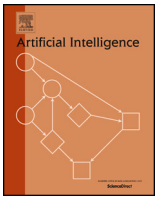
Artificial Intelligence 336 (2024) 104207

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



Characterising harmful data sources when constructing multi-fidelity surrogate models

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

[a] School of Mathematics and Statistics, The University of Melbourne, Parkville, Melbourne, 3010, VIC, Australia
[b] ARC Training Centre in Optimisation Technologies, Integrated Methodologies, and Applications (OPTIMA), Parkville, Melbourne, 3010, VIC, Australia
[c] School of Computer and Information Systems, The University of Melbourne, Parkville, Melbourne, 3010, VIC, Australia

ARTICLE INFO

Keywords:
Expensive Black-Box
Surrogate modelling
Bi-fidelity
Kriging
Co-Kriging
Instance Space Analysis

ABSTRACT

Surrogate modelling techniques have seen growing attention in recent years when applied to both modelling and optimisation of industrial design problems. These techniques are highly relevant when assessing the performance of a particular design carries a high cost, as the overall cost can be mitigated via the construction of a model to be queried in lieu of the available high-cost source. The construction of these models can sometimes employ other sources of information which are both cheaper and less accurate. The existence of these sources however poses the question of which sources should be used when constructing a model. Recent studies have attempted to characterise harmful data sources to guide practitioners in choosing when to ignore a certain source. These studies have done so in a synthetic setting, characterising sources using a large amount of data that is not available in practice. Some of these studies have also been shown to potentially suffer from bias in the benchmarks used in the analysis. In this study, we approach the characterisation of harmful low-fidelity sources as an algorithm selection problem. We employ recently developed benchmark filtering techniques to conduct a bias-free assessment, providing objectively varied benchmark suites of different sizes for future research. Analysing one of these benchmark suites with the technique known as Instance Space Analysis, we provide an intuitive visualisation of when a low-fidelity source should be used. By performing this analysis using only the limited data available to train a surrogate model, we are able to provide guidelines that can be directly used in an applied industrial setting.

1. Introduction

In recent years, the development and analysis of techniques for Multi-fidelity Expensive Black-Box (Mf-EBB) problems has gathered a lot of momentum. The growing attention for this type of problem can easily be justified by how often they can be found in industrial design problems such as aerodynamic modelling [35] and materials design [19]. These *black-box* design problems are characterised by the unknown relationship between design outcomes and decision variables, for which the only way to evaluate decision outcomes is via a deterministic procedure. Each evaluation is deemed to be *expensive*, meaning the amount of sampling of the black-box is severely restricted due to its high computational, monetary or time cost. Furthermore, these *multi-fidelity* problems have multiple sources of

* Corresponding author at: School of Mathematics and Statistics, The University of Melbourne, Parkville, Melbourne, 3010, VIC, Australia.
E-mail addresses: nandres@student.unimelb.edu.au (N. Andrés-Thió), munoz.m@unimelb.edu.au (M.A. Muñoz), smith-miles@unimelb.edu.au (K. Smith-Miles).

https://doi.org/10.1016/j.artint.2024.104207
Received 13 November 2023; Received in revised form 15 August 2024; Accepted 16 August 2024

Available online 23 August 2024

information available on the design outcome, with varying degrees of cost and accuracy. The simplest variant of such problems, known as Bi-fidelity Expensive Black-Box (Bf-EBB) problems, has only a single low-fidelity (as well as a high-fidelity) information source available.

The existence of more than one source allows techniques to mitigate the high cost of relying exclusively on a single, very expensive black-box by relying on cheaper (and less accurate) sources of information. This is often achieved via the construction of surrogate models which combine scarce high-fidelity data with more abundant data of lower fidelities. The aim of these models is to accurately predict the outcome value in regions that have not yet been sampled. This information is then used to guide further sampling of the objective function, where the end goal can be to accurately model the relationship of the design variables and the design outcome, or the optimisation of the design problem. A very large proportion of surrogate modelling techniques for Mf-EBB problems are either based on or variations of the seminal work by Kennedy and O'Hagan [17]. Their work presents a Bayesian methodology that fuses multiple sources of information into a single surrogate model based on Gaussian processes. Perhaps the most well-known technique derived from this work is the adaptation of Kriging [20,27,16], a surrogate technique that employs only high-fidelity data, to the multi-fidelity setting in the form of the technique known as Co-Kriging [10]. A multitude of studies exists which present variations to this technique, such as different approaches to train the surrogate model [34,41], variations of the surrogate model structure [13], different frameworks for integrating multiple non-hierarchical fidelity sources [8,6], and different procedures for guiding further sampling both within the sample space and among different sources [11,55], among others.

Due to the importance of the accuracy of the surrogate model when conducting design exploration or optimisation, recent studies have analysed the impact that the quality of low-fidelity sources has on the accuracy of a model built from them. Toal [48] was one of the first to highlight the potential risk in assuming low-fidelity sources can always be relied upon. He focused on Bf-EBB problems where, for a given sample of high and low-fidelity data, he compared the accuracy of a (single-source) Kriging model with the accuracy of a (two-source) Co-Kriging model. He concluded that the correlation between the high- and low-fidelity sources is very important, in fact stating that if the low-fidelity source is not highly correlated with the high-fidelity source, it is recommended to ignore it entirely and train a (single-source) Kriging model instead. Further work in this area [7,33,23,40] has reinforced this guideline for low-fidelity data usage. These results have led to the inclusion in new research of an analysis of the test instances employed to assess the performance of new techniques. This analysis is often carried out by measuring the correlation between the high and low-fidelity sources [45,25,49], or by plotting the high-fidelity function values against the low-fidelity function values [56,47] in order to analyse how well the low-fidelity source represents the high-fidelity source.

A recent study by Andrés-Thió et al. [4] however has shown that using a low-fidelity source can be valuable when training a model if it is often locally accurate, even if its overall accuracy is relatively low. More importantly, their work highlighted the potential bias in literature test instances. This bias comes about from the creation procedures of synthetic high- and low-fidelity function pairs, where the quality of the low-fidelity source (either good, average, or bad) is the same throughout the design space. Indeed, it is rare to find synthetic test instances in the literature for which the low-fidelity source is only sometimes locally accurate. To remedy this, a new instance creation procedure was proposed, as well as new measures to assess differences between test instances. This work also showed the lack of and need for an unbiased test suite for Mf-EBB problems, especially when attempting to characterise what constitutes a harmful low-fidelity data source in surrogate model construction. Another key aspect of both the work of Toal [48] and Andrés-Thió et al. [4] is its synthetic setting. Indeed, whilst the models being assessed are trained with limited data, the characterisation of the high- and low-fidelity sources has so far been conducted using a large amount of data that is not available in practice. The guidelines developed in this setting have enhanced the understanding of the requirements of a useful low-fidelity source and the inner workings of the models being assessed, but these guidelines cannot be directly applied to an industrial setting. A set of guidelines based only on the limited sample available is therefore still lacking in the literature.

Existing approaches to the algorithm selection problem [39] are well suited to addressing these literature shortcomings. Accepting the fact that no method will perform best in all possible scenarios, recent research on optimisation problems has often focused on the accurate selection of the best algorithm for a specific test instance [5,21,29]. Approaches of this type seek to understand what key instance properties (here referred to as *features*) have an impact on algorithm performance. This allows techniques to steer away from the question "Which algorithm is best?", and instead answer "Which algorithm is best *for a certain type of instances*?". By using this approach to compare when a single-source surrogate model like Kriging is more accurate than a two-source surrogate model like Co-Kriging, an answer to the question "When can a low-fidelity source be relied upon?" can be found. The Instance Space Analysis (ISA) framework [42] in particular is well suited to this study, as its analysis of instances, features and algorithm performance is centred around visualising the space of all possible instances. Guidelines derived from this approach are likely to be taken up by industry practitioners, as the visualisation aspect provides an intuitive understanding that can generate human trust in the procedure. Finally, by using features that are calculated only with the available data (and not a much larger set of samples), we can ensure the resulting guidelines can be directly applied to industrial problems.

The work presented in this study extends the work of both Toal [48] and Andrés-Thió et al. [4] which compared the accuracy of Kriging and Co-Kriging models in Bf-EBB problems with given high- and low-fidelity data sets. The approach taken here presents (to our knowledge) the first study of the prediction capability of *approximated* features to assess the usefulness of low-fidelity sources in surrogate model construction. Whilst previous work [4,48] found in the literature has provided important insights into how instance features impact model performance, the reliance on feature values which are calculated exactly has meant that current guidelines remain untested in the applied setting where the same features can only be approximated due to a limited sample size. Furthermore, while ISA has a theoretical foundation based on Rice's algorithm selection framework [39], the end goal is not to assess the performances of Kriging and Co-Kriging in different regions of the instance space as is the case in classical algorithm selection studies. Rather, the aim is to exploit ISA methodology to generate intuitive insights aimed at generating trust from industrial

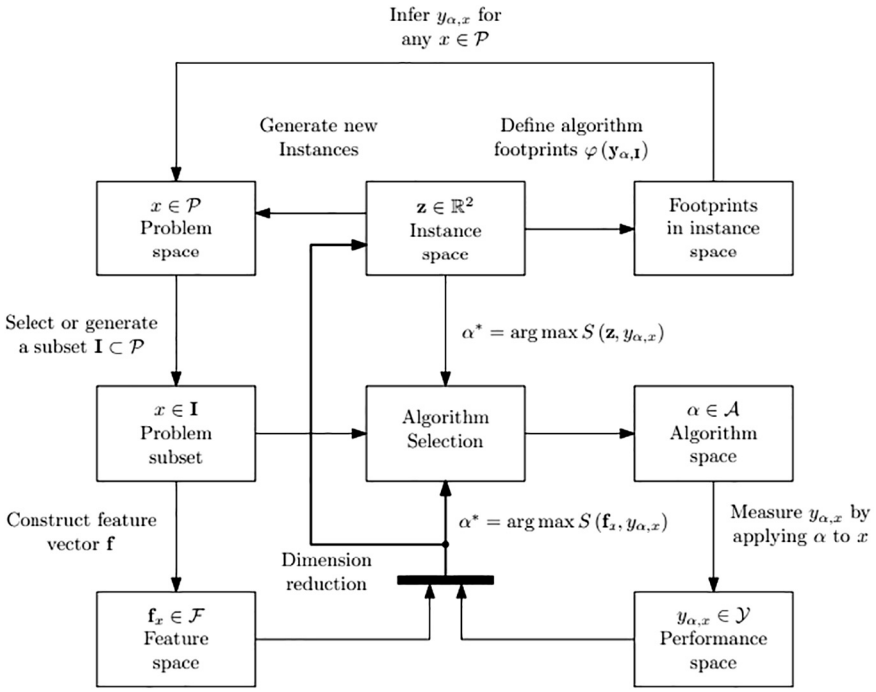


Fig. 1. ISA framework [42].

practitioners and which rely on feature values calculated with only a small sample. This constitutes the main contribution of this work: the characterisation of harmful low-fidelity data sources, at least when constructing Co-Kriging models, using only the limited data available. Furthermore, to ensure the findings are applicable to a wide set of problems, emphasis is placed in selecting a large (an order of magnitude larger than known previous studies) set of benchmarks which is both objectively diverse and unbiased. This is also highly relevant to the literature as currently most literature benchmarks are only used in the study that first defines them, and are not reused in further studies by different authors. The second main contribution of this work is therefore the construction of unbiased sets of high- and low-fidelity function pairs for future algorithm testing.

The remainder of this paper is structured as follows. Section 2 provides an introduction to ISA and a definition of the instances, features and algorithms used to perform the analysis. Section 3.1 presents the creation of a set of objectively varied function pairs which are then used to generate a set of instances for ISA. Section 3.2 uses these instances to generate the instance space and identify the regions for which Kriging and Co-Kriging should be given precedence. Section 3.3 combines the identification of these regions and the feature value trends to explain why a low-fidelity source is harmful or beneficial. Section 3.4 provides a simplified single-feature analysis in order to provide some easy-to-use guidelines for practitioners in the field. Finally, Section 4 concludes the paper with some closing remarks.

2. Preliminaries

2.1. Instance Space Analysis

Instance Space Analysis (ISA) [42] is based both on the work of Rice [39], which proposed using instance features to predict algorithmic performance, and the No-Free Lunch theorems proposed by Wolpert and Macready [51], the first of which states that "any two optimization algorithms are equivalent when their performance is averaged across all possible problems". As such, rather than analysing algorithm performance via taking the average performance across a set of test instances, ISA constructs a 2-dimensional instance space to expose similarities and differences among test instances. This instance space is constructed using a chosen set of instance features. By giving a visual representation of where instances are located within the instance space, and how different algorithms perform in different regions of the space, meaningful insights on algorithmic behaviour can be obtained.

