Multi-fidelity Expensive Black-Box Modelling: An examination of trust

by

Nicolau Andrés-Thió

ORCID: 0000-0002-5417-8571

A thesis submitted in total fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Science
School of Mathematics and Statistics
THE UNIVERSITY OF MELBOURNE

June 2024

Abstract

In recent years, methods which tackle Multi-fidelity Expensive Black-Box (Mf-EBB) problems have received increasing attention due to their widespread presence in the industrial setting. Problems of this type are characterised by a lack of an analytical expression defining the relationship between design variables and chosen quality metrics. The key challenge is that knowledge of this relationship is limited to a small set of sample observations of varying quality. In the field of Mf-EBB, a problem instance consists of an expensive yet accurate source of information, and one or more cheap yet less accurate sources of information. The field aims to provide techniques either to accurately explain how decisions affect design outcome, or to find the best decisions to optimise design outcomes.

Surrogate modelling techniques in particular have successfully been applied to Mf-EBB problems in the past. 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 also simultaneously incorporate multiple information sources. Only in recent years, however, have researchers begun to explore the conditions under which these techniques are reliable. The existence of multiple sources in particular poses the question of which ones should be used when constructing a model.

The work of this thesis examines different aspects of Mf-EBB surrogate modelling methods. The main contributions center around the concept of trust on an information source during model construction. Predicting whether incorporating certain datasets will increase or decrease model performance is no trivial matter, as will be shown in this work. Novel test instances which are more varied than those found in the literature are proposed, as well as new features used to determine when the low-fidelity information source should be used. These are used to expose the criticality of a more heterogeneous test suite for algorithm assessment, as results are obtained which contradict existing literature guidelines. Next, the characterisation of harmful low-fidelity sources is approached as an algorithm selection problem. By performing this analysis using only the limited data available to train a surrogate model, new guidelines are provided that can be directly used in an applied industrial setting. Finally, the obtained characterisation is used to create novel techniques for model creation with budget constraints which adaptively select the model to be trained based on the available data. These techniques are shown the be safer than traditional approaches, and highlight the benefits that result from the insights found in this thesis.

Declaration of Authorship

I, Nicolau Andrés-Thió, declare that this thesis titled, 'Multi-fidelity Expensive Black-Box problems: Learning to exploit low-fidelity approximations' and the work presented in it are my own. I confirm that:

- The thesis comprises only my original work towards the Doctor of Philosophy except where indicated in the preface;
- due acknowledgement has been made in the text to all other material used; and
- the thesis is fewer than the maximum word limit in length, exclusive of tables, maps, bibliographies and appendices as approved by the Research Higher Degrees Committee.

Signed: Nicolau Andrés-Thió

Date: 04/06/2024

Preface

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). Funding was also received through the Australian Government Research Training Program Scholarship from The University of Melbourne. Travel funding was received through the travel grant scheme of the University of Melbourne. This research was also supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative.

List of publications

Andrés-Thió, Nicolau, Mario Andrés Muñoz, and Kate Smith-Miles. "Methodology and challenges of surrogate modelling methods for multi-fidelity expensive black-box problems." ANZIAM journal, accepted for publication.

This paper provides the content for most of Chapter 2.

Andrés-Thió, Nicolau, Mario Andrés Muñoz, and Kate Smith-Miles. "Bi-fidelity Surrogate Modelling: Showcasing the Need for New Test Instances." INFORMS Journal on Computing 34.6 (2022): 3007-3022.

This paper provides the content for Chapter 3.

Andrés-Thió, Nicolau, Mario Andrés Muñoz, and Kate Smith-Miles. "Characterising harmful data sources when constructing multi-fidelity surrogate models." Artificial Intelligence, submitted and under review.

This paper provides the content for Chapter 4.

List of software outputs

Andrés-Thió, Nicolau. "Bifidelity Surrogate Modelling benchmark problems", doi:10.5281/zenodo.8353690, available for download at https://github.com/nandresthio/bifiEBBbenchmarks, 2024.

This repository provides the implemented literature and novel benchmarks used throughout the thesis and detailed in Appendix C.

Andrés-Thió, Nicolau. "Bifidelity Surrogate Modelling methods", doi:10.5281/zenodo.8353700, available for download at https://github.com/nandresthio/bifiEBBmethods, 2024.

This repository provides the implemented methods used throughout the thesis and detailed in Appendices A and B.

Acknowledgements

The work presented in this thesis could not have been completed without the help of several people. First and foremost, thank you to my supervisors Prof. Kate Smith-Miles and Dr Mario Andrés Muñoz for your supervision these past three and a half years. I have felt very lucky to work with both of you from the very start of this project. To Kate, I have lost count of the number of times I was convinced I was approaching a problem with the right frame of mind, only to be proven wrong by you after listening to my struggles for only an hour. Your capacity to quickly get to the root of a problem has kept me inspired throughout my thesis. To Andrés, having someone there who remembers the struggles of being a PhD student and was happy to talk through and support me with them was immensely valuable. Thank you also for your help with understanding the inner workings of your implemented work.

To my industrial supervisor Niall O'Shea at Boeing Australia, thank you for making the effort of meeting me halfway in terms of the mathematics behind airplane testing, despite the two of us speaking different (scientific) languages. Thank you also to Dr. Francis Valentinis for accepting to be my industrial supervisor for the second half of my PhD, and for spending the time to help me understand the exact problems that engineers at Boeing Australia are tackling. Thank you as well to Dr Vera Somers; being able to have discussions with someone with a strong background in the two worlds of aerodynamics and mathematics was very helpful.

Thank you also to Prof. Andrea Lodi and Prof. Sébastien Le Digabel for being kind enough to host me in Montreal for two years. Feeling immediately included in the operations research community there made it very easy to work on my PhD remotely.

To the many friends in Montreal and Melbourne, both near and far, Alex, Fede, Will, Flek, Tim, Dom & Dom, Stoj, Rachael, Rob and many others. Completing a PhD without all of you in my life would have been a very lonely journey.

To my parents Alberto and Marta, and my brother Pol, thank you also as always for being the best support group anyone could ever wish for. You're incredible examples of hard work and dedication in as varied walks of life as running a deli, saving babies and exploring the many paths of an artist and dancer. I love you all very much.

Finally, a very special thanks to my life partner Carlie, my girlfriend at the start of this PhD and now my wife at the end of it. It is so reassuring to know that no matter what adventure I am in, from moving countries twice, to organising a wedding, to working on the same crazy maths for almost four years, I am doing it with a teammate by my side who loves me and supports me. Thank you for being in my life.

