriging performs no worse than Kriging most of the time, and therefore in general if a function $f_l$ is available in an EBB problem, it should be used when constructing a surrogate model. Figure 6, however, indicates that this is likely the result of most instances having a large $LCC_{0.5}^{0.2}$ value, despite the whole test suite being varied in terms of the *CC* feature. When constructing the augmented test suite with instances with varying *CC* and $LCC_{0.5}^{0.2}$ values, Co-Kriging performs no worse than Kriging only 38.3% of the time. This does not mean Co-Kriging is not a good technique; rather, as Toal points out, one should be cautious and selective about when to use it.

Interestingly, the use of the feature *RRMSE* can increase the accuracy of decision trees for the augmented test suite, as the trees built using literature features were on average 5.6% more accurate than trees built using only the *CC* feature. This is surprising as, despite this feature being discussed by Toal, little importance has been given to it in the literature regarding its impact on algorithm performance. It is

likely, however, that this feature is correlated with the newly presented features due to the instance creation procedure used in this work, rather than being a useful feature in and of itself more generally. Indeed, the variable importance of *RRMSE* when constructing decision trees using all features is relatively small compared with the features with the highest importance. It appears that out of all features discussed in this work, those pertaining to local correlation (i.e., $LCC^{0.2}_{coeff}$ and $LCC^{0.2}_{p}$) have the largest impact on model accuracy.

The usefulness of the newly presented features is also clearly shown in Table 2. Despite the literature consensus being that the feature *CC* is sufficient to estimate whether Co-Kriging can be used, classification trees constructed using all features achieved a 77.3% accuracy on the augmented test suite, which is a significant improvement over the 65.3% accuracy of trees constructed using only the *CC* feature. It is worth mentioning, however, that this accuracy is still far from the 91.0% accuracy achieved on the more homogeneous literature test suite alone. This highlights the need for further work in this area, and in particular the development of new features, to further assess the intricacies and differences between heterogeneous instances.

### 4.2. Impact of New Instances

The results shown in this work also highlight the usefulness of the proposed instance creation framework. Both Toal (2015) and Wang et al. (2017) rightly emphasised the need for new instances in the field of Mf-EBB surrogate modelling and optimisation, and their work proposed instance creation techniques that led to a varied set of instances. Their work has been influenced, however, by the fact that the definition of "varied" has been based solely on the *CC* feature. The instance creation technique proposed by Wang et al. in particular leads to instances with a variety of *CC* values and therefore a variety of overall $f_l$ quality. However, these instances suffer from having a very high $LCC^{0.2}_{0.5}$ value, indicating that nowhere in $\Omega$ does the quality of $f_l$ worsen significantly. It is, however, possible in industry problems that the low-fidelity source of information worsens only in certain regions due to physical assumptions no longer holding, a lack of training data on a simulation model, or the readout of a measuring device moving away from its operating range.

The instance generating procedure presented in this work has shown its ability to create synthetic Bf-EBB instances with these more real-world characteristics. Despite only working with a single basic disturbance, the four modifications (two height based and two location based) are able to create instances with a large variety of *CC* and $LCC^{0.2}_{0.5}$ values. These instances show the promise of defining $f_l$ functions with their quality being affected by either the output of $f_h$ or the region of $\Omega$ being sampled. These types of modifications can be further adapted to suit the needs of the instance being created. The basic disturbance could be modified to closely resemble an error term found in a particular industry, such as the creation of stochastic instances via the use of white noise as the basic disturbance. The impact of the modification could also be altered, for instance defining the modification to "flatline" $f_h$ in certain regions when creating $f_l$, in order to represent a loss in the ability of $f_l$ to represent the intricate behaviour of $f_h$. The procedure could also be modified to construct Mf-EBB instances with more than two levels of fidelity by modulating the impact of the added disturbance based on the fidelity level. This could be achieved by modulating the amplitude of the basic disturbance, or by increasing/decreasing the radius of the final disturbance.

## 5. Conclusion

This study has highlighted the need for both new features and new instances in the field of Bf-EBB through the analysis of surrogate model accuracy, both when assessing algorithm performance and when developing rules to predict when a two-source algorithm can be used in a bifidelity setting. A framework for new instance creation procedures has been proposed, which is shown to create extremely varied instances, some of which are very different from any seen in the literature and are capable of representing important real-world characteristics. These procedures showcase the potential of quantifying instances using local rather than global features. New features are also presented that can better differentiate between instances, furthering the understanding of what impacts the performance of a particular algorithm in this field.

The work presented here constitutes a proof of concept rather than a complete work, however, and has some limitations. Despite the computational setup being very similar to that of Toal (2015), this study is more limited in the sense that both the sample budget and the cost ratio have been fixed to $5d$ and $C_r = 0.1$, respectively. For this reason, it is important to place the obtained results in perspective. Namely, the decision tree presented at the end of Section 3 stating that Co-Kriging can be used for instances with $LCC^{0.2}_{0.3} \geq 0.875$ or $RRMSE < 0.0843$ should be taken as an indication of the importance of these features, rather than as a set of rules to be followed in further work. Furthermore, whereas it is quite likely that the newly presented features will have a significant impact on algorithm performance for other budgets and cost ratios, the generalisation of these findings to other experimental settings remains to be conducted in future work.