Fig. 1 illustrates the conceptual framework behind ISA. The set $\mathcal{P}$ denotes the *problem space* and contains all possible instances of a particular problem. The set $\mathbf{I} \subset \mathcal{P}$ is the *subset of instances* for which feature values and algorithmic performance are known. These are the instances that will be used for the analysis of the algorithms. For a given instance $x \in \mathbf{I}$, its feature values are represented by $\mathbf{f}_x \in \mathcal{F}$, with $\mathcal{F}$ representing the *feature space*. The set $\mathcal{A}$ represents the *algorithm space*, i.e. the set of all algorithms available to solve the chosen problem. The performance of a given algorithm $\alpha \in \mathcal{A}$ in an instance $x \in \mathbf{I}$ is given by $y_{\alpha,x} \in \mathcal{Y}$, where $\mathcal{Y}$ represents the *performance space*. A user-defined measure of "good" performance, either absolute or relative, is required to compare algorithms and analyse in which regions the performance of a particular algorithm is considered to be good. Finally, ISA aims to learn the mapping $S(\cdot)$ from a position of an instance in the 2-dimensional instance space to the algorithm with the best performance. This mapping

is used to estimate the *footprint* $\varphi(\mathbf{y}_{\alpha,\mathbf{I}})$ of each algorithm. This is defined as a region within the instance space for which ISA has statistically inferred good performance. This visual property of algorithm performance is what can be used to draw conclusions on the strengths and weaknesses of algorithms, and for which types of instances a particular algorithm should be used. A MATLAB toolkit that performs ISA automatically [32] is publicly available, and the online tool Melbourne Algorithm Test Instance Library with Data Analytics (MATILDA) [43] provides tutorials, examples and an interface for online analysis.

The collection $\{\mathcal{P}, \mathcal{F}, \mathcal{A}, \mathcal{Y}\}$ denotes the problem's *metadata*. In practice, before running ISA one must choose subsets from each of the elements in the metadata. That is, one must choose which instances, features, algorithms and algorithm performance measures to use in the analysis. Choosing both the algorithms and the performance metric is conditional on the user's need, namely what kind of analysis is required. The choice of features can rely upon the automated feature selection provided by the implemented toolkit, which is denoted Selection of Instance Features to Explain Difficulty (SIFTED). This selection first discards all candidate features which show low correlation with the performance of all algorithms. That is, features are only considered if they have an absolute correlation of at least 0.3 with the performance of at least one of the algorithms being analysed. This initial culling is performed as ISA relies on related global linear trends between the feature values and algorithm performance metric when generating the 2-dimensional projection of the instance space. These global trends will be exploited by the projection method in ISA to ensure visual interpretability of relationships once the data has been projected onto 2 dimensions. It is possible for the user to lower the correlation threshold to discard less features; however in this study a large amount of features remain after this initial culling and therefore this default threshold is used. Next the remaining features are grouped by similarity using k-means clustering, where k is a user specified number of desired features. MATILDA suggests a value of k based on the silhouette values for different numbers of clusters given the input metadata, although choosing between 3 and 10 features is recommended. The final stage of the feature selection consists of an optimisation problem to find k features taking one from each of the k clusters which minimise algorithm performance prediction error when projected into a temporary 2-dimensional space. The temporary 2-dimensional projection is created using Principal Component Analysis (PCA), and the temporary prediction of algorithm performance is assessed using a set of Random Forest (RF) models. Note that both PCA and RF are not used at later stages of ISA; they are only used here as a proxy both of the projection and the prediction of model performance due to their speed during the feature selection problem. Further details of this procedure are given by Smith-Miles and Muñoz [42].

Choosing a subset $\mathbf{I} \subset \mathcal{P}$ of instances can be a harder problem, as recently shown by Alipour et al. [1]. Their work showed that over-representation of types of instances in $\mathbf{I}$ can lead to bias in the overall analysis of ISA. They proposed an instance filtering technique that removes "similar" instances to generate an unbiased test suite. Two instances x_i and x_j with feature vectors $\mathbf{f}_i$ and $\mathbf{f}_j$ are deemed to be similar if $\|\mathbf{f}_i - \mathbf{f}_j\| \leq \theta$, where θ is a user-defined similarity threshold. The instance filtering algorithm simply iteratively removes instances from an initial set if at least one other "similar" instance is still in the set. This generates a set of "dissimilar" instances, denoted $\mathcal{D}_\theta$, so that for every pair of instances $x_i, x_j \in \mathcal{D}_\theta$, $\|\mathbf{f}_i - \mathbf{f}_j\| > \theta$.

Based on both the algorithm performance of each algorithm, and a user-defined measure of "good" performance, a binary performance vector $\boldsymbol{\delta}_i$ for each instance i can be defined, where the k^{th} entry of $\boldsymbol{\delta}_i$ is 1 if the performance of algorithm α_k is deemed "good" for instance i, and 0 otherwise. As the algorithm performance for instances that are deemed to be similar is assumed to be the same, a more intricate instance filtering procedure uses both the feature vectors $\mathbf{f}_i$ and binary performance vectors $\boldsymbol{\delta}_i$ when choosing which instances to filter out. Specifically, starting from a set of instances $\mathbf{I}$, an instance i is removed from $\mathbf{I}$ if there exists an instance j with $\|\mathbf{f}_i - \mathbf{f}_j\| \leq \theta$ and $\boldsymbol{\delta}_i = \boldsymbol{\delta}_j$. Instances that are similar feature-wise but have different binary performances are said to violate the similarity assumption. These instances should be kept in order to maintain valuable information in the set. Therefore, in this second filtering procedure, a set of dissimilar instances $\mathcal{D}_\theta$ is found as well as the set of instances $\mathcal{V}_\theta$ which violate the similarity assumption. The chosen test instances lie in the union of these two sets, known as the critical set $\mathcal{C}_\theta := \mathcal{D}_\theta \cup \mathcal{V}_\theta$. This set satisfies the property that for every pair of instances $x_i, x_j \in \mathcal{C}_\theta$ either $\|\mathbf{f}_i - \mathbf{f}_j\| > \theta$ or $\boldsymbol{\delta}_i \neq \boldsymbol{\delta}_j$. The user-defined θ determines the trade-off between the diversity of the final set of instances, and uniformity in feature-distance between instances. In order to choose this θ, a uniformity measure $\mathrm{u}_{\mathcal{D}_\theta}$ is introduced, with

$$\mathrm{u}_{\mathcal{D}_\theta} = 1 - \frac{\sigma_{\mathcal{D}_\theta^{NN}}}{\mu_{\mathcal{D}_\theta^{NN}}}$$

where $\mathcal{D}_\theta^{NN} = \{\min_{j \in \mathcal{D}_\theta, j \neq i} \|\mathbf{f}_i - \mathbf{f}_j\| \quad | \quad i \in \mathcal{D}_\theta\}$ is the nearest neighbour feature distance for the instances in $\mathcal{D}_\theta$, and $\mu_{\mathcal{D}_\theta^{NN}}$ and $\sigma_{\mathcal{D}_\theta^{NN}}$ are its mean and variance, respectively. The authors recommend scaling $\mathrm{u}_{\mathcal{D}_\theta}$ so that its values lie between 0 and 1, and choosing the smallest θ for which $\mathrm{u}_{\mathcal{D}_\theta} \geq 0.5$. Finally, it is worth noting that it is possible to implement the filtering procedure so that certain instances are prioritised when choosing which instances to keep. This characteristic will be employed below to retain as many literature instances as possible. For further details on this procedure, the reader is directed to the work of Alipour et al. [1].

Having chosen the algorithms, algorithm performance metric, features and instances, ISA next applies the Projecting Instances with Linearly Observable Trends (PILOT) procedure. This algorithm finds a projection of the k-dimensional feature space defined by the k selected features into a 2-dimensional space. The aim is to generate a projection which allows for the intuitive identification of relationships between instance features and algorithm performance. To this end PILOT finds a projection for which individual linear models can accurately predict feature and algorithm performance values based only on the 2-dimensional coordinates of the instances. Mathematically this is represented by the optimisation problem

$$\min \|\mathbf{F} - \mathbf{BZ}\|_F^2 + \|\mathbf{Y} - \mathbf{CZ}\|_F^2$$

$$
\begin{aligned}
s.t. \quad & \mathbf{Z} = \mathbf{AF} \\
& \mathbf{A} \in \mathbb{R}^{2\times k} \\
& \mathbf{B} \in \mathbb{R}^{k\times 2} \\
& \mathbf{C} \in \mathbb{R}^{m\times 2}
\end{aligned}
$$

where $\mathbf{F}$ is the $k \times n$ matrix containing the k feature values for each of the n chosen instances, $\mathbf{Y}$ is the $m \times n$ matrix containing the m algorithm performances for each of the n chosen instances, and $\|\cdot\|_F^2$ is the Frobenius norm. The optimisation finds entries for the matrix $\mathbf{A}$ which represents the projection from k-dimensional space onto 2 dimensions, with $\mathbf{Z}$ being the $2 \times n$ matrix containing the new pair of coordinates for each of the instances given the projection. The matrices $\mathbf{B}$ and $\mathbf{C}$ represent linear models for each of the features and models, respectively. Thus the objective function minimises the difference between the actual and predicted values for both the features and algorithm performance given a 2-dimensional projection and the use of linear models. Further details of this procedure can be found in the work of Muñoz et al. [31] and of Smith-Miles and Muñoz [42]. Having obtained this projection, models can be trained using Support Vector Machines (SVMs) which predict when each of the algorithms will perform well given only the 2-dimensional coordinates of each of the instances. Visually observing when algorithms are predicted to perform well and comparing with feature value trends can provide valuable and intuitive insights into the impact of instance characteristics in algorithm performance, as will be shown in this study.

2.2. Instances, features and algorithms

2.2.1. Instances

The predominant characteristic of Bf-EBB problems is the existence of two black boxes that can be queried for information. Both the high-fidelity black box, denoted f_h, and the low-fidelity black box, denoted f_l, are defined within a hypercube Ω. That is

$$
\begin{aligned}
f_h, f_l : \quad & \Omega \to \mathbb{R} \\
\Omega = [x_1^{\perp}, x_1^{\top}] \times \cdots \times [x_d^{\perp}, x_d^{\top}]
\end{aligned}
$$

where $\mathbf{x}^{\perp} = (x_1^{\perp}, \ldots, x_d^{\perp})$ and $\mathbf{x}^{\top} = (x_1^{\top}, \ldots, x_d^{\top})$ are the vectors representing the lower and upper bounds of Ω, and $d \in \mathbb{N}$ is the dimension (i.e. number of variables) of the problem. It is worth noting that in some cases in the literature, one or both sources can be assumed to be stochastic. That is not the case in this study, as we assume that both f_h and f_l are deterministic. Whilst f_h is assumed to be expensive, how this cost is translated to the problem statement can vary. The two most common settings are providing an existing sample of both f_h and f_l, or providing a maximum number of times that an algorithm can query f_h. Furthermore, as discussed in the introduction, the aim can be to either minimise f_h, or to train a surrogate of f_h which is as accurate as possible.

In this study, the focus is put on what is arguably the simplest variant; where the sampling of both f_h and f_l has already been conducted, no further sampling is possible, and the only decision is to choose which surrogate model to train. An instance in this study is therefore defined by the tuple (f_h, f_l, n_h, n_l), where n_h and n_l are the number of locations at which the values of f_h and f_l are known, respectively. As both f_h and, to a lesser extent, f_l are considered to be expensive, n_h and n_l should be kept relatively small. Therefore the values used are $n_l \in \{4d, 8d, 12d, 16d, 20d\}$ and $n_h \in \{2d, 4d, \ldots, 18d, 20d\}$, with $n_h \leq n_l$. In order to generate the set $\mathbf{I}$ however, a choice must be made of which function pairs (f_h, f_l) to include from a large candidate pool. This is expanded on in the next section.

2.2.2. Algorithms and performance

The aim of this study is to characterise when a low-fidelity source can be harmful when constructing a surrogate model. As such, two types of models are compared, namely a surrogate model which is trained using only high-fidelity data, and a surrogate model which is trained using both high and low-fidelity data. As the findings should be as widely applicable as possible to the field, two well-established models are used, namely Kriging [20,27,16] and Co-Kriging [10]. For an in-depth mathematical derivation, the reader is directed to the work of Jones [16] and Forrester et al. [10]; an introduction to both is also given in the implementation of both techniques used in this work [3]. The wide usage of these two techniques in the literature can be attributed to three main components. The first is their strong theoretical backing, which assumes the observed data from either source is the result of a multivariate random normal variable, and as such the training of the model's hyperparameters is done by finding the maximum likelihood estimators of a probability distribution. The second is the high accuracy these models tend to have even with relatively small amounts of data. This is in part due to the high number of hyperparameters which allows the models to capture how the objective function changes along each of the input dimensions. It is worth noting that the training of these hyperparameters can take a long time when a lot of data is available. This however is not a problem for Bf-EBB problems, as one of its key assumptions is that the data available is scarce. Finally, an attractive characteristic of these models once they have been trained is the existence of not only a prediction of the objective function value, but also an error estimate for the prediction. Whilst this characteristic is of no benefit if no further sampling is allowed, further sampling can be balanced between exploration and exploitation of the sample space through the use of this error metric.