Contents

Abstract i

Declaration of Authorship ii

Preface iii

Acknowledgements v

List of Figures ix

List of Tables xi

Abbreviations xii

1 Introduction 1
- 1.1 Motivation . . . 2
- 1.2 Aims . . . 3
- 1.3 Thesis outline and contributions . . . 5

2 Background 9
- 2.1 Problem definitions . . . 10
  - 2.1.1 Problem 1: Model creation with fixed sample . . . 11
  - 2.1.2 Problem 2: Model creation with sample budget . . . 13
  - 2.1.3 Problem 3: Function optimisation with sample budget . . . 14
- 2.2 Surrogate modelling techniques . . . 15
  - 2.2.1 Single-source surrogate models . . . 15
    - 2.2.1.1 Polynomial regression . . . 15
    - 2.2.1.2 Radial Basis Functions . . . 16
    - 2.2.1.3 Kriging . . . 17
  - 2.2.2 Multi-source surrogate models . . . 19
    - 2.2.2.1 Co-Kriging . . . 21
  - 2.2.3 Initial design of experiments . . . 23
  - 2.2.4 Acquisition function . . . 26
- 2.3 Illustrative example . . . 30
- 2.4 Experimental studies . . . 37
- 2.5 Conclusion . . . 39

3 Showcasing the need for new test instances **41**
3.1 Existing test suites . . . 42
3.1.1 Existing features . . . 44
3.1.2 Algorithm performance analysis on literature test suite . . . 44
3.2 Creation of new test suites . . . 50
3.2.1 New features . . . 50
3.2.2 Generating new instances . . . 53
3.2.2.1 Height-based disturbance . . . 54
3.2.2.2 Location-based disturbance . . . 55
3.2.3 Augmented test suite construction . . . 56
3.2.4 Updated algorithm performance analysis . . . 58
3.3 Discussion . . . 61
3.3.1 Impact of new features . . . 62
3.3.2 Impact of new instances . . . 63
3.4 Conclusion . . . 64

4 Characterising harmful data sources **66**
4.1 Preliminaries . . . 68
4.1.1 Instance Space Analysis . . . 68
4.1.2 Instances, Features and Algorithms . . . 71
4.1.2.1 Instances . . . 71
4.1.2.2 Algorithms and Performance . . . 72
4.1.2.3 Features . . . 74
4.2 ISA of Bf-EBB surrogate modelling with fixed sample . . . 75
4.2.1 Generating an unbiased test suite . . . 75
4.2.2 Algorithm performance analysis via ISA . . . 78
4.2.3 Analysis of features . . . 83
4.2.4 Proposal of new guidelines . . . 87
4.3 Conclusion . . . 92

5 Adapting to budget constraints **94**
5.1 Surrogate modelling framework . . . 96
5.1.1 Design of Experiments . . . 97
5.1.2 Choice of model . . . 99
5.1.3 Data acquisition . . . 100
5.1.3.1 IMSPE for Kriging . . . 101
5.1.3.2 IMSPE for Co-Kriging . . . 102
5.1.3.3 Acquisition functions . . . 106
5.2 Computational study . . . 107
5.2.1 Experimental setup . . . 107
5.2.2 Results . . . 109
5.3 Conclusion . . . 113

6 Concluding remarks **115**
6.1 Summary of contributions . . . 115
6.2 Future directions . . . 117

A Theoretical derivation of Kriging and Co-Kriging **119**
A.1 Kriging . . . 119
A.2 Co-Kriging . . . 122

B Auxiliary optimisation algorithms **127**
B.1 Accelerated Random Search . . . 128
B.2 Random Heavy Tuning . . . 129

C Implemented literature instances **133**
C.1 Fixed instances . . . 133
C.2 Parameter-based instances . . . 146
C.3 Error-based instances . . . 148
C.4 Disturbance-based instances . . . 150
C.5 SOLAR . . . 156

D Flacco features **157**
D.1 ELA distribution features . . . 157
D.2 ELA levelset prediction . . . 158
D.3 ELA meta modelling . . . 158
D.4 Information content . . . 159
D.5 Nearest better clustering . . . 160
D.6 Principal component analysis . . . 162
D.7 Dispersion . . . 162

E IMSPE derivation **164**
E.1 W functions . . . 164
E.2 Change in Kriging IMSPE . . . 168
E.3 Change in Co-Kriging IMSPE . . . 169

Bibliography **173**

List of Figures

1.1 Thesis structure 8

2.1 A worst case 2-dimensional LHS plan of 25 samples 25
2.2 An initially random then locally optimised LHS plan of 25 samples in 2 dimensions 25
2.3 Procedure for choosing a subset of an LHS plan 26
2.4 Elevation map of Australia 31
2.5 Error based approximation of the elevation map of Australia 32
2.6 Approximation based approximation of the elvation map of Australia . . . 32
2.7 Locations at which elevation samples have been gathered 33
2.8 Average error of three surrogate models trying to model the elevation of Australia 34
2.9 Average best point found during the optimisation of three surrogate modelling techniques 35

3.1 Plot of literature test suite instances using CC and RRMSE feature values 45
3.2 Comparative performance of Kriging and Co-Kriging on the literature test suite, plotted with respect to CC and $RRMSE$ 47
3.3 Decision tree constructed using the literature features and the performance of Co-Kriging and Kriging on the literature test suite 48
3.4 Average feature importance of 100 decision trees using literature features and instances 49
3.5 Plot of literature test suite instances using CC and a newly defined feature 52
3.6 Comparative performance of Kriging and Co-Kriging on the literature test suite, plotted with respect to CC and a newly defined feature 53
3.7 Basic disturbance added to a high-fidelity function 54
3.8 Height based disturbances added to a high-fidelity function 55
3.9 Location based disturbances added to a high-fidelity function 57
3.10 Newly generated instances and augmented test suite, plotted with respect to CC and a newly defined feature 58
3.11 Comparative performance of Kriging and Co-Kriging on the augmented test suite 59
3.12 Decision tree constructed using new and literature features and the performance of Co-Kriging and Kriging on the augmented test suite 61
3.13 Average feature importance of 100 decision trees using new and literature features and instances 61