This study has also been restricted to the analysis of surrogate model accuracy given a fixed sample. It is

not clear, however, whether a technique that will create the most accurate model given an initial sample will do so also if given a further sampling budget, and whether it will also perform best in function optimisation. Within this context, the usefulness of the new instances and features presented here remains to be studied. Furthermore, this work has looked at single-source versus two-source algorithm performance in the form of Kriging versus Co-Kriging. These techniques, however, are by no means the only ones available. In answering the question of when a two-source algorithm can safely be used for a bifidelity problem, many such algorithms should be considered. Further work in this area should pay special attention to newly presented techniques that focus on bifidelity instances for which $f_l$ and $f_h$ are not necessarily correlated. The work of both van Rijn et al. (2022) and Müller (2020) are of particular interest in this regard, as they both present ways of dealing with instances with different *CC* values, the former in surrogate model accuracy and the latter in function optimisation. Further analysis of these techniques in the context of the newly defined instances and features would be beneficial.

Finally, the increase in scope of future analysis in terms of varying instance budgets and cost ratios, and simultaneous assessment of multiple algorithms, will require the use of more sophisticated classification and prediction techniques. In this work, a relatively simple machine learning method in the form of decision trees has been used. This choice can be justified due to the relative simplicity of the classification "can Co-Kriging be used or not?" Furthermore, the aim here has not been to obtain a set of rules that further researchers should follow, but rather to showcase the importance of the presented instances and features. Further study will require an analysis of how well each of the features can be approximated given a limited budget, and the adoption of more sophisticated classification and prediction techniques. Instance space analysis (Muñoz and Smith-Miles 2017) in particular is likely to be beneficial for future analysis, as it has been shown to support a more insightful analysis of which features have an impact on algorithm performance, the strengths and weaknesses of different algorithms, and for which types of instances certain algorithms can be expected to outperform others.

## References

Aleman DM, Romeijn HE, Dempsey JF (2009) A response surface approach to beam orientation optimization in intensity-modulated radiation therapy treatment planning. *INFORMS J. Comput.* 21(1): 62–76.

Andrés-Thió N (2022) Bifidelity surrogate modelling. http://dx.doi.org/10.5281/zenodo.6208147, available for download at https://github.com/nandresthio/Bifidelity-Surrogate-Modelling.

Andrés-Thió N, Muñoz M, Smith-Miles K (2021) Bi-fidelity surrogate modelling: Showcasing the need for new test instances. http://dx.doi.org/10.5281/zenodo.6578060, available for download at https://github.com/INFORMSJoC/2021.0299.

Andrés-Thió N, Muñoz MA, Smith-Miles K (2022) Data folder. http://dx.doi.org/10.6084/m9.figshare.19196594.v1, available for download at https://figshare.com/articles/dataset/Data_folder/19196594/1.

Dong H, Song B, Wang P, Huang S (2015) Multi-fidelity information fusion based on prediction of Kriging. *Structural Multidisciplinary Optim.* 51(6):1267–1280.

Duchon J (1977) Splines minimizing rotation-invariant semi-norms in Sobolev spaces. *Constructive Theory of Functions of Several Variables* (Springer, Berlin), 8–100.

Durantin C, Rouxel J, Désidéri JA, Glière A (2017) Multifidelity surrogate modeling based on radial basis functions. *Structural Multidisciplinary Optim.* 56(5):1061–1075.

Fernández-Godino MG, Park C, Kim NH, Haftka RT (2019) Issues in deciding whether to use multifidelity surrogates. *AIAA J.* 57(5):2039–2054.

Forrester AI (2010) Black-box calibration for complex-system simulation. *Philos. Trans. Roy. Soc. A Math. Physical Engrg. Sci.* 368(1924): 3567–3579.

Forrester AI, Sóbester A, Keane AJ (2007) Multi-fidelity optimization via surrogate modelling. *Proc. Roy. Soc. A Math. Physical Engrg. Sci.* 463(2088):3251–3269.

Gutmann HM (2001) A radial basis function method for global optimization. *J. Global Optim.* 19(3):201–227.

Hansen N, Auger A, Ros R, Mersmann O, Tušar T, Brockhoff D (2021) COCO: A platform for comparing continuous optimizers in a black-box setting. *Optim. Methods Software* 36(1):114–144.

Jones DR (2001) A taxonomy of global optimization methods based on response surfaces. *J. Global Optim.* 21(4):345–383.

Kennedy MC, O'Hagan A (2000) Predicting the output from a complex computer code when fast approximations are available. *Biometrika* 87(1):1–13.

Krige DG (1951) A statistical approach to some basic mine valuation problems on the witwatersrand. *J. Southern Africa Inst. Mining Metallurgy* 52(6):119–139.