By taking $\mathcal{A} = \{\text{Kriging, Co-Kriging}\}$, the next requirement to perform ISA is the definition of model performance, as well as a definition for "good" model performance. The performance of the two methods in an instance (f_h, f_l, n_h, n_l) is assessed through a statistical comparison via repetitions of the training of each of the models. At each repetition, the sets $\mathbf{X}_h, \mathbf{y}_h, \mathbf{X}_l$ and $\mathbf{y}_l$ defined as

$$\mathbf{X}_l = \{\mathbf{x}_1^l, \mathbf{x}_2^l, \dots, \mathbf{x}_{n_l}^l\} \subset \Omega$$
$$\mathbf{X}_h = \{\mathbf{x}_1^h, \mathbf{x}_2^h, \dots, \mathbf{x}_{n_h}^h\} \subset \Omega$$
$$\mathbf{y}_l = \{f_l(\mathbf{x}_1^l), \dots, f_l(\mathbf{x}_{n_l}^l)\}$$
$$\mathbf{y}_h = \{f_h(\mathbf{x}_1^h), \dots, f_h(\mathbf{x}_{n_h}^h)\}$$

are constructed, with $|\mathbf{X}_l| = n_l$ and $|\mathbf{X}_h| = n_h$. These two sampling plans are often generated by space-filling methods, as these are preferred by practitioners when the sources being sampled are deterministic [9] (as is assumed in this study). Furthermore, assuming that the sampling plans are well spread out allows for a statistical comparison of the models' performance by measuring their accuracy on multiple sampling plans which are spread out but are non-identical. The generation of all sets $\mathbf{X}_l$ and $\mathbf{X}_h$ therefore follows the approach of Forrester et al. [10]. That is, an initial random LHS is optimised to be locally optimal in terms of the minimum distance between any pair of points. This is achieved by iteratively swapping one of the entries between two pairs of points, as long as this increases the distance between the two closest points in the set. Once this can no longer be achieved, the resulting set is locally optimal. Similarly, the set $\mathbf{X}_h$ is constructed as a subset of $\mathbf{X}_l$ and also made locally optimal in terms of the minimum distance. This is achieved by starting with a random subset of $\mathbf{X}_l$, and iteratively swapping a point inside the subset for a point outside the subset as long as this increases the minimum distance between any pair of points. Once this can no longer be done, the resulting set is locally optimal. It is worth noting that both of these "brute force" approaches can be lengthy for larger sample sizes. For this reason as well as for reproducibility purposes, the sampling plans used in this study are made available in the implementation repository [3]. Note also that the impact of skewed sampling plans as well as the usage of other sampling generation techniques (e.g. Sobol sampling [44]) on overall model accuracy is left to future work.

Once the sets $\mathbf{X}_h, \mathbf{y}_h, \mathbf{X}_l$ and $\mathbf{y}_l$ have been generated, a Kriging model is trained using $\mathbf{X}_h$ and $\mathbf{y}_h$, and a Co-Kriging model is trained using $\mathbf{X}_h, \mathbf{y}_h, \mathbf{X}_l$ and $\mathbf{y}_l$. Note that in the applied setting the accuracy of a surrogate model cannot be assessed as it cannot be compared with the values of the true objective function. However as is standard practice when using synthetic test functions, here a large sample set is used to assess model accuracy. That is we generate the datasets $\mathbf{X}$ and $\mathbf{y}$ defined as

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \subset \Omega$$
$$\mathbf{y} = \{f_h(\mathbf{x}_1), \dots, f_h(\mathbf{x}_N)\}$$

with $N = 1000d$. Given a surrogate model $s : \Omega \to \mathbb{R}$ with a predicted objective function value $s(\mathbf{x}_i)$ at location $\mathbf{x}_i$, its accuracy is assessed using its Pearson's sample correlation P_{corr} with the objective function, given by

$$P_{corr} = \frac{1}{N-1}\left(\frac{\sum_{i=1}^N (f_h(\mathbf{x}_i) - \bar{y})(s(\mathbf{x}_i) - \bar{S})}{s_Y s_S}\right)$$

where
$$\bar{y} = \frac{1}{N}\sum_{i=1}^N f_h(\mathbf{x}_i)$$
$$s_Y = \left[\frac{\sum_{i=1}^N (f_h(\mathbf{x}_i) - \bar{y})^2}{N-1}\right]^{1/2}$$
$$\bar{S} = \frac{1}{N}\sum_{i=1}^N s(\mathbf{x}_i)$$
$$s_S = \left[\frac{\sum_{i=1}^N (s(\mathbf{x}_i) - \bar{S})^2}{N-1}\right]^{1/2}$$

Here, a high P_{corr} indicates high model accuracy as the model predicts the behaviour of the objective function. Finally, as the analysis of interest is not to find when Kriging and Co-Kriging perform well, but when they perform well comparatively, the performance measure used is relative in nature. That is, after 40 repetitions of generating the datasets and training the models, the resulting 40 Kriging and Co-Kriging model accuracies are compared using the Wilcoxon test [50]. This test is often used in clinical trials [37] to assess whether patients provided with a new treatment improve in health due to the treatment (rather than random factors). This is assessed by testing the base assumption (known as the null hypothesis) that a treatment has no effect given the observed response of the patients. If the observed response is less than 5% likely assuming the null hypothesis is true, the null hypothesis is said to be rejected with 95% confidence. The aim of this test is to reject the null hypothesis with sufficient certainty and to therefore continue with the development of the treatment being analysed.

In this paper, for each of the two modelling methods a one-sided Wilcoxon statistical test is conducted to test whether a given modelling choice under-performs when compared to its competitor. That is, the performance of using a modelling method $M \in \{\text{Kriging, Co-Kriging}\}$ on a particular instance is the p-value (i.e. the probability) of having observed the 40 Kriging and Co-Kriging model accuracies given the assumption that using M leads to models with an accuracy at least 0.001 lower than its competitor. A method is labelled as good if the respective p-value is below 0.05. In practical terms, this metric and labelling approach can be thought of as a "careful" approach to labelling a model as good, where a technique is assumed to be bad unless there is strong

statistical evidence to contradict this claim. It is worth noting that both p-values can be below 0.05 (i.e. the performance is "good" for both) if the performance of both models is statistically similar. Finally, for roughly 13% of the instances investigated neither method is labelled as good. Acknowledging that a method must be chosen even when there is no clear safe choice, for instances of this type the method with the lowest p-value (i.e. the one for which the null hypothesis is least likely to be true) is labelled as having good performance.

2.2.3. Features

The features used in this study characterise the relationship between f_h and f_l, the landscape of f_h, f_l, and $f_h - f_l$ (the difference between the two sources), as well as the amount of data available when training the models. The relationship features used are the Correlation Coefficient (CC) and Relative Root Mean Squared error ($RRMSE$) proposed by Toal [48] which indicate the overall quality of f_l relative to f_h, as well as the Local Correlation Coefficient features $LCC^r_{mean}, LCC^r_{sd}, LCC^r_{coeff}$ and LCC^r_p proposed by Andrés-Thió et al. [4] which quantify the distribution of the local quality of f_l. The value of p gives a threshold of "good" local correlation, and the value of r impacts the radius of the d-dimensional ball used to calculate the local correlations. Here $p \in \{0.1, 0.2, \dots, 0.9, 0.95, 0.975\}$, and $r = \{0.2, 0.2^{1/d}\}$, where d is the problem dimension. The value $r = 0.2$ is used as it has been shown to lead to helpful features in a synthetic setting, and the value $r = 0.2^{1/d}$ is used here for the first time to assess whether growing the hypervolume of the sphere constant relative to the volume of a hypersphere encompassing the whole sample space also leads to good features. The mathematical definition of these features is given in Appendix A. The landscape features are taken from the R package flacco [18], which is made up of 17 feature sets, such as classic Exploratory Landscape Analysis (ELA) or Information Content feature sets. Not all features in the package are used however; features that split up the sample space into cells cannot be used on problems of high dimension ($d \geq 20$), and features that are related to the function's domain and range are not used as they are not comparable between functions. The features which characterise the amount of data available are the actual sizes of the data i.e. n_h and n_l, the sizes relative to the dimension $\frac{n_h}{d}$ and $\frac{n_l}{d}$, and the ratio of high to low-fidelity data $\frac{n_h}{n_l}$. Finally, the problem dimension itself is taken as a feature as well. All of the features used are given in Table 1, along with their range of values.

The feature values must be processed before being used both for instance filtering and ISA. It is a desirable property for the processed feature values to have similar ranges. Therefore, for the features which are known to be bounded a linear transformation is applied so that the range of the transformed features is $[-2, 2]$. This linear transformation is also applied to the dispersion features $\overline{DISP}_\epsilon$ and $\widetilde{DISP}_\epsilon$, as despite being unbounded in theory, in practice the distribution of the feature values is close to uniform $[0, 1]$. The Box-Cox transformation is applied to the remaining (unbound) features to remove the effect of outliers, and then transformed a second time in order to produce feature values that have a standard normal distribution. Finally, any remaining outliers are bound to the range $[-4, 4]$. As 95% of the values are within $[-2, 2]$, this makes them comparable to the features processed with a linear transformation.

When choosing a set of function pairs (f_h, f_l) (as described in the next section) in order to construct the benchmark suites, the feature values are calculated using a very large sample of size $1000d$, where d is the problem dimension. Once the set of instances $\mathbf{I}$ has been constructed, all features are calculated using only the sample used to train the Kriging and Co-Kriging models. As for each instance, 40 repetitions are conducted (and therefore 40 high- and low-fidelity data sets are generated), and the sample feature values are the average of the feature value calculated with each of the 40 data sets.

3. ISA of Bf-EBB surrogate modelling with fixed sample

3.1. Generating an unbiased test suite

As stated in the previous Section, choosing a set of function pairs (f_h, f_l) to generate the set of instances $\mathbf{I}$ is not a trivial problem. A large set of candidates has been implemented to be considered for this purpose. This set of candidates consists of over 200 literature function pairs, supplemented by over 80,000 function pairs generated by the instance-generating procedure proposed by Andrés-Thió et al. [4]. These disturbance-based function pairs are generated by starting with a chosen high-fidelity function, and adding a "disturbance" either around a particular objective function value, or near disturbance "centres" in the sample space. The literature function pairs are gathered from a variety of studies [38,26,22,23,53,40,7,54,46,33] and include function pairs of dimensions $d \in \{1, 2, 3, 4, 5, 6, 8, 10, 20\}$. The 38 high-fidelity functions from this set as well as functions from the COCO test suite [14] with $d \in \{2, 4, 6, 8, 10\}$ are used to generate the disturbance-based function pairs. For further details on the implemented function pairs, the reader is directed to the publicly available implementation [2].

It is worth stressing here that both the initial number of considered function pairs (over 200 from the literature and over 80,000 newly created) as well as the number of unique high-fidelity functions used to generate these pairs (159 from both the COCO test suite and existing literature studies) present a significant increase over previous studies of this kind. For reference, the influential work of Toal [48] tested 40 function pairs derived from 4 unique high-fidelity functions, and the work of Shi et al. [40] considered 20 function pairs generated from 20 unique high-fidelity functions and is one of the studies with the largest number of benchmarks in the literature. Whilst the functions studied in this work are not exhaustive in terms of all possible applications of Mf-EBB methods (such as other industrial applications or hyper-parameter tuning [52]), we consider the analysis and ensuing findings to be relevant to most practitioners in the field.

As algorithm performance cannot be assessed for such a large candidate set of function pairs, the simpler instance filtering procedure is first used which relies only on feature values. The aim here is to generate a set of function pairs that can be used in future research to allow for comparable analysis across studies. In order to prevent a certain type of feature to have an unwanted weight

Table 1
A short description of the features used in this study, as well as the range of feature values for each feature.