4.1 ISA framework 69
4.2 Sources of the function pairs used for ISA 80

4.3 Binary performance of Kriging and Co-Kriging models 81
4.4 SVM predictions of Kriging and Co-Kriging performance 81
4.5 SVM selector which predicts whether Kriging or Co-Kriging should be used 82
4.6 Feature distribution of the budget feature B_h^r 84
4.7 Feature distributions of the 9 chosen features for ISA 85
4.8 Proportion of instances for which Kriging and Co-Kriging perform well for different B_h^r values 87
4.9 Proportion of instances for which Kriging and Co-Kriging perform well for different B^r values 87
4.10 Proportion of instances for which Kriging and Co-Kriging perform well for different $LCC_{0.4}^{0.2^{1/d}}$ values 89
4.11 Proportion of instances for which Kriging and Co-Kriging perform well for different $LCC_{0.95}^{0.2^{1/d}}$ values 89
4.12 Proportion of instances for which Kriging and Co-Kriging perform well for different $\bar{R}_L^2$ 89
4.13 Proportion of instances for which Kriging and Co-Kriging perform well for different CC values 90
4.14 Selector which predicts whether Kriging or Co-Kriging should be used based on the new industrial guidelines 91

5.1 Surrogate modelling framework 97
5.2 Correlation of constructed models for varying cost ratios and relative budgets 110
5.3 Error of constructed models for varying cost ratios and relative budgets . 111

List of Tables

2.1 Error of constructed models given a fixed sample 34

3.1 Accuracies of different techniques predicting when to use Co-Kriging on the literature test suite 48
3.2 Accuracies of different techniques predicting when to use Co-Kriging on the literature and augmented test suites 59

4.1 Short description of the features used for ISA 76
4.2 Accuracy of the constructed SVMs which predict algorithm performance . 82

Abbreviations

EBB	**E**xpensive **B**lack-**B**ox
Mf-EBB	**M**ulti-**f**idelity **EBB**
Bf-EBB	**B**i-**f**idelity **EBB**
WT	**W**ind **T**unnel
CFD	**C**omputational **F**luid **D**ynamics
MLE	**M**aximum **L**ikelihood **E**stimator
LHS	**L**atin **H**ypercube **S**ampling
CC	**C**orrelation **C**oefficient
RMSE	**R**oot **M**ean **S**quared **E**rror
RRMSE	**R**elative **RMSE**
WCC	**W**eighted **CC**
LCC	**L**ocal **CC**
ISA	**I**nstance **S**pace **A**nalysis
ELA	**E**xploratory **L**andscape **A**nalysis
IC	**I**nformation **C**ontent
SVM	**S**upport **V**ector **M**achine
IMSPE	**I**ntegrated **M**ean **S**quare **P**rediction **E**rror
DoE	**D**esign **o**f **E**xperiments

Chapter 1

Introduction

Industrial design problems are often centred around the performance analysis of a new design under a variety of use scenarios, or the search for design parameters that optimise the performance of a new product. These can be seen for example when characterising the lift and drag of an aircraft under a wide range of flight conditions [1] or when optimising the design of a ship's hull form to minimise its water drag [2]. In both cases, a set of inputs (i.e. flight conditions or a hull's length, draught, etc...) affect the investigated output (i.e. aircraft lift and drag or a hull's water drag), but often no analytical expression of this relationship is known. Problems of this type are known as *black-box* problems, where the only way to evaluate the outcome for a new decision point is through the use of some predetermined procedure. Carrying out these procedures is often *expensive*, meaning that sampling carries a high cost, measured in terms that are either computational, monetary or temporal. Problems of this type are known as Expensive Black-Box (EBB). Algorithms developed for these problems often aim to better understand the relationship between the design variables and outcome to accurately predict the outcome value in regions not yet sampled.

Surrogate modeling methods [3] have long been applied to EBB problems to surmount the limitations of working with very little available data. These techniques rely on constructing an accurate model of the black-box to help guide further sampling, whether to explore the input space to further learn the input-output relationship, or to optimise a design by exploring unknown regions and exploiting promising ones. Furthermore, a key assumption with EBB problems is that the cost to evaluate a sample (i.e. obtain an

output of the black-box for a given input) dominates almost any other computational cost. Since the time taken to train a model and choose further sample sites is assumed to be negligible, and the only significant computational cost is the sample evaluation, the ability to construct highly accurate models depends on the budget available for sample evaluation.

It is often the case when dealing with a high-cost, highly accurate black-box that other sources of information are available with varying degrees of cost and accuracy. A survey by Fernández-Godino et al. [4] shows many potential reasons for different levels of cost and accuracy, including the simplification of a mathematical model, an increase in model coarseness, and the difference between simulations and experiments. Problems of this type are known as Multi-fidelity Expensive Black-Box (Mf-EBB) problems. The special cases where only one additional lower fidelity source is available are known as Bi-fidelity Expensive Black-Box (Bf-EBB) problems. These sources offer a trade-off between cost and accuracy, often providing a large amount of less accurate data which can potentially enhance the understanding of the expensive source. When these sources are available, surrogate models often integrate them into the construction of a single model of the costly black-box. The fusion of multiple information sources into a single surrogate model has readily been applied to a very wide range of industrial design problems, from ship component [2, 5, 6] and civil infrastructure [7, 8] design to traffic state estimation [9] and inter-satellite calibration [10]. Aeroplane component design in particular often relies on multi-fidelity surrogate models [11–16], where available data sources include high-fidelity Wind Tunnel (WT) and low-fidelity Computational Fluid Dynamics (CFD) simulation data.

1.1 Motivation

The contributions of this thesis are particularly motivated by the work of stability and control engineers performed at Boeing Australia. The development of mathematical models that predict the performance of plane designs such as lift and drag under certain flight conditions is a key component of the aviation industry. Indeed, constructed surrogate models can be used to assess and improve the performance of new prototype aircraft, but are also used "to drive training simulators, to reconstruct the flight conditions involved in accidents, and to study the effects of modifications to the design" [17].

The construction of these models must take many elements into account, foremost among them the characteristics of the available information sources.

A key element of the industry is the availability of physics-based mathematical models [17] derived over 100 years of aircraft development. These simpler models are often used as the basis for the construction of more refined surrogate models, and as they are analytical they can be viewed as an information source with no associated querying cost. Despite not being able to perfectly predict aircraft performance in all situations, the predictions they provide can sometimes be extremely accurate. The data-gathering process from other data sources is often driven by the need to know the limitations of this first basic source. Once boundary regions where the behaviour of the aircraft begins to differ from the physical models are identified, supplementary data is obtained from other sources. These often include CFD simulation data which can be gathered in large quantities but can sometimes be inaccurate, and WT data which requires the construction of prototypes and can be costly to acquire.