Liu B, Koziel S, Zhang Q (2016) A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems. *J. Comput. Sci.* 12:28–37.

Liu H, Ong YS, Cai J (2018a) A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural Multidisciplinary Optim.* 57(1):393–416.

Liu Y, Chen S, Wang F, Xiong F (2018b) Sequential optimization using multi-level coKriging and extended expected improvement criterion. *Structural Multidisciplinary Optim.* 58(3):1155–1173.

Liu H, Ong YS, Cai J, Wang Y (2018c) Cope with diverse data structures in multi-fidelity modeling: A Gaussian process method. *Engrg. Appl. Artificial Intelligence* 67:211–225.

Lv L, Zong C, Zhang C, Song X, Sun W (2021) Multi-fidelity surrogate model based on canonical correlation analysis and least squares. *J. Mech. Design* 143(2):021705.

March A, Willcox K (2012) Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives. *AIAA J.* 50(5):1079–1089.

Matheron G (1963) Principles of geostatistics. *Econom. Geology* 58(8): 1246–1266.

Müller J (2020) An algorithmic framework for the optimization of computationally expensive bi-fidelity black-box problems. *INFOR Inform. Systems Oper. Res.* 58(2):264–289.

Müller J, Shoemaker CA (2014) Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *J. Global Optim.* 60(2):123–144.

Muñoz MA, Smith-Miles KA (2017) Performance analysis of continuous black-box optimization algorithms via footprints in instance space. *Evolutionary Comput.* 25(4):529–554.

Park C, Haftka RT, Kim NH (2017) Remarks on multi-fidelity surrogates. *Structural Multidisciplinary Optim.* 55(3):1029–1050.

Park C, Haftka RT, Kim NH (2018) Low-fidelity scale factor improves Bayesian multi-fidelity prediction by reducing bumpiness of discrepancy function. *Structural Multidisciplinary Optim.* 58(2):399–414.

Rajnarayan D, Haas A, Kroo I (2008) A multifidelity gradient-free optimization method and application to aerodynamic design. *Proc. 12th AIAA/ISSMO Multidisciplinary Anal. Optim. Conf.*, 6020.

Regis RG, Shoemaker CA (2007) A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. Comput.* 19(4):497–509.

Ruan X, Jiang P, Zhou Q, Hu J, Shu L (2020) Variable-fidelity probability of improvement method for efficient global optimization of expensive black-box problems. *Structural Multidisciplinary Optim.* 62(6):3021–3052.

Shahpar S, Brooks C, Forrester A, Keane A (2011) Multi-fidelity design optimisation of a transonic compressor rotor. *9th Eur. Conf. Turbomachinery Fluid Dynamics Thermodynamics*, Istanbul, Turkey, vol. 2.

Shi M, Lv L, Sun W, Song X (2020) A multi-fidelity surrogate model based on support vector regression. *Structural Multidisciplinary Optim.* 61:2363–2375.

Song X, Lv L, Sun W, Zhang J (2019) A radial basis function-based multi-fidelity surrogate model: Exploring correlation between high-fidelity and low-fidelity models. *Structural Multidisciplinary Optim.* 60(3):965–981.

Surjanovic S, Bingham D (2020) Virtual library of simulation experiments: Test functions and datasets. Retrieved December 14, 2020, from http://www.sfu.ca/~ssurjano.

Toal DJ (2015) Some considerations regarding the use of multi-fidelity Kriging in the construction of surrogate models. *Structural Multidisciplinary Optim.* 51(6):1223–1245.

van Rijn S, Schmitt S, van Leeuwen M, Bäck T (2022) Finding efficient trade-offs in multi-fidelity response surface modeling. *Engrg. Optim.* 1–18.

Wang H, Jin Y, Doherty J (2017) A generic test suite for evolutionary multifidelity optimization. *IEEE Trans. Evolutionary Comput.* 22(6):836–850.

Wilcoxon F (1992) Individual comparisons by ranking methods. *Breakthroughs in Statistics* (Springer, New York), 196–202.

Wild SM, Regis RG, Shoemaker CA (2008) ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM J. Sci. Comput.* 30(6):3197–3219.

Wu Y, Hu J, Zhou Q, Wang S, Jin P (2020) An active learning multi-fidelity metamodeling method based on the bootstrap estimator. *Aerospace Sci. Tech.* 106:106116.

Xiong S, Qian PZ, Wu CJ (2013) Sequential design and analysis of high-accuracy and low-accuracy computer codes. *Technometrics* 55(1):37–46.

Zhao H, Gao Z, Xu F, Xia L (2021) Adaptive multi-fidelity sparse polynomial chaos-Kriging metamodeling for global approximation of aerodynamic data. *Structural Multidisciplinary Optim.* 64:829–858.

Zhou Q, Wu Y, Guo Z, Hu J, Jin P (2020) A generalized hierarchical Co-Kriging model for multi-fidelity data fusion. *Structural Multidisciplinary Optim.* 62:1885–1904.