Method	Feature name	Description	Range
Overall quality of f_l [48]	CC	Overall correlation between f_h and f_l	$[0,1]$
	$RRMSE$	Overall error between f_h and f_l	$[0,\infty)$
Local quality of f_l [4]	LCC^r_p	Proportion of time the local correlation between f_h and f_l is above a given $p \in [0,1]$	[0,1]
	LCC^r_{mean}	Average local correlation between f_h and f_l	[0,1]
	LCC^r_{sd}	Standard deviation of the local correlation between f_h and f_l	$[0,1]$
	LCC^r_{coeff}	Coefficient of variation of the local correlation between f_h and f_l	$[0,\infty)$
Distribution (ELA distr) [28]	Skewness	Skewness of the objective function values	$\mathbb{R}$
	Kurtosis	Kurtosis of the objective function values	$\mathbb{R}$
	Peaks	Estimated number of peaks in the objective function	$[1,\infty)$
Levelset prediction (ELA levelset) [28]	$MMCE^q_{lda}$	Mean misclassification error of a predictive linear model with data quantile split $q \in \{0.1, 0.25, 0.5\}$	$[0,1]$
Meta modelling (ELA meta) [28]	$\tilde{R}^2_L$	Adjusted R^2 of a linear model without interactions	$[0,1]$
	$\tilde{R}^2_{LI}$	Adjusted R^2 of a linear model with interactions	$[0,1]$
	$\tilde{R}^2_Q$	Adjusted R^2 of a quadratic model without interactions	$[0,1]$
	$\tilde{R}^2_{QI}$	Adjusted R^2 of a quadratic model with interactions	$[0,1]$
	CN_L	Ratio of the minimum and maximum absolute coefficients of a linear model without interactions	$[0,\infty)$
	CN_Q	Ratio of the minimum and maximum absolute coefficients of a quadratic model without interactions	$[0,\infty)$
Information content [30]	H_{max}	Maximum information content of fitness sequence	$[0,1]$
	ϵ_S	Settling sensitivity; epsilon for which sequence contains 0 almost exclusively	$\mathbb{R}$
	ϵ_{max}	Value for which $H(\epsilon_{max}) = H_{max}$	$\mathbb{R}$
	ϵ_{ratio}	Ratio of partial information sensitivity	$\mathbb{R}$
	M_0	Initial partial information	$[0,1]$
Nearest Better Clustering [36]	NBC_{mean}	Ratio of the mean distance of a point's closest neighbour, and a point's better closest neighbour	$[0,1]$
	NBC_{sd}	Ratio of the standard deviation of the distance of a point's closest neighbour, and a point's better closest neighbour	$[0,1]$
	NBC_{coeff}	Coefficient of variation of the ratios of the distance of the closest neighbour, and the distance of the better closest neighbour	$[0,\infty)$
	NBC_{corr}	Correlation between the distance of a point's nearest neighbour and a point's nearest neighbour with a lower objective function	$[-1,1]$
	$NBC_{fitcorr}$	Correlation between a point's objective function value and its "in-degree"	$[-1,1]$
Principal component analysis	PCA_{corr}	Relative amount of principal components required to explain high variability in the sample correlation	$[0,1]$
	PCA_{cov}	Relative amount of principal components required to explain high variability in the sample covariance	$[0,1]$
Dispersion [24]	$\overline{DISP}_\epsilon$	Ratio of the mean distance between all points, and the mean distance between the ϵ% best points, $\epsilon \in \{2,5,10,25\}$	$[0,\infty)$
	$\widehat{DISP}_\epsilon$	Ratio of the median distance between all points, and the median distance between the ϵ% best points, $\epsilon \in \{2,5,10,25\}$	$[0,\infty)$
Data budget	$B_h = n_h$	Number of high-fidelity samples available	$[2,400]$
	$B_l = n_l$	Number of low-fidelity samples available	$[2,400]$
	$B^r_h = n_h/d$	Relative number of high-fidelity samples available	$[2,20]$
	$B^r_l = n_l/d$	Relative number of low-fidelity samples available	$[2,20]$
	$B^r = n_h/nl$	Ratio of number of high-fidelity and low-fidelity samples	$[0,1]$
Problem dimension	d		$[1,20]$

when deciding which instances to filter out, only the "real" (i.e. calculated with a large sample) features CC, $RRMSE$, $LCC_{0.5}^{0.2^{1/d}}$, $LCC_{0.95}^{0.2^{1/d}}$, $LCC_{sd}^{0.2^{1/d}}$, Skewness, Kurtosis, Peaks, $MMCE_{lda}^{0.25}$, $\bar{R}_Q^2$, CN_Q, H_{max}, H_0, ϵ_{max}, ϵ_{ratio}, $\overline{DISP}_{10}$, NBC_{mean}, NBC_{sd} and PCA_{cov} are used, calculated for f_h, f_l and $f_h - f_l$. The filtering prioritises literature function pairs, only choosing newly generated disturbance-based function pairs if they are sufficiently different from every literature function implemented. The filtering leads to a set of 312 function pairs, 42 from the literature and 280 disturbance-based function pairs. It is worth noting here that it can be seen the generating procedure can be very helpful in generating a varied set, as the objective filtering of the instances chooses a large number of non-literature function pairs that are demonstrably different.

An additional set of function pairs from the SOLAR simulation engine [12] are also used in this study. This simulation engine simulates the behaviour of a solar power-plant, and provides truly black-box sources in the sense that sampling a source requires running processes that simulate the physical processes of the plant and for which no analytical expression is known. This simulator contains a variety of black-box objective functions, some of which are constrained and/or stochastic. The tenth objective function has 5 inputs, is constrained only to lie in a hypercube, is deterministic, and can be queried along with a fidelity value $fid \in (0, 1]$, where setting $fid = 1$ returns the true value of the objective function. Defining $SOLAR_{fid}$ with $f \in \{0.1, 0.2, \ldots, 0.8, 0.9\}$ as the function pair (f_h, f_l) where f_h and f_l query the objective function with fidelity values of 1 and fid respectively, these additional nine function pairs are added to the 312 synthetic function pairs already chosen, bringing the total to 321 function pairs. Note that the SOLAR function pairs were not included in the instance filtering as they are all deemed to be valuable since they lead to "real" black-box instances. Finding whether the synthetic instances are similar to the SOLAR instances can guide the creation of further synthetic instances.

Having chosen the set of function pairs, the set $\mathbf{I}$ is created using each of the 321 function pairs with $n_l \in \{4d, 8d, 12d, 16d, 20d\}$, $n_h \in \{2d, 4d, \ldots, 18d, 20d\}$ and $n_h \leq n_l$, leading to a total of 9630 instances. As described in the previous section, for each instance 40 Kriging and Co-Kriging models are trained and their performances are compared using the Wilcoxon test. The remaining steps include choosing a set of relevant features, and the construction and analysis of the instance space. Before each of these steps, instance filtering is conducted to avoid bias both when choosing the features and when conducting the analysis as outlined by Alipour et al. [1]. Note that once again, certain instances are prioritised when choosing which instances to keep. Namely, the highest priority is given to SOLAR-based instances, the second-highest to literature-based instances, and the lowest priority is given to disturbance-based instances. The results are presented next.

3.2. Algorithm performance analysis via ISA

Using the automated feature selection procedure of the MATILDA toolkit described in Section 2.1 leads to 9 features being chosen. This signifies that each instance lies in a 9-dimensional space, where each of the coordinates is the value of each of the 9 features. In order to visualise this space, a projection onto 2-dimensional space is achieved via the projection matrix given in Equation (1). This matrix is generated following the procedure described in Section 2.1 and aims to generate a 2-dimensional space where global trends in the feature and performance values can be visualized. Recall that the features $LCC_{sd}^{0.2}$, $LCC_{0.5}^{0.2^{1/d}}$, $LCC_{0.9}^{0.2^{1/d}}$ and $RRMSE$ all measure the quality of f_l relative to f_h, and the feature B^r is the ratio $\frac{n_h}{n_l}$, namely the amount of high-fidelity data available relative to the amount of low-fidelity data available. The remaining four features are landscape features, two of which characterise the landscape of f_h, whilst the other two characterise the landscape of the difference $f_h - f_l$ of the two sources. Further discussion on the features, their values and their impact on Kriging and Co-Kriging performance is given in Section 3.3. For now simply note that in all plots that follow, different properties of the constructed space in two dimensions are shown, where each point represents an instance and its coordinates (z_1, z_2) represent only its projection in two dimensions of a 9-dimensional space. It is important also to note here that, as described in Section 2.1, each of the selected features is chosen from a set of features in clusters which have similar values. Therefore whilst it is possible to choose different features to the ones presented here, the ensuing analysis from doing so will likely be qualitatively similar.

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0.1649 & 0.5368 \\ -0.0704 & 0.439 \\ 0.4054 & -0.0672 \\ 0.4314 & -0.1441 \\ -0.4726 & 0.0305 \\ -0.1082 & -0.418 \\ -0.1447 & -0.4773 \\ 0.3435 & -0.0636 \\ 0.0532 & -0.4654 \end{bmatrix}^\top \begin{bmatrix} B^r \\ LCC_{sd}^{0.2} \\ LCC_{0.5}^{0.2^{1/d}} \\ LCC_{0.9}^{0.2^{1/d}} \\ RRMSE \\ f_h,\ MMCE_{lda}^{0.5} \\ f_h,\ H_0 \\ f_h - f_l,\ NBC_{fitcorr} \\ f_h - f_l,\ \bar{R}_{LI}^2 \end{bmatrix} \tag{1}$$

Fig. 2 shows the distributions of instances from different sources in the space. The benefit of supplementing literature instances with the disturbance-based procedure is clear in this plot, as despite not prioritising disturbance-based instances when applying the filtering procedure, many of them remain in the final set. This indicates that they are interesting either because they lie in regions of the space where traditional instances do not exist, or because they are similar to other instances but have different algorithm

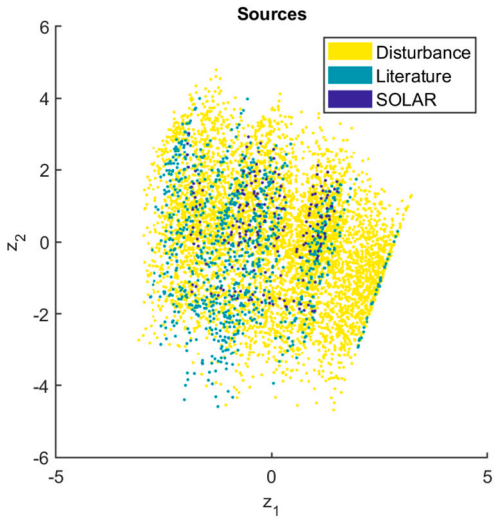


Fig. 2. Sources of the function pairs used in each of the instances, where the dark blue points represent instances created from the SOLAR simulation, the light blue points represent classical literature instances, and the yellow points represent disturbance-based instances. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

performance. Furthermore, it can be seen that both classical synthetic instances and disturbance-based instances can be similar to the truly black-box instances from the SOLAR simulator. The fact that the SOLAR instances lie only towards the centre of the space seems to indicate that the synthetic instances which lie towards the border might not be representing realistic examples, at least when compared to a solar energy plant design problem. It is entirely possible however that other industrial problems such as a combustion application [15] might also lie in the border of the space and therefore be similar to synthetic instances which lie there. The true applicability of these synthetic instances to industrial problems remains to be verified in future work.

Figs. 3a and 3b show the binary performance of the Kriging and Co-Kriging models respectively. Recall that a model is said to have good performance if the null hypothesis that the model is less accurate than its competitor can be rejected with at least 95% confidence. When neither model is labelled as good using this criterion, the model with the lowest p-value is labelled as good. Fig. 3a clearly shows a bottom-right region where Kriging has bad performance, indicating that ignoring the low-fidelity data is counter-productive. On the other hand, Fig. 3b shows that, whilst Co-Kriging performs well in the bottom-right part of the space, moving towards the top-left leads to a region containing a lot of instances where Co-Kriging performs badly. It is worth stressing once again that it is possible for both models to perform well in the same instances, leading to blue regions in both plots. The displayed instance space indicates that one should not consider Kriging and Co-Kriging as complementary techniques. Rather, a more subtle description would be that of two specialised methods with requirements for good performance which are often (but not always) complementary. However, despite the top left region containing instances where Co-Kriging performs well, Kriging performs well in almost all instances in the same region, indicating that the safer approach is to use Kriging and therefore ignore f_l. Regions that are predominantly blue in both figures indicate easier regions in the space in the sense that using either method is valid. The characteristics of these types of instances are discussed in Section 3.3.

Given the good/bad labels of both techniques, two Support Vector Machines (SVMs) are trained to individually predict when Kriging and Co-Kriging can be used. The SVMs predictions for Kriging and Co-Kriging performance are shown in Figs. 4a and 4b, respectively. It can be seen that in both the top-left and top-right regions both Kriging and Co-Kriging are predicted to be good, whereas only Co-Kriging is predicted to perform well in the bottom-right of the space and only Kriging is predicted to perform well in the bottom-left of the space. Fig. 5 provides a visualisation of these regions by overlaying the regions with predicted good performance for Kriging and Co-Kriging. Note that rather than training the SVMs using the individual values of all 9 features, only the (z_1, z_2) coordinates are used. This is a key component to ISA, as the trained algorithm prediction emulates a human approach by assigning techniques to groups of similar instances, here defined as instances which are near each other in the instance space. A core benefit is the creation of predictions which are intuitive to the naked eye when compared to the feature values in the same regions as discussed below.