Additional considerations add a layer of complexity to the data gathering and surrogate building process. Some of the information sources, particularly WT, might suffer from noise and therefore be stochastic. Furthermore, the information sources can have different domains i.e. they can't all be queried in some regions of the flight envelope. The analysis of prototypes in a wind tunnel in particular is restricted to certain flight configurations, meaning that CFD data must be used to supplement it. The notion of "trust" in a particular source is also very relevant in the data generation process. One of the key responsibilities of a control engineer is to identify the source that is most likely to provide an accurate prediction of the truth whenever discrepancies exist between information sources. Gaining insights into the characteristics of harmful and beneficial data can therefore lead to improved data gathering and surrogate model construction techniques.

1.2 Aims

The work presented in this thesis aims to investigate some of the many facets of Mf-EBB surrogate modelling, with an emphasis on the concept of trust in an information source. Engineers at Boeing Australia work with the clear notion that information sources can

only be relied upon until a certain point, and finding when the provided output cannot be trusted is crucial for surrogate model performance. To investigate what qualifies as a trustworthy information source, in this thesis some simplifications are taken with regards to the motivating problems from the aviation industry. Most notably, rather than assuming that the truth must be obtained through a combination of information sources, the classic setting of Mf-EBB problems adopted in the academic literature will be investigated. That is, this work assumes that a high-fidelity source provides the true objective function values in all locations of the sample space, but only a sample will be used to understand the truth anywhere in the design space. Information from this source can then be augmented by one or more low-fidelity sources. Furthermore, it is assumed all information sources are deterministic. Most of the work of this thesis focuses on the Bf-EBB setting, by comparing the performance of models constructed using only high-fidelity data with the performance of models using both high- and low-fidelity data. This comparison is used as a proxy when assessing the usefulness of a given low-fidelity source in surrogate model construction.

Work of this nature is sparse but does exist in the literature. Most notably the work of Toal [18] provided a set of guidelines intended to aid industry practitioners in choosing when to rely on a particular information source when constructing surrogate models. The contributions of his work are presented in Chapter 2, as well as some recently developed techniques that benefit from this kind of analysis. Toal's analysis and his proposed guidelines can be seen as an answer to the following questions:

1. What are the properties required of a low-fidelity function to ensure a bi-fidelity model performs no worse than a single source model?
2. How can we develop an algorithm or rules to choose when to use a bi-fidelity or single source model?

The importance of adequate and diverse benchmarks and benchmark measures in answering these questions is illustrated by Toal's work, as he proposed a new set of benchmarks with varying low-fidelity function quality. His newly proposed benchmarks allowed him to investigate the impact of overall low-fidelity function quality on surrogate model accuracy. This thesis continues this line of exploration by also investigating the impact of the local (rather than global) quality characteristics of low-fidelity function

sources on model accuracy. As will be shown in this thesis however, exploring this new aspect requires new benchmarks and measures as the literature has so far only focused on low-fidelity sources with unchanging quality in the whole sample space. Furthermore, most of the studies available of this kind have analysed low-fidelity sources with large datasets which are not available in practice to train the surrogate models. This thesis moves past these "perfect conditions" and instead provides tools to predict when to trust a low-fidelity source based only on the very little information available. Establishing trust in this prediction can lead to new techniques which make informed decisions when generating a dataset given a sample budget rather than an already collected dataset.

Contributions to establishing trust in an information source are therefore provided in this thesis via the development of new benchmarks and benchmark measures, a thorough and bias-free experimental study of Bf-EBB problems, and taking the first steps towards new adaptive techniques for Bf-EBB problems in particular and Mf-EBB problems in general. The goals are therefore to:

- Generate new and more diverse testing benchmarks, or *instances*, for bi-fidelity surrogate modelling construction methods.
- Define new benchmark measures, or *features*, which allow for a more intricate differentiation between benchmarks.
- Employ the newly defined instances and features to conduct an unbiased and comprehensive study of the impact of source quality on surrogate model construction.
- Develop industrial guidelines on information source usage based only on the limited available data used to train the surrogate models.
- Develop new data sampling and model choice techniques which adapt to the characteristics of any given instance.

1.3 Thesis outline and contributions

Chapter 2 provides the necessary fundamentals on the usage of surrogate modelling techniques in Mf-EBB problems. This is achieved by reviewing the classical single-source technique known as Kriging, and its two-source variation known as Co-Kriging,

among others. Methods used to train these models and to choose future samples based on the constructed model as well as other aspects to consider are presented here. A novel illustrative example using Australian elevation data is presented to highlight many of the potential pitfalls when relying on low-fidelity sources without proper prior assessment. This chapter also reviews existing work on characterising harmful data sources.

Chapter 3 argues the need for further work in both benchmark creation and instance characterisation of Bf-EBB problems. An analysis of single-source vs two-source surrogate model accuracy in the form of Kriging vs Co-Kriging is performed twice using different benchmark test suites. Performing this analysis twice will reveal the need for new instances and new instance measures both of which are proposed in this chapter.

Chapter 4 keeps the same focus as the previous chapter, but exploits the newly proposed benchmarks to construct a demonstrably unbiased set of high- and low-fidelity function pairs. This is used to conduct a thorough experimental study of the impact of low-fidelity function characteristics on surrogate model construction. This leads to the characterisation of harmful data sources calculated using only the limited data available, which in turn generates a new set of industrial guidelines that can be directly applied to industrial problems. The procedure followed in this chapter centers on applying the technique known as Instance Space Analysis (ISA) to Bf-EBB problems for the first time in the literature. This technique is exploited to generate an intuitive understanding of what constitutes a harmful low-fidelity source. This intuition is key both in developing trust in available information sources, and trust from the industrial practitioners in the obtained results.

Chapter 5 shifts focus to budget-constrained modelling, where techniques aim to model a black-box but are given the freedom to select both where in the sample domain and from which source to gather the data samples used to train the surrogate models. A variety of sampling approaches, which are guided by models trained on available data, are examined. This includes an adaptive sampling approach which chooses a source to sample based on the trade-off between cost and predicted model accuracy gain. This chapter also showcases how the newly proposed guidelines can enhance the performance of current surrogate modelling tools via adaptive model choice. Techniques that rely on the guidelines to learn whether the low-fidelity source can be trusted are shown to select models with improved accuracy over techniques that do not perform this assessment.

Chapter 6 provides concluding remarks and promising areas for future research.