Based on the constructed SVMs, a simple selector can be created which assigns one of the two techniques to each of the instances in the space. This selector is constructed following a three step process by assigning either Kriging or Co-Kriging to each of the regions shown in Fig. 5. First, Kriging is selected for instances for which the Kriging SVM predicts good performance (corresponding to the yellow and blue regions), as this SVM is the more accurate of the two. For the remaining instances, Co-Kriging is selected when the Co-Kriging SVM predicts good performance, corresponding to the green region. This leaves a small amount of instances which lie in the purple region with no assigned method. Given that Kriging has good performance for 64.5% of the instances in this region (compared to 35.5% for Co-Kriging), it is assumed that Kriging should be used for future instances which lie in this region. The resulting selector is shown in Fig. 6. The construction process is reflected in this figure, as the selector is almost identical to the SVM which predicts when Kriging performance is good or bad. Intuitively, this is the result of the division being much clearer for Kriging

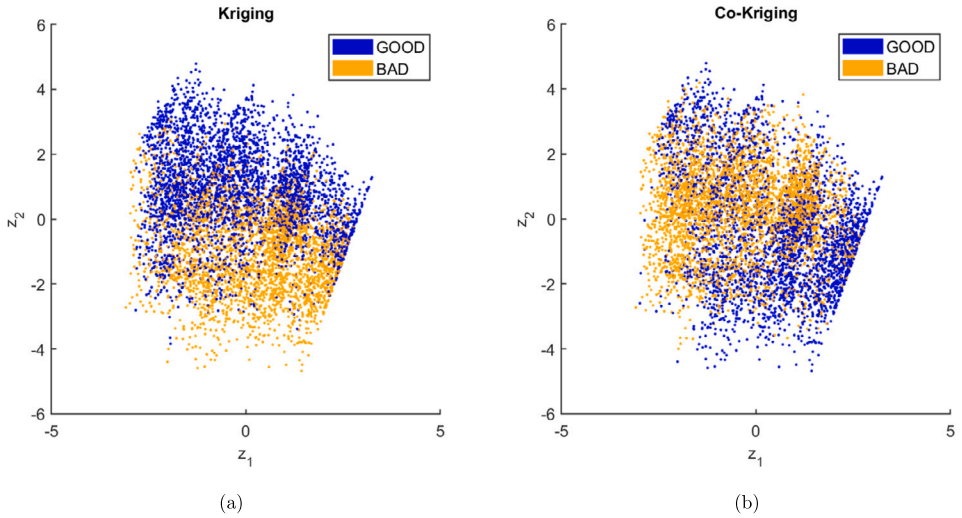


Fig. 3. Binary performance of (a) Kriging and (b) Co-Kriging models. The blue points represent instances for which the model's performance is labelled good, and the orange points represent instances for which the performance is labelled bad.

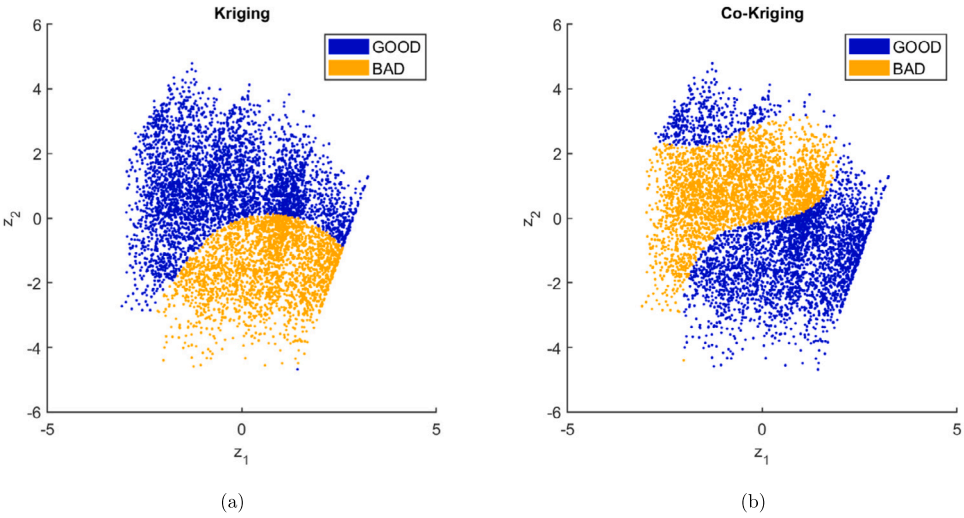


Fig. 4. SVM predictions of (a) Kriging and (b) Co-Kriging performance. Blue points represent instances where the model's performance is predicted to be good, and the orange points represent instances where the performance is predicted to be bad.

Table 2
Accuracy of the constructed SVMs which predict when each algorithm will perform well, as well as a selector which chooses an algorithm based on the location in the instance space. The second column shows the proportion of instances for which an algorithm is labelled good. The remaining columns show the statistical quality of the SVMs.

Algorithm	Pr(Good)	Accuracy	Precision	Recall
Kriging	0.595	76.8%	79.5%	82.3%
Co-Kriging	0.569	69.5%	74.5%	70.4%
Selector	0.787	-	-	-

performance than for Co-Kriging performance. This is further reflected by Table 2, which presents the performance of the two SVMs as well as the selector. The constructed SVM which predicts Kriging performance has a much higher accuracy, precision and recall than that of the SVM which predicts Co-Kriging performance. This implies it is a lot easier to predict when a low-fidelity source will be harmful than when it will be beneficial, indicating that a low-fidelity source should only be used if one can be fairly certain it will be an asset.

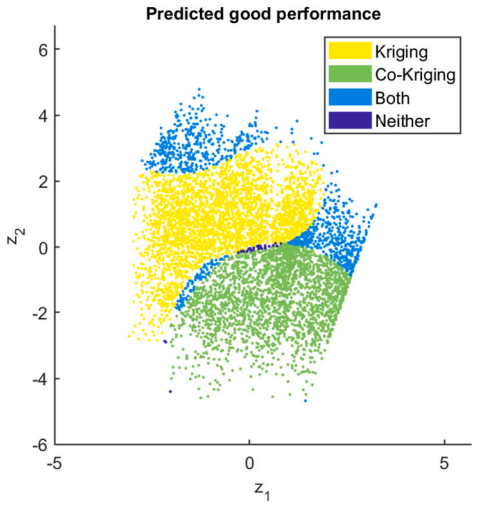


Fig. 5. Overlay of the regions where SVMs predict good model performance: In the yellow region only Kriging is predicted to have good performance, in the green region only Co-Kriging is predicted to have good performance, in the blue regions both Kriging and Co-Kriging are predicted to perform well, and in the small purple region where each SVM predicts underperformance.

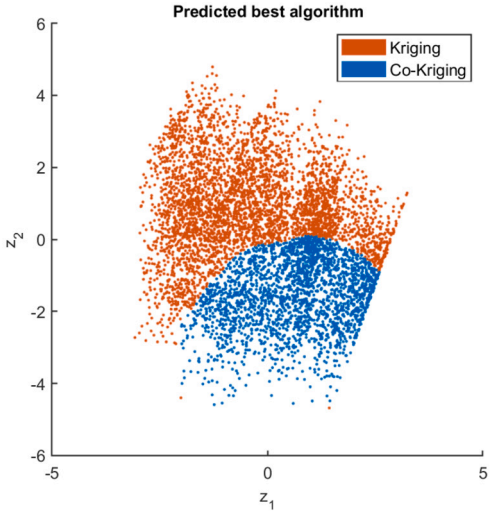


Fig. 6. SVM selector which predicts whether Kriging or Co-Kriging should be used. The red dots indicate instances for which Kriging is predicted to be best, and the blue dots indicate instances for which Co-Kriging is predicted to be best.

Finally, it is worth remarking that after filtering the metadata, the probability that Kriging will perform well is only slightly higher than that of Co-Kriging i.e. 0.595 vs. 0.569, respectively. Therefore, simply choosing to ignore or always use the low-fidelity source is the correct strategy only 59.5% and 56.9% of the time, respectively. The trained selector however correctly chooses a good model 78.7% of the time, which represents a significant improvement. Note that whilst it is common for Algorithm Selection methods to provide higher accuracy than choosing to always use any one algorithm (or in this case, model), what is surprising is the prediction capability can be derived from *approximated* features. As stated in the introduction, previous work of this kind [4,48] has always relied on features calculated with large samples which are not available in practice. This study provides the first indication of the usefulness of these features even when they are approximated using a small sample. This accuracy is also remarkable due to the fact that it is derived from a 2-dimensional projection of a 9-dimensional space, as this is a simplification of the true characteristics of the instances. Furthermore, the use of the filtering procedure ensures the trained selector does not benefit from an (incorrectly) biased benchmark suite containing many instances that are similar to one another both in terms of feature values and algorithm performance. The constructed SVMs are therefore considered reliable enough to assess how different features affect algorithm performance in the next section.

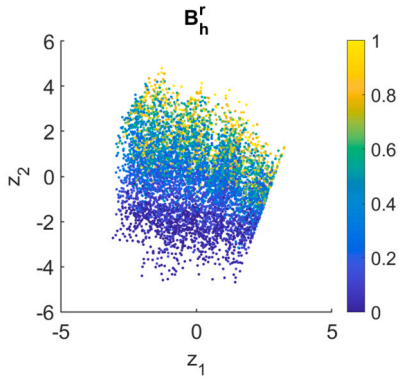


Fig. 7. Feature distribution of the budget feature B_h^r. This feature indicates that going from the right to the left of the space means going through instances with the lowest to the highest relative amount of high-fidelity data. Note the colour grading is relative, with a value of 1 indicating a real feature value of 20, and a value of 0 indicating a real feature value of 2.

3.3. Analysis of features

Before analysing the trends of the selected features against the predicted Kriging and Co-Kriging performance, informative conclusions can be drawn simply by looking at which features do not seem to help predict this performance. The automated feature selection of the implemented ISA toolkit follows a two-step process, first discarding any features that do not have an absolute correlation of at least 0.3 with either Kriging or Co-Kriging performance. Interestingly, all landscape features that characterise the low-fidelity source f_l are discarded here, as the highest absolute feature correlation is 0.121, an extremely low value. Furthermore, the features B_l and B_l^r which indicate how much low-fidelity data is available have almost zero correlation with algorithm performance. The feature $B^r = \frac{n_h}{n_l}$ however has a correlation of 0.489 with Kriging performance, the highest correlation of all the features used. This indicates that when choosing whether a low-fidelity source is useful, the analysis should not focus on what this source looks like and how much data it provides. Rather, it should focus on how it behaves *in relation to* the high-fidelity source f_h, and how much data it provides *in relation to* the amount of high-fidelity data.

The correlation of the remaining budget features $B_h = n_h$ and $B_h^r = \frac{n_h}{d}$ also provides a helpful insight. The feature B_h measures how much high-fidelity data is available, but only has a correlation of 0.179 with algorithm performance. The feature B_h^r however measures how much high-fidelity data is available relative to the problem dimension and has a correlation of 0.411, the second highest of all features being analysed. This indicates that studies on Bf-EBB problems should focus on sample budgets relative to problem dimension, something that intuitively makes sense and a large section of the literature is already doing. In fact, as shown in Fig. 7, despite the feature B_h^r not having been chosen to generate the instance space, analysing its value trend sheds light on the existence of regions on the top of the space where both Kriging and Co-Kriging perform well. These regions can be explained by the fact that when a lot of high-fidelity data is available, very accurate Kriging and Co-Kriging models can be trained and thus either technique is a good choice.

Fig. 8 illustrates the trends of the selected features within the space. Most features characterise the relationship between f_h and f_l. Recall the LCC^r features measure the local correlation characteristics of f_l relative to f_h. The feature $LCC_{sd}^{0.2}$ measures the standard deviation of this local correlation using a relative radius of 0.2. The features $LCC_{0.5}^{0.2^{1/d}}$ and $LCC_{0.9}^{0.2^{1/d}}$ measure the probability that f_l and f_h have a local correlation of 0.5 and 0.9, respectively, using a relative radius of $0.2^{1/d}$. The feature $RRMSE$ measures the error between f_h and f_l. The features $NBC_{fitcorr}$ and $\bar{R}_{LI}^2$ are used to characterise the difference $f_h - f_l$ between sources. The feature $\bar{R}_{LI}^2$ measures how easy it is to model this difference using a linear model with interactions, with higher feature values indicating higher modelling accuracy. The feature $NBC_{fitcorr}$ measures the complexity of this difference, with a low feature value indicating a highly multimodal landscape, and a high value indicating a landscape with very few peaks. Finally, only two features characterise the landscape of f_h. The feature $MMCE_{lda}^{0.5}$ measures the error when separating the samples with the top and bottom 50% f_l values using a linear model. The feature M_0 measures the ruggedness of f_h. It is worth stressing once again that the constructed space is by no means unique, especially since features other than these 9 passed the minimum correlation threshold of 0.3 with algorithm performance. Careful consideration of the created space however leads to good insights into the characterisation of a harmful low-fidelity source.

It appears overlaying the local correlation features $LCC_{0.5}^{0.2^{1/d}}$ and $LCC_{0.9}^{0.2^{1/d}}$ divides the space into a right, middle and left region. In the right region, $LCC_{0.9}^{0.2^{1/d}}$ values close to 1 indicate that f_l has a very high probability of being very accurate locally (i.e. having a correlation of at least 0.9 locally). This corresponds to a region where Co-Kriging can be used, and therefore the low-fidelity source is valuable. In the left region, $LCC_{0.5}^{0.2^{1/d}}$ values close to 0 indicate that f_l has a very low probability of being somewhat accurate locally (i.e. having a correlation of at least 0.5 locally). This corresponds to a region where Co-Kriging should not be used, and therefore the low-fidelity source is harmful. The middle region corresponds to instances that are often somewhat locally accurate, but rarely very locally accurate. For these instances, Co-Kriging should only be used if little high-fidelity data is available. It is worth noting that only LCC^r features calculated with a relative radius of $0.2^{1/d}$ seem to help predict algorithm performance. Despite the recommendation

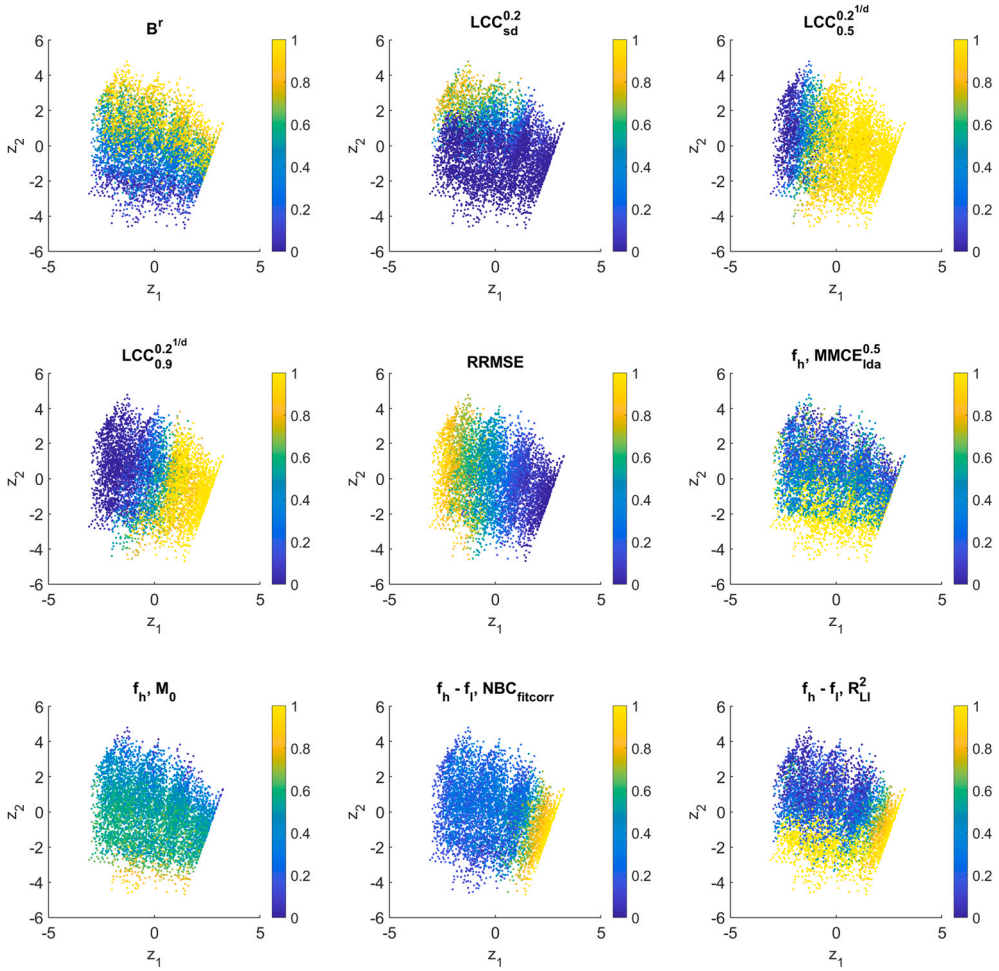


Fig. 8. Feature distributions of the 9 chosen features. Note that the colour gradient is relative; for the range of each of the feature values consult Table 1.

by Andrés-Thió et al. [4] to set $r = 0.2$, in the applied setting using this relative radius leads to small neighbourhoods which contain very few sample points, and lead to unchanging local correlation values. The $LCC^{0.2}_{sd}$ plot in Fig. 8 illustrates this, as for most of the space the calculated standard deviation of the local correlation is 0. Therefore, in an applied setting a relative radius of $0.2^{1/d}$ which grows with the problem dimension should be used when calculating these features.

The pair of features $\bar{R}^2_{LI}$ and $NBC_{fitcorr}$ are also helpful in predicting when a low-fidelity source can be used. When the value of these features is high, this indicates the difference between the two sources is relatively simple, either because it is easy to model or because it has a low number of peaks. In these cases, Co-Kriging should be used. Intuitively this makes sense, as training a Co-Kriging model can be seen as the combination of training a model of f_l, and training a model of the difference $f_h - f_l$. Therefore, if most of the complex behaviour of f_h is represented by f_l (and therefore $f_h - f_l$ is simple), f_l can be very beneficial. It is important to note that perhaps the feature $\bar{R}^2_{LI}$ captures this logic best, as it is possible to imagine a simultaneously complex and unimodal function which is hard to model whilst having a high $NBC_{fitcorr}$ value. Therefore it is likely that $\bar{R}^2_{LI}$ is the better choice of the two for an indication of which model to choose. It is also worth noting that for very small sample sizes a linear model with interactions can fit the sample perfectly. This leads to cases in the bottom right of the space where the $\bar{R}^2_{LI}$ value is 1, but Co-Kriging performs worse than Kriging. Comparing the feature value of $\bar{R}^2_{LI}$ with other features will therefore lead to a good indication of when f_l can be relied upon. Finally, note that a low $\bar{R}^2_{LI}$ or $NBC_{fitcorr}$ value does not necessarily mean Co-Kriging cannot be used.

Not all selected features are necessarily helpful. The landscape features $MMCE^{0.5}_{lda}$ and M_0, in particular, appear to be strongly affected by the sample size used to calculate them, as their trend is very similar to the trend of the budget feature B^r_h. It is therefore not recommended to use these two features when predicting Kriging and Co-Kriging performance. Furthermore, despite the error feature $RRMSE$ seemingly giving a nice separation of the space, this is the only selected feature to be unbounded and therefore the only selected feature to have been normalised. It is hard to predict the impact of this feature value for instances with larger $RRMSE$ than the benchmark suite being analysed. As such, it is also not recommended to use this feature when predicting algorithm performance.

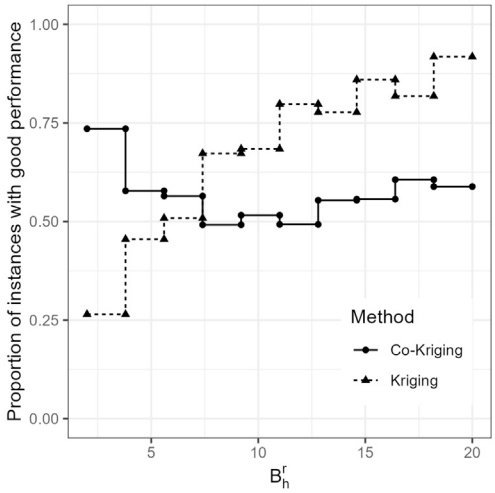


Fig. 9. Proportion of instances for which Kriging and Co-Kriging perform well for different B^r_h values, the amount of high-fidelity data relative to problem dimension.

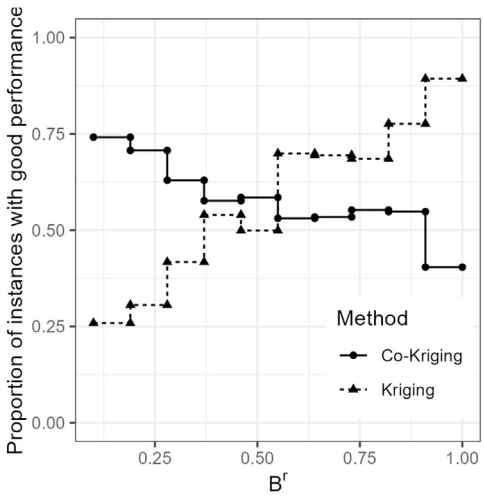


Fig. 10. Proportion of instances for which Kriging and Co-Kriging perform well for different B^r values, the amount of high-fidelity data available relative to low-fidelity data available.

3.4. Proposal of new guidelines

The trained SVMs of the previous subsection provide predictions of regions where Kriging and Co-Kriging should be used based on the 2-dimensional projection. This has the advantage of being intuitive for users to identify the driving features in algorithm performance. It can also be beneficial however to analyse how single feature values impact model performance in order to derive some simple rules that industrial practitioners can follow.

In order to generate these guidelines, the proportion of instances with good Kriging/Co-Kriging performance is plotted against a variety of feature value bins. In each of the plots, the lowest and highest x-axis value of a flat segment represents the range of a bin, and the y-axis value represents the proportion of instances within that range which are labelled as good for a particular model. The first such graph is shown in Fig. 9. This graph indicates that the amount of high-fidelity data available alone is not a very strong indicator of whether Co-Kriging should be used. It does however show that when a lot of high-fidelity data is available, it is highly likely that the best decision is to not consider other data sources when constructing a model. When the relative amount of high-fidelity data available is 20 times the dimension of the problem, Kriging is a good choice for 91.8% of the instances. It is likely that when even more high-fidelity data is available such as $50d$ function evaluations, Kriging will almost always dominate Co-Kriging. Further analysis of harmful data sources should therefore be restricted to high-fidelity budgets within this range. Fig. 10 indicates that in order to rely on a low-fidelity source, it is important to have a lot more low-fidelity data than high-fidelity data. In particular, training a Co-Kriging model when $n_l = n_h$ is to be avoided, as in this special case Kriging has good performance for 89.3% of the instances. Based on this graph, it is recommended to have at least 1.5 times as much low-fidelity as high-fidelity data when using Co-Kriging.

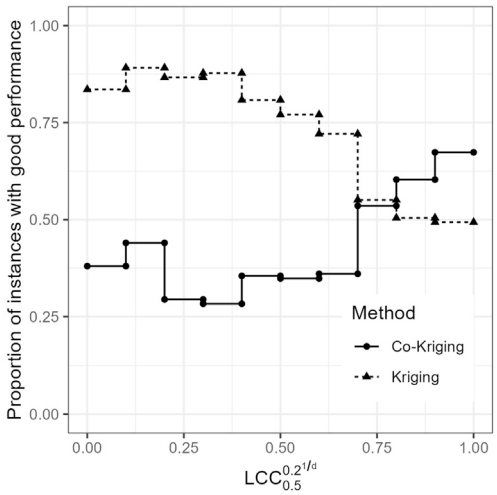


Fig. 11. Proportion of instances for which Kriging and Co-Kriging perform well for different $LCC_{0.5}^{0.2^{1/d}}$ values.

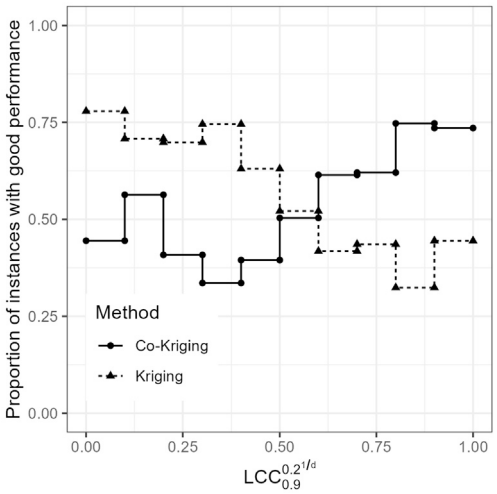


Fig. 12. Proportion of instances for which Kriging and Co-Kriging perform well for different $LCC_{0.9}^{0.2^{1/d}}$ values.

Figs. 11, 12 and 13 provide good cut-off points for three feature values when choosing which model to use. In particular, Fig. 11 shows that when $LCC_{0.5}^{0.2^{1/d}} \leq 0.7$, Kriging performs well for at least 72% of the instances in each of the feature bins. Similarly, Fig. 12 shows that when $LCC_{0.9}^{0.2^{1/d}} \geq 0.8$, Co-Kriging performs well for at least 74% of the instances in each of the feature bins. Finally, Fig. 13 shows that when $\bar{R}^2_{LI} \geq 0.6$, Co-Kriging also performs well for at least 71% of the instances in each of the feature bins.