The linear structure of this thesis is illustrated by Figure 1.1. The main contributions of Chapters 3, 4 and 5 in particular are obtained by focusing on both data choices (shown in green) and modelling choices (shown in blue). The modelling choices are investigated through the development of new features, new guidelines based on these features, and an adaptive choice of model based on these guidelines. The data choices ensure the insights obtained are as applicable and accurate as possible. This is achieved by generating new and diverse instances, ensuring the generated benchmark suite is unbiased, and that the source sampling leads to an accurate choice of model.

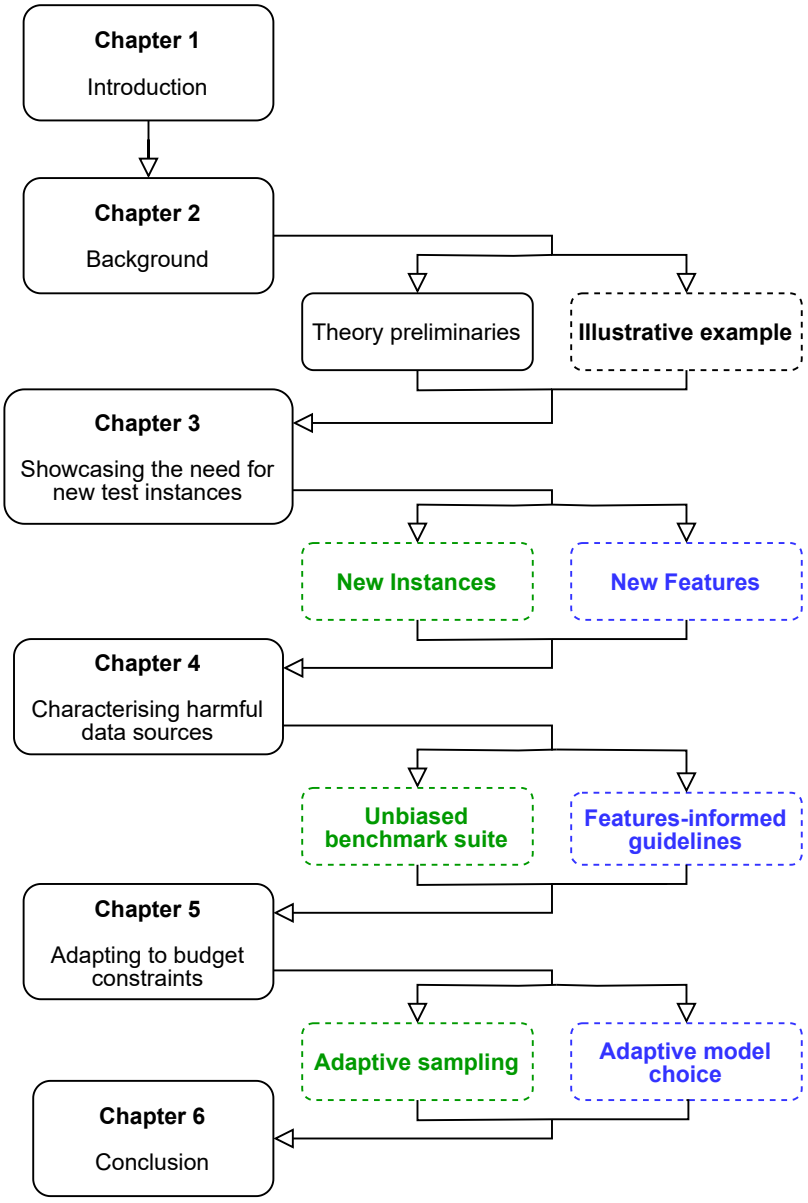


FIGURE 1.1: Thesis structure (left) with the main contents of each chapter (right). The boxes with a dashed outline signify novel contributions to the field, and the colours indicate the focus that has driven them: data choices (green) and modelling choices (blue).

Chapter 2

Background

As stated in the preface, most of the content of this chapter constitutes an adaptation of the accepted paper "Methodology and challenges of surrogate modelling methods for multi-fidelity expensive black-box problems". The introduction and conclusion have been adapted from this paper to fit the thesis format. Section 2.4 is adapted from the published paper "Bi-fidelity Surrogate Modelling: Showcasing the Need for New Test Instances".

This chapter provides the necessary background when working with Mf-EBB problems. Section 2.1 formally defines the different variations of Mf-EBB problems one might consider, as different scenarios arise when the end goal is the construction of an accurate model or the optimisation of a design. Section 2.2 presents a variety of surrogate models, including polynomial regression models, Radial Basis Functions (RBFs) and Kriging, and the adaptation of surrogate models to the Mf-EBB setting. Other considerations such as choosing further sample sites via acquisition functions and the initial design of experiments are also provided. Section 2.3 introduces an illustrative example using Australian elevation data and compares the performance of Kriging and Co-Kriging models which rely on low-fidelity sources of varying quality. This helps highlight the importance of being selective when augmenting high-fidelity data with low-fidelity data. Section 2.4 presents existing work in the literature that examines what low-fidelity characteristics might be required for low-fidelity data to be beneficial. Finally, Section 2.5 concludes the chapter with some observations and open questions in the literature which drive the rest of the work in this thesis.

2.1 Problem definitions

The common component to Mf-EBB problems is the existence of a high-fidelity source f_h and one or more low-fidelity sources $f_{l_1}, f_{l_2}, \ldots, f_{l_k}$. The term $f_h(\mathbf{x})$ denotes the true performance metric value of a design $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ obtained via the costly high-fidelity procedure. This is the source which will be modelled and potentially optimised as discussed in the following subsections. The term $f_{l_i}(\mathbf{x})$ denotes the approximation of the true value obtained from source l_i. It is assumed that the input space being investigated is a hypercube, namely the sources are formally defined as

$$f_h, f_{l_1}, f_{l_2}, \ldots, f_{l_k} : \Omega \to \mathbb{R},$$
$$\Omega = [x_1^{\perp}, x_1^{\top}] \times \cdots \times [x_d^{\perp}, x_d^{\top}] \subset \mathbb{R}^d,$$

where the hypercube Ω has bounds $[x_i^{\perp}, x_i^{\top}]$ for dimension i. It is often the case that each of the low-fidelity sources represents a progressive decrease both in accuracy and cost relative to f_h. This however is not a requirement, and a source could be cheaper but more accurate than another, or more accurate in certain regions of the space. It is up to the surrogate modelling methods being employed to determine how to rely on these sources based on cost and apparent usefulness.