Before combining these insights into simple rules to decide which model to use, it is worth also looking at the feature CC. This feature measures the overall correlation between f_h and f_l, giving a sense of the global quality of f_l. It was shown by Toal [48] to provide a good indication of when f_l can be relied upon, and is often used in the literature when assessing low fidelity sources. Despite not having been chosen to generate the instance space, Fig. 14 shows that it is better to construct Co-Kriging models when $CC \geq 0.7$, although doing so only when $CC \geq 0.9$ as Toal recommends is a more conservative estimate. Relying on the basic rule of constructing Co-Kriging models when $CC \geq 0.7$ and Kriging models otherwise leads to choosing the right model 71.1% of the time. Whilst this accuracy is not as large as that of the selector trained in the previous subsection, it is a marked improvement over always choosing the same model regardless of instance characteristics. This can be considered the performance of current literature guidelines.

Moving now to the proposal of new basic guidelines based on the insights of this section, the following rules are proposed when choosing whether to construct a Kriging or a Co-Kriging model:

1. For instances where either $B_h^r \geq 20$, $B^r \geq 1$ or $LCC_{0.5}^{0.2^{1/d}} \leq 0.7$, Kriging should be used.
2. For instances which do not satisfy the above conditions, but for which $LCC_{0.9}^{0.2^{1/d}} \geq 0.8$, or $\bar{R}^2_{LI} \geq 0.6$, Co-Kriging should be used.

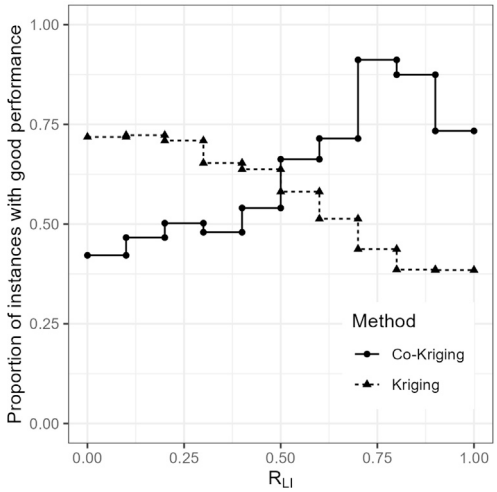


Fig. 13. Proportion of instances for which Kriging and Co-Kriging perform well for different $\tilde{R}^2_{LI}$.

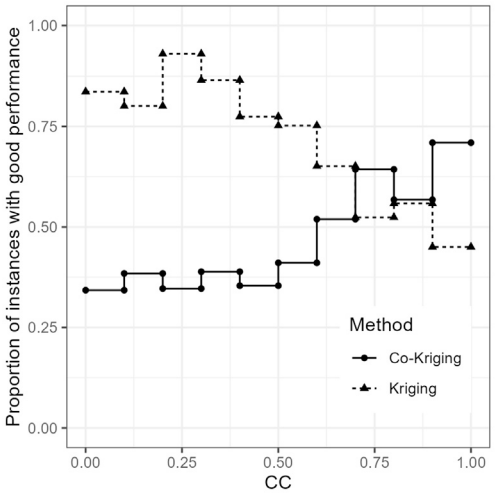


Fig. 14. Proportion of instances for which Kriging and Co-Kriging perform well for different CC values, the global correlation between f_l and f_h.

3. For instances which are not covered by the previous two sets of rules, Co-Kriging should be used when $B^r_h \leq 6$, and Kriging should be used otherwise.

These three sets of rules can be reduced to the following: Kriging should be used when plenty of data is available, the same amount or less of f_l data is available compared to f_h data, or if f_l appears to be clearly harmful. If this is not the case, and f_l appears to be clearly beneficial, Co-Kriging should be used. In the remainder of the cases, that is when f_l is somewhat helpful, Co-Kriging should be used only if limited high-fidelity data is available.

Applying these guidelines to the filtered benchmark set containing 6393 instances yields high accuracies. In particular, 2612 of those instances fall in the category defined by the first set of rules. For these instances, Kriging is recommended and is indeed labelled as having good performance 84.0% of the time. A total of 2564 instances fall in the category defined by the second set of rules. For this second set of instances, Co-Kriging is recommended, and it is labelled as a good algorithm 79.6% of the time. This indicates that the first two sets of rules not only cover a very large set of instances, but also provide a recommendation which is highly accurate, particularly for instances for which Kriging is recommended. The accuracy of applying the full set of rules to the entire filtered benchmark set leads to a good algorithm being chosen 79.3% of the time, which is an improvement over the already accurate selector of the previous subsection. To see why this is the case, it is worth plotting the algorithm prediction within the space, which is shown in Fig. 15. It can be seen here that following these sets of rules leads to a prediction which contains a fair amount of “confetti”, something the selector trained in the previous section is not allowed to do. The reason behind this is that despite the rules leading to a higher accuracy, they do not provide the powerful intuitive insights inherent to ISA. When applying Kriging and Co-Kriging to industrial problems however, these sets of rules can provide an accurate suggestion of the best model to use based on the limited information available.

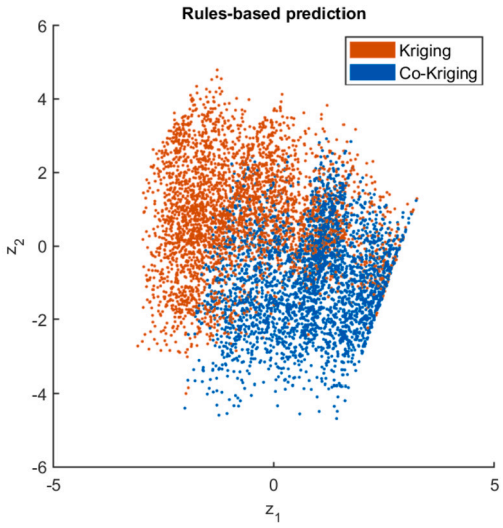


Fig. 15. Selector which predicts whether Kriging or Co-Kriging should be used based on the simple rules given in this section. The red dots indicate instances for which Kriging is predicted to be best, and the blue dots indicate instances for which Co-Kriging is predicted to be best. Note that contrary to the predictor shown in Fig. 6, no clear regions arise from this prediction.

4. Conclusion

This study has characterised beneficial and harmful low-fidelity sources when modelling an expensive black-box f_h. Previous studies that have conducted similar work have done so under perfect conditions, in the sense that the analysis conducted relied on a large amount of data that is unavailable in practice. In this study, this characterisation is constructed using only the limited information available to train a model, meaning both the generated simple guidelines and the more intricate instance space can be directly applied to industrial problems. Despite relying only on limited information, however, the trained predictor still achieves 78.7% accuracy when predicting which model will be most accurate. This is a marked improvement over choosing to always use either Kriging or Co-Kriging, as both options perform well less than 60% of the time. This is particularly remarkable as the predictor is trained on a 2-dimensional projection of a 9-dimensional space that, despite leading to more intuitive predictions, is often at the cost of a loss in prediction accuracy. Using the insights of ISA and departing from the restriction of this 2-dimensional projection, a set of simple guidelines have been developed which show a further improvement on model prediction. Despite being based on easy to calculate features approximated with limited data, these rules lead to a good model being chosen 79.3% of the time, a big improvement over existing guidelines.

The characterisation of a harmful source has been achieved by comparing the performance of (single-source) Kriging models with (two-source) Co-Kriging models. The widespread usage of these models and the fact that many new techniques are either an extension or rely on a similar framework implies the findings of this work are very likely applicable to other models in the literature. The derived analysis presented here therefore supplies another important step in the ongoing debate within the literature around the validity of current benchmarks and guidelines used in the field. As stated often in the text, ISA is iterative in nature. Further instances and features can be developed however, as well as the analysis extended to other models, to further the understanding of how low-fidelity sources can be exploited.

Finally, the work presented here has focused on the simplest variant of Bf-EBB problems, namely the case where the data has already been gathered. In this case, the only decision available to the practitioner is which model to train. In many cases, a total budget is supplied instead. Further research will shift its attention to this more complex scenario, where the user needs to decide how to split the budget between sources, and where to gather further samples in the space. Future work will use the findings of this work to develop adaptive techniques that dynamically choose when to rely on low-fidelity sources. The development of these techniques will focus both on optimisation, and on the case where the sole aim is to construct as accurate a model as possible. We believe the findings of this work will allow newly developed techniques to outperform existing methods that choose to always or never use a source.

Replication of results

The code implemented for this study is divided in two modular repositories. The first repository [2] implements all of the referenced function pairs, provides the code used to measure the feature values, and generates the benchmark suite with 321 instances, as well as additional benchmark suites of sizes 65, 100, 140, 206 and 509. Researchers wanting to assess the performance of other techniques on these benchmarks are directed to this repository. The second repository [3] implements both Kriging and Co-Kriging, and the framework to analyse their performance on the selected benchmark suite. Researchers wanting to rerun these experiments, or to use

the implementation of Kriging and Co-Kriging (as well as other techniques) are directed to this repository. All relevant metadata (i.e. features and algorithm performance) is also made available in the relevant repositories for further analysis if desired.

Funding

This research was supported by the Australian Research Council under grant number IC200100009 for the ARC Training Centre in Optimisation Technologies, Integrated Methodologies and Applications (OPTIMA). The first author is also supported by a Research Training Program Scholarship from The University of Melbourne.

CRediT authorship contribution statement

Nicolau Andrés-Thió: Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Mario Andrés Muñoz:** Writing – review & editing, Supervision, Conceptualization. **Kate Smith-Miles:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative.

Appendix A. Feature definitions

The work of Toal [48] proposes two features which measure the global quality of f_l relative to f_h, namely the Correlation Coefficient (CC) and the Root Mean Squared Error ($RMSE$). Both features are calculated on a given set of sample points $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \Omega$. The objective function values of both f_h and f_l are used to calculate these features, namely the sets $\mathbf{y}_l = \{f_l(\mathbf{x}_1), \dots, f_l(\mathbf{x}_n)\}$ and $\mathbf{y}_h = \{f_h(\mathbf{x}_1), \dots, f_h(\mathbf{x}_n)\}$. The definition of both features is the following

$$RMSE = \left[\frac{\sum_{i=1}^{n}(f_l(\mathbf{x}_i) - f_h(\mathbf{x}_i))^2}{n}\right]^{1/2}$$

$$CC = \left[\frac{1}{n-1}\left(\frac{\sum_{i=1}^{n}(f_l(\mathbf{x}_i) - \bar{y}_l)(f_h(\mathbf{x}_i) - \bar{y}_h)}{s_{y_l} s_{y_h}}\right)\right]^2$$

$$\bar{y}_l = \frac{1}{n}\sum_{i=1}^{n} f_l(\mathbf{x}_i)$$

$$s_{y_l} = \left[\frac{\sum_{i=1}^{n}(f_l(\mathbf{x}_i) - \bar{y}_l)^2}{n-1}\right]^{1/2}$$

$$\bar{y}_h = \frac{1}{n}\sum_{i=1}^{n} f_h(\mathbf{x}_i)$$

$$s_{y_h} = \left[\frac{\sum_{i=1}^{n}(f_h(\mathbf{x}_i) - \bar{y}_h)^2}{n-1}\right]^{1/2}$$

A high CC value indicates that overall, f_l behaves similarly to f_h. A low $RMSE$ value indicates that overall, the two sources differ little in the space. The work of Andrés-Thió et al. [4] proposes a scaling of the $RMSE$ feature in order to make it comparable between pairs of functions. This scaled feature is denoted Relative $RMSE$ ($RRMSE$) and is given by

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

The same authors propose a set of features which assess the local correlation characteristics of f_l and f_h. First the definition of Weighted Correlation Coefficient ($WCC(\mathbf{w})$) is given for a set of weights $\mathbf{w} = \{w_1, \dots, w_n\}$

$$WCC(\mathbf{w}) = \left[\frac{1}{\sum_{i=1}^{n} w_i}\left(\frac{S}{s_{y_l} s_{y_h}}\right)\right]^2$$

$$S = \sum_{i=1}^{n} w_i (f_l(\mathbf{x}_i) - \bar{y}_l)(f_h(\mathbf{x}_i) - \bar{y}_h)$$

$$\bar{y}_l = \frac{\sum_{i=1}^{n} w_i f_l(\mathbf{x}_i)}{\sum_{i=1}^{n} w_i}$$

$$s_{y_l} = \left[\frac{\sum_{i=1}^{n} w_i (f_l(\mathbf{x}_i) - \bar{y}_l)^2}{\sum_{i=1}^{n} w_i}\right]^{1/2}$$

$$\bar{y}_h = \frac{\sum_{i=1}^{n} w_i f_h(\mathbf{x}_i)}{\sum_{i=1}^{n} w_i}$$

$$s_{y_h} = \left[\frac{\sum_{i=1}^{n} w_i (f_h(\mathbf{x}_i) - \bar{y}_h)^2}{\sum_{i=1}^{n} w_i}\right]^{1/2}$$

Next the Local Correlation Coefficient at a point $\mathbf{x}$ with radius r is defined as

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

$$\text{where} \qquad w_i = \min\{0, 1 - \frac{\|\mathbf{x} - \mathbf{x}_i\|}{r\|\mathbf{x}^\top - \mathbf{x}^\perp\|}\}$$

It is recommended to scale the data to lie inside the unit hypercube $[0,1]^d$ before calculating this feature. This measure calculates the correlation between f_l and f_h inside the d-sphere centred at a point $\mathbf{x}$ with radius $r\|\mathbf{x}^\top - \mathbf{z}\mathbf{x}^\perp\|$. The weights lead to a higher impact of the correlation of points closer to the centre. Now define the sets $\mathcal{L}^r = \{LCC^r(\mathbf{x}_1), \ldots, LCC^r(\mathbf{x}_n)\}$ and $\mathcal{L}^r_p = \{LCC^r(\mathbf{x}) \in \mathcal{L}^r \mid LCC^r(\mathbf{x}) \geq p\}$, which are used to define the features

$$LCC^r_p = \frac{|\mathcal{L}^r_p|}{|\mathcal{L}^r|}$$

$$LCC^r_{mean} = \frac{1}{n}\sum_{i=1}^{n} LCC^r(\mathbf{x}_i)$$

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

$$LCC^r_{coeff} = \frac{LCC^r_{sd}}{LCC^r_{mean}}$$

The LCC^r_p calculate the probability that f_l has a local correlation of at least p with f_h, whereas the features LCC^r_{mean}, LCC^r_{sd} and LCC^r_{coeff} calculate various distribution measures of this local correlation.