As discussed in Chapter 1, one of the underlying assumptions in this work is the fact that all information sources are deterministic, and therefore sampling a source at an already sampled location will only return an already known value. It is possible however for one or more of the sources to be stochastic, that is sampling the same location repeatedly will lead to different values. Modelling techniques developed for stochastic sources can be quite different to the approaches presented here. For the interested reader in surrogate models of stochastic EBB sources the reader is directed to the work of Ankenman et al. [19].

It is worth highlighting at this point that studies of Mf-EBB problems are conducted in one of two broad settings, namely a synthetic and an applied setting. In a synthetic setting, all sources are only assumed to be expensive. In reality, they are almost always analytical (i.e. they are based on a mathematical expression), and the low-fidelity sources are often a modification of the high-fidelity source via the addition of some terms or the modification of the coefficients of the expression [18, 20]. Studies that focus on this

setting have large amounts of data from each of the sources available despite using only limited samples when training models or optimising. This large amount of data can be used to accurately assess the accuracy of constructed models, as well as to characterise the sources themselves to analyse their impact on algorithm performance [18]. This type of study can provide insights into the inner workings of surrogate model methods and provide guidelines for their usage.

It is also possible for researchers to focus on the applied setting, that is studies that focus on the limited data available and how to exploit it. This is particularly relevant to industrial problems, where the information sources are true expensive black-boxes and very little is known about them. Studies of this type primarily propose new ways to identify useful low-fidelity sources [21], or ways to exploit low-fidelity information when it seems beneficial to do so [22]. Guidelines and insights developed in a synthetic setting can often impact the development of these applied techniques. Both settings therefore have their place and are beneficial to the literature. Mf-EBB problems can be further classified into three types of sub-problems. This division is characterised by the given resources, the aim and the success measure being used. These sub-problems are defined next and are later illustrated in Section 2.3.

2.1.1 Problem 1: Model creation with fixed sample

In this scenario, data from the available sources has already been gathered, and no further sampling of any of the sources is allowed. That is, the sets $\mathbf{X}_h, \mathbf{X}_{l_1}, \ldots, \mathbf{X}_{l_k}$ and $\mathbf{y}_h, \mathbf{y}_{l_1}, \ldots, \mathbf{y}_{l_k}$ are given defined as

$$\begin{aligned}
\mathbf{X}_h &= \{\mathbf{x}_1^h, \mathbf{x}_2^h, \ldots, \mathbf{x}_{n_h}^h\} \subset \Omega, \\
\mathbf{X}_{l_i} &= \{\mathbf{x}_1^{l_i}, \mathbf{x}_2^{l_i}, \ldots, \mathbf{x}_{n_{l_i}}^{l_i}\} \subset \Omega \qquad \text{for } 1 \leq i \leq k, \\
\mathbf{y}_h &= \{f_h(\mathbf{x}_1^h), \ldots, f_h(\mathbf{x}_{n_h}^h)\}, \\
\mathbf{y}_{l_i} &= \{f_{l_i}(\mathbf{x}_1^{l_i}), \ldots, f_{l_i}(\mathbf{x}_{n_{l_i}}^{l_i})\} \qquad \text{for } 1 \leq i \leq k.
\end{aligned}$$

As the data gathered comes from an expensive source, the amount of high-fidelity samples is relatively small; a bound of $n_h \leq 20d$ is common in the literature [18], where d is the problem dimension. Furthermore, if the sources $f_h, f_{l_1}, f_{l_2}, \ldots, f_{l_k}$ are assumed to

be of decreasing quality, it may be that $\mathbf{X}_h \subseteq \mathbf{X}_{l_1} \subseteq \mathbf{X}_{l_2} \subseteq \cdots \subseteq \mathbf{X}_{l_k}$, although this is not always the case.

The aim in this problem is to construct a model $s_h : \Omega \to \mathbb{R}$ of f_h which is as accurate as possible given the available data. The performance of the constructed model in a synthetic setting is measured using a large sample set $\mathbf{X}$ and $\mathbf{y}$ defined as

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

where N should be a large number e.g. $N = 1000d$. Given this sample, two accuracy measures are often used in the literature. The first is the Root Mean Squared Error (RMSE) between s_h and f_h, given by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(s_h(\mathbf{x}_i) - f_h(\mathbf{x}_i))^2}{n}},$$

with a lower error indicating better performance. It is also possible to make this measure relative to the range of the function being modelled to obtain an accuracy between 0 and 1. This is known as the Relative Root Mean Squared Error (RRMSE) and is given by

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

Another possible measure is the Pearson's correlation P_{corr} between s_h and f_h, given by

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

$$\text{where} \qquad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} f_h(\mathbf{x}_i),$$

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

$$\bar{S} = \frac{1}{N}\sum_{i=1}^{N} s_h(\mathbf{x}_i),$$

$$s_S = \left[\frac{\sum_{i=1}^{N}(s_h(\mathbf{x}_i)-\bar{S})^2}{N-1}\right]^{1/2},$$

with a higher correlation indicating better performance. Whilst it is rare to find industrial examples that fit this problem definition, it is quite often used in studies that propose new surrogate modelling methods. Here the practitioner's decision-making is restricted to a single question: Given a set of data, which model should be trained? The performance of different choices such as how to divide a total budget between different phases of an algorithm or the usage of different data acquisition strategies does not need to be assessed. This allows for an easy comparison between models.

2.1.2 Problem 2: Model creation with sample budget

In this scenario, rather than being given already collected data from each of the sources, the user is given a total sampling budget of B to be used as desired. This budget indicates how many times the source f_h can be queried. Furthermore, a given set of cost ratios $0 \leq C_r^{l_1}, C_r^{l_2}, \ldots, C_r^{l_k} \leq 1$ indicates the cost of sampling the sources $f_{l_1}, f_{l_2}, \ldots, f_{l_k}$, respectively, relative to the cost of f_h. For instance, a cost ratio $C_r^{l_i} = 0.1$ indicates that sampling f_{l_i} is 10 times cheaper than sampling f_h, and so one f_h sample can be replaced by 10 f_{l_1} samples. The special case where $C_r^{l_i} = 0$ indicates that the source f_{l_i} is cheap, that is it can be queried as often as desired with no associated cost.

The aim here is to gather the data $\mathbf{X}_h, \mathbf{X}_{l_1}, \ldots, \mathbf{X}_{l_k}$ and $\mathbf{y}_h, \mathbf{y}_{l_1}, \ldots, \mathbf{y}_{l_k}$ from each of the sources as defined in the previous subsection whilst satisfying

$$B \geq |\mathbf{X}_h| + C_r^{l_1}|\mathbf{X}_{l_1}| + \cdots + C_r^{l_k}|\mathbf{X}_{l_k}|.$$

This data is used to train a model s_h of f_h which is as accurate as possible. Once again the performance is measured in terms of the accuracy of the constructed model, with either the $RMSE$ or P_{corr} measures.