References

[1] H. Alipour, M.A. Muñoz, K. Smith-Miles, Enhanced instance space analysis for the maximum flow problem, Eur. J. Oper. Res. 304 (2) (2023) 411–428.

[2] N. Andrés-Thió, Bifidelity Surrogate Modelling benchmark problems, https://doi.org/10.5281/zenodo.8353690, available for download at https://github.com/nandresthio/bifiEBBbenchmarks, 2023.

[3] N. Andrés-Thió, Bifidelity Surrogate Modelling methods, https://doi.org/10.5281/zenodo.8353700, available for download at https://github.com/nandresthio/bifiEBBmethods, 2023.

[4] N. Andrés-Thió, M.A. Muñoz, K. Smith-Miles, Bifidelity surrogate modelling: showcasing the need for new test instances, INFORMS J. Comput. 34 (6) (2022) 3007–3022.

[5] B. Bischl, P. Kerschke, L. Kotthoff, M. Lindauer, Y. Malitsky, A. Fréchette, H. Hoos, F. Hutter, K. Leyton-Brown, K. Tierney, et al., Aslib: a benchmark library for algorithm selection, Artif. Intell. 237 (2016) 41–58.

[6] M. Cheng, P. Jiang, J. Hu, L. Shu, Q. Zhou, A multi-fidelity surrogate modeling method based on variance-weighted sum for the fusion of multiple non-hierarchical low-fidelity data, Struct. Multidiscip. Optim. 64 (2021) 3797–3818.

[7] H. Dong, B. Song, P. Wang, S. Huang, Multi-fidelity information fusion based on prediction of kriging, Struct. Multidiscip. Optim. 51 (6) (2015) 1267–1280.

[8] J.T. Eweis-Labolle, N. Oune, R. Bostanabad, Data fusion with latent map Gaussian processes, J. Mech. Des. 144 (9) (2022) 091703.

[9] M.G. Fernández-Godino, C. Park, N.H. Kim, R.T. Haftka, Issues in deciding whether to use multifidelity surrogates, AIAA J. 57 (5) (2019) 2039–2054.

[10] A.I. Forrester, A. Sóbester, A.J. Keane, Multi-fidelity optimization via surrogate modelling, Proc. R. Soc. A, Math. Phys. Eng. Sci. 463 (2088) (2007) 3251–3269.

[11] Z.Z. Foumani, M. Shishehbor, A. Yousefpour, R. Bostanabad, Multi-fidelity cost-aware Bayesian optimization, Comput. Methods Appl. Mech. Eng. 407 (2023) 115937.

[12] M.L. Garneau, Modelling of a solar thermal power plant for benchmarking blackbox optimization solvers, Master's thesis, Polytechnique Montréal https://publications.polymtl.ca/1996/, text available at https://publications.polymtl.ca/1996, code available at https://github.com/bbopt/solar, 2015.

[13] Z.-H. Han, S. Görtz, Hierarchical kriging model for variable-fidelity surrogate modeling, AIAA J. 50 (9) (2012) 1885–1896.

[14] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, D. Brockhoff, COCO: a platform for comparing continuous optimizers in a black-box setting, Optim. Methods Softw. 36 (2021), https://doi.org/10.1080/10556788.2020.1808977.

[15] C. Huang, W.E. Anderson, C.L. Merkle, V. Sankaran, Multifidelity framework for modeling combustion dynamics, AIAA J. 57 (5) (2019) 2055–2068.
[16] D.R. Jones, A taxonomy of global optimization methods based on response surfaces, J. Glob. Optim. 21 (4) (2001) 345–383.
[17] M.C. Kennedy, A. O'Hagan, Predicting the output from a complex computer code when fast approximations are available, Biometrika 87 (1) (2000) 1–13.
[18] P. Kerschke, H. Trautmann, Comprehensive feature-based landscape analysis of continuous and constrained optimization problems using the R-package flacco, in: N. Bauer, K. Ickstadt, K. Lübke, G. Szepannek, H. Trautmann, M. Vichi (Eds.), Applications in Statistical Computing – From Music Data Analysis to Industrial Quality Improvement, Studies in Classification, Data Analysis, and Knowledge Organization, Springer, 2019, pp. 93–123.
[19] D. Khatamsaz, A. Molkeri, R. Couperthwaite, J. James, R. Arróyave, A. Srivastava, D. Allaire, Adaptive active subspace-based efficient multifidelity materials design, Mater. Des. 209 (2021) 110001.
[20] D.G. Krige, A statistical approach to some basic mine valuation problems on the Witwatersrand, J. S. Afr. Inst. Min. Metall. 52 (6) (1951) 119–139.
[21] M. Lindauer, J.N. van Rijn, L. Kotthoff, The algorithm selection competitions 2015 and 2017, Artif. Intell. 272 (2019) 86–100.
[22] B. Liu, S. Koziel, Q. Zhang, A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems, J. Comput. Sci. 12 (2016) 28–37.
[23] H. Liu, Y.-S. Ong, J. Cai, Y. Wang, Cope with diverse data structures in multi-fidelity modeling: a Gaussian process method, Eng. Appl. Artif. Intell. 67 (2018) 211–225.
[24] M. Lunacek, D. Whitley, The dispersion metric and the CMA evolution strategy, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, 2006, pp. 477–484.
[25] L. Lv, C. Zong, C. Zhang, X. Song, W. Sun, Multi-fidelity surrogate model based on canonical correlation analysis and least squares, J. Mech. Des. 143 (2) (2021) 021705.
[26] A. March, K. Willcox, Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives, AIAA J. 50 (5) (2012) 1079–1089.
[27] G. Matheron, Principles of geostatistics, Econ. Geol. 58 (8) (1963) 1246–1266.
[28] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, G. Rudolph, Exploratory landscape analysis, in: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, 2011, pp. 829–836.
[29] M. Mısır, M. Sebag, Alors: an algorithm recommender system, Artif. Intell. 244 (2017) 291–314.
[30] M.A. Muñoz, M. Kirley, S.K. Halgamuge, Exploratory landscape analysis of continuous space optimization problems using information content, IEEE Trans. Evol. Comput. 19 (1) (2014) 74–87.
[31] M.A. Muñoz, L. Villanova, D. Baatar, K. Smith-Miles, Instance spaces for machine learning classification, Mach. Learn. 107 (2018) 109–147.
[32] M.A. Muñoz, K. Smith-Miles, Instance Space Analysis: a toolkit for the assessment of algorithmic power, source code is available at https://github.com/andremun/InstanceSpace, 2023.
[33] C. Park, R.T. Haftka, N.H. Kim, Remarks on multi-fidelity surrogates, Struct. Multidiscip. Optim. 55 (3) (2017) 1029–1050.
[34] C. Park, R.T. Haftka, N.H. Kim, Low-fidelity scale factor improves Bayesian multi-fidelity prediction by reducing bumpiness of discrepancy function, Struct. Multidiscip. Optim. 58 (2018) 399–414.
[35] X. Peng, J. Kou, W. Zhang, Multi-fidelity nonlinear unsteady aerodynamic modeling and uncertainty estimation based on hierarchical kriging, Appl. Math. Model. 122 (2023) 1–21.
[36] M. Preuss, Improved topological niching for real-valued global optimization, in: Applications of Evolutionary Computation: EvoApplications 2012: EvoCOMNET, EvoCOMPLEX, EvoFIN, EvoGAMES, EvoHOT, EvoIASP, EvoNUM, EvoPAR, EvoRISK, EvoSTIM, and EvoSTOC, Málaga, Spain, April 11-13, 2012, Proceedings, Springer, 2012, pp. 386–395.
[37] S. Prigent, X. Descombes, D. Zugaj, L. Petit, A.-S. Dugaret, P. Martel, J. Zerubia, Skin lesion evaluation from multispectral images, Ph.D. thesis, INRIA, 2012.
[38] D. Rajnarayan, A. Haas, I. Kroo, A multifidelity gradient-free optimization method and application to aerodynamic design, in: 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2008, p. 6020.
[39] J.R. Rice, The Algorithm Selection Problem, Advances in Computers, vol. 15, Elsevier, 1976, pp. 65–118.
[40] M. Shi, L. Lv, W. Sun, X. Song, A multi-fidelity surrogate model based on support vector regression, Struct. Multidiscip. Optim. (2020) 1–13.
[41] L. Shu, P. Jiang, X. Song, Q. Zhou, Novel approach for selecting low-fidelity scale factor in multifidelity metamodeling, AIAA J. 57 (12) (2019) 5320–5330.
[42] K. Smith-Miles, M.A. Muñoz, Instance space analysis for algorithm testing: methodology and software tools, ACM Comput. Surv. 55 (12) (2023) 1–31.
[43] K. Smith-Miles, M.A. Muñoz, N. Neelofar, Melbourne Algorithm Test Instance Library with Data Analytics (MATILDA), source code is available at https://matilda.unimelb.edu.au/matilda/, 2023.
[44] I.M. Sobol', On the distribution of points in a cube and the approximate evaluation of integrals, Zh. Vychisl. Mat. Mat. Fiz. 7 (4) (1967) 784–802.
[45] X. Song, L. Lv, W. Sun, J. Zhang, A radial basis function-based multi-fidelity surrogate model: exploring correlation between high-fidelity and low-fidelity models, Struct. Multidiscip. Optim. 60 (3) (2019) 965–981.
[46] S. Surjanovic, D. Bingham, Virtual Library of Simulation Experiments: Test Functions and Datasets, retrieved December 14, 2020, from http://www.sfu.ca/~ssurjano, 2020.
[47] A. Thenon, V. Gervais, M. Le Ravalec, Sequential design strategy for kriging and cokriging-based machine learning in the context of reservoir history-matching, Comput. Geosci. 26 (5) (2022) 1101–1118.
[48] D.J. Toal, Some considerations regarding the use of multi-fidelity Kriging in the construction of surrogate models, Struct. Multidiscip. Optim. 51 (6) (2015) 1223–1245.
[49] D.J. Toal, Applications of multi-fidelity multi-output Kriging to engineering design optimization, Struct. Multidiscip. Optim. 66 (6) (2023) 125.
[50] F. Wilcoxon, Individual comparisons by ranking methods, in: Breakthroughs in Statistics, Springer, 1992, pp. 196–202.
[51] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[52] J. Wu, S. Toscano-Palmerin, P.I. Frazier, A.G. Wilson, Practical multi-fidelity Bayesian optimization for hyperparameter tuning, in: Uncertainty in Artificial Intelligence, PMLR, 2020, pp. 788–798.
[53] Y. Wu, J. Hu, Q. Zhou, S. Wang, P. Jin, An active learning multi-fidelity metamodeling method based on the bootstrap estimator, Aerosp. Sci. Technol. 106 (2020) 106116.
[54] S. Xiong, P.Z. Qian, C.J. Wu, Sequential design and analysis of high-accuracy and low-accuracy computer codes, Technometrics 55 (1) (2013) 37–46.
[55] H. Yang, S.H. Hong, Y. Wang, A sequential multi-fidelity surrogate-based optimization methodology based on expected improvement reduction, Struct. Multidiscip. Optim. 65 (5) (2022) 153.
[56] Y. Zhao, J. Liu, Z. He, A general multi-fidelity metamodeling framework for models with various output correlation, Struct. Multidiscip. Optim. 66 (5) (2023) 101.