This type of problem is a step up from the previous one. The practitioner needs to choose not only what kind of surrogate model to use, but also how to divide the budget among fidelities and how to guide future data gathering. It is also much more realistic in terms of industrial problems, where the aim is not to optimise a design but rather to understand its performance under a range of conditions.

2.1.3 Problem 3: Function optimisation with sample budget

In this final scenario, the user is once again given the sources $f_h, f_{l_1}, f_{l_2}, \ldots, f_{l_k}$, a total budget B and a set of cost ratios $0 \leq C_r^{l_1}, C_r^{l_2}, \ldots, C_r^{l_k} \leq 1$. The aim is to either minimise or maximise f_h, whilst sampling each of the sources at locations $\mathbf{X}_h, \mathbf{X}_{l_1}, \ldots, \mathbf{X}_{l_k}$ and satisfying

$$B \geq |\mathbf{X}_h| + C_r^{l_1}|\mathbf{X}_{l_1}| + \cdots + C_r^{l_k}|\mathbf{X}_{l_k}|.$$

The performance measure is simply the best objective function value of the sampled points $\mathbf{X}_h$. Whilst being very similar to the previous problem, the key difference here lies in the fact that one must balance the construction of an accurate surrogate to aid in optimisation, and the optimisation process itself. Sampling low- and high-fidelity sources in unexplored regions can help generate more accurate models. Data gathering therefore needs to balance a need for exploration of the space to find new promising regions, with the exploitation of already discovered good regions to find the best points possible.

2.2 Surrogate modelling techniques

As outlined in the introduction, one of the more prominent ways to approach Mf-EBB problems is through the construction of a surrogate model. Doing so applies to all three of the Mf-EBB sub-problems defined in the previous section, whether the construction of a model is the end goal itself or when optimising an expensive source. Practitioners need to choose a model that not only accurately models the expensive source with limited data, but also has a suitable mechanism to fuse information from low-fidelity sources. Furthermore, a suitable acquisition function must be chosen, a measure of how promising further sample locations are when exploring the space, exploiting known promising regions, and often balancing the two. Finally, when given a total budget with no initial data, an initial sample needs to be gathered to construct an initial surrogate model. Techniques that solve each of these aspects are presented here.

2.2.1 Single-source surrogate models

Despite a very large amount of data modelling techniques reported in the literature, dealing with expensive sources and small data sets often leads to the usage of models which might take some time to train, but which can be highly accurate even with limited information. Techniques such as Neural Networks are ill-suited to problems of this type as training such models with little data can easily lead to over fitting. Instead, techniques such as Kriging [3, 23, 24] and Radial Basis Functions (RBFs) [25–29] are preferred. These techniques have a relatively small amount of parameters to train and can provide remarkably accurate predictions of an objective function even with small data sets.

2.2.1.1 Polynomial regression

Polynomial regression might be the simplest approach to modelling a set of observations. The prediction at a location $\mathbf{x}$ is given by

$$s^{poly}(\mathbf{x}) = \alpha_0 + \sum_{i=1}^{d} \alpha_i x_i + \sum_{i=1}^{d} \sum_{j=i}^{d} \alpha_{i,j} x_i x_j + \cdots + \epsilon(\mathbf{x}).$$

Here the user must choose the degree of the polynomial being trained. The term $\epsilon(\mathbf{x})$ represents the difference between the model prediction and the actual function value, that is,

$$\epsilon(\mathbf{x}) = f_h(\mathbf{x}) - s^{poly}(\mathbf{x}).$$

Given a set of samples $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ the coefficients $\alpha_0, \alpha_1, \ldots, \alpha_d, \ldots$ are calculated by minimising some form of the error function. This often takes the form of the root of the sum of squares

$$\underset{\alpha_0,\alpha_1,\ldots,\alpha_d,\ldots}{\operatorname{argmin}} \sqrt{\sum_{i=1}^{n} \epsilon(\mathbf{x}_i)^2}.$$

This type of surrogate model is non-interpolating; that is the predictions might differ from the objective function value at the observed locations. Polynomial regression is rarely used in EBB problems however as they tend to under-perform when compared to other methods [29, 30].

2.2.1.2 Radial Basis Functions

Radial basis functions (RBF) were first introduced by J. Duchon [25]. Given a set of n samples $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the constructed model takes the form

$$s^{rbf}(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + \sum_{j=1}^{m} c_j \pi_j(\mathbf{x}),$$

where $\|.\|$ represents the Euclidean norm, $\lambda_1, \ldots, \lambda_n, c_1, \ldots, c_m$ are the parameters to be trained, and $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ form a basis which spans all multivariate polynomials of degree m_0 (note $m = \binom{m_0+d}{d}$). For example, for $d = 3$ and $m_0 = 2$, the set of $\pi_i(\mathbf{x})$'s is given by

$$\{1, x_1, x_2, x_3, x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2\}.$$

Note that the polynomial term is not a strict requirement of an RBF model, however its usage is recommended to guarantee the existence and uniqueness of an RBF model

which interpolates the given data [26]. The chosen function $\phi(r)$ for $r \geq 0$ constitutes the basis for the model and several options exist, including

$$\begin{aligned}
\phi(r) &= r && \text{(linear)},\\
\phi(r) &= r^3 && \text{(cubic)},\\
\phi(r) &= r^2 \log r && \text{(thin plate spline)},\\
\phi(r) &= \sqrt{r^2 + \gamma^2} && \text{(multiquadratic)},\\
\phi(r) &= e^{-\gamma r^2} && \text{(Gaussian)},
\end{aligned}$$

where γ is a fixed positive constant. A total of $n + m$ constraints are simultaneously solved to choose the $n + m$ parameters of the model. The first n constraints guarantee the trained model interpolates the function observations

$$s(\mathbf{x}_k) = \sum_{i=1}^{n} \lambda_i \phi(\|\mathbf{x}_k - \mathbf{x}_i\|) + \sum_{j=1}^{m} c_j \pi_j(\mathbf{x}_k) = f(\mathbf{x}_k) \quad \forall \quad k = 1, \ldots, n.$$

The second set of constraints guarantees that if the function being modelled is a polynomial of degree m_0 or less, the surface will represent it exactly

$$\sum_{i=1}^{n} \lambda_i \pi_k(\mathbf{x}_i) = 0 \qquad \forall \quad k = 1, \ldots, m.$$

RBFs have many attractive qualities for EBB problems. They are known to model the curvature of functions being modelled quite well, and the predictions at known locations are the same as the objective function value. One of their potential shortcomings however is the fact that these models assume the impact on the objective function is similar for each of the input variables. This shortcoming is addressed by the following technique.

2.2.1.3 Kriging

Kriging provides very accurate models due to the combination of its high number of hyperparameters and an approach to choose them rooted in sound probability theory. What follows is a brief summary of Kriging; the full theoretical derivation is given in Appendix A. The formulation given by Kriging assumes the function samples made so far at locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are realisations of random normal variables $Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_n)$

with mean μ and variance σ^2. Further, the correlation between variables is based on the distance between them, that is,

$$\mathrm{Corr}(Y(\mathbf{x}_i), Y(\mathbf{x}_j)) = R(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\sum_{k=1}^{d} \theta_k |\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}|^{p_k}\right\},$$

where $\mathbf{x}^{(k)}$ denotes the kth coordinate of point $\mathbf{x}$. Thus, the multivariate random variable $\mathbf{Y} = [Y(\mathbf{x}_1) \dots Y(\mathbf{x}_n)]$ has the distribution $\mathbf{Y} \sim N(\mathbf{1}\mu, \sigma^2\mathbf{R})$, with $\mathbf{R}_{i,j} = R(\mathbf{x}_i, \mathbf{x}_j)$. Simple Kriging assumes constant mean μ across the space, but it is possible to fit a regression model to the data before adding the Kriging "layer"; models of this kind are known as Universal Kriging [31] models. It is also possible to use alternative formulas (also known as kernel functions) for the correlation between two samples. Other notable examples include the Matern-3/2, Matern-5/2 and cubic kernels; for a comparison of their performances the reader is directed to the work of Palar et al. [32]. Another common choice is to use the formulation given above, but with the simplification $p_1 = \dots = p_d = 2$. Nevertheless, the kernel function used in this work is the same as in the classical study of Jones [3].

The hyperparameters of a simple Kriging model are $\mu, \sigma^2, \theta_1, \dots, \theta_d$ and $p_1, \dots, p_d$. The values θ_k and p_k give an indication of the effect of moving along any of the dimensions (i.e. changing the value of a single variable) on the objective function. The hyperparameter θ_k represents how the correlation changes with distance: large values mean there is no correlation even for close points in the kth dimension, but small values indicate even relatively distant sample points (in the kth dimension) are correlated. The constant p_k allows the technique to model from smooth functions ($p_k = 2$) to rough, non-differentiable ones ($p_k \to 0$). These hyperparameters are chosen to be Maximum Likelihood Estimators (MLEs), found by maximising the so-called concentrated log-likelihood

$$\underset{\theta_1,\dots,\theta_d,p_1,\dots,p_d}{\mathrm{argmax}} \; -\frac{n}{2}\log(\hat{\sigma}^2) - \frac{1}{2}\log(|\mathbf{R}|),$$

$$\begin{aligned} \text{where } \hat{\mu} =& \frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}, \\ \hat{\sigma}^2 =& \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})}{n}. \end{aligned}$$

Training a Kriging model therefore consists of solving an auxiliary optimisation problem where the concentrated log-likelihood is maximised with respect to $\theta_1, \ldots, \theta_d$ and $p_1, \ldots, p_d$. Finding good black-box algorithms to solve this problem is an active area of research in and of itself [33, 34], but in theory, any solver can be used such as Accelerated Random Search [35]. Appendix B provides the implemented auxiliary black-box algorithms used in this thesis when training Kriging models.

Once the model has been trained, Kriging treats a prediction at a location $\mathbf{x}$ in the space as having the distribution $\sim N(s^{krig}(\mathbf{x}), v^{krig}(\mathbf{x}))$. That is, Kriging provides not only the most likely objective function value $s^{krig}(\mathbf{x})$, but also a prediction error metric $v^{krig}(\mathbf{x})$, defined as

$$
\begin{aligned}
s^{krig}(\mathbf{x}) &= \hat{\mu} + \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}), \\
v^{krig}(\mathbf{x}) &= \hat{\sigma}^2 \left[1 - \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1}\mathbf{r}(\mathbf{x})\right], \\
\text{where } \mathbf{r}(\mathbf{x}) &= \begin{bmatrix} R(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ R(\mathbf{x}, \mathbf{x}_n) \end{bmatrix}.
\end{aligned} \tag{2.1}
$$

The prediction error metric $v^2(\mathbf{x})$ in particular is a helpful metric used to guide further sampling. This is further discussed in Section 2.2.4.

2.2.2 Multi-source surrogate models

Having chosen the type of surrogate model, the next decision to be made is how to incorporate multiple information sources into a single prediction of the high-fidelity source. By far the most common approach is take an additive approach; that is in the case where the sources $f_h = f_{l_0}, f_{l_1}, \ldots, f_{l_k}$ are given, to first train a model s_k of f_{l_k}, and to then iteratively train a model s_i of the difference $f_{l_i} - f_{l_{i+1}}$ for $i = 0, \ldots, k-1$. The resulting prediction of the high-fidelity source is then given by

$$
s_h(\mathbf{x}) = \sum_{i=0}^{k} s_i(\mathbf{x}).
$$

Applying this framework to Kriging leads to the technique known as Hierarchical Kriging [36]. The use of probability theory underlying Kriging however allows for a more intricate approach, first proposed by Kennedy and O'Hagan [37]. Their work assumes the observed objective function value $f_{l_k}(\mathbf{x})$ at the location $\mathbf{x}$ is is the realisation of the random process $Y_{l_k}(\mathbf{x})$. For fidelities l_t with $0 \leq t \leq k-1$, the observed value $f_{l_t}(\mathbf{x})$ at the location $\mathbf{x}$ is assumed to come from the random process

$$Y_{l_t}(\mathbf{x}) = \rho_{l_{t+1}} Y_{l_{t+1}}(\mathbf{x}) + Y_{l_t}^{diff}(\mathbf{x}) \qquad \text{for } 0 \leq t \leq k-1. \tag{2.2}$$

Here $\rho_{l_{t+1}}$ is kind of a regression parameter, $Y_{l_{t+1}}(\mathbf{x})$ is the random process representing the values obtained from source l_{t+1}, and the difference term $Y_{l_t}^{diff}(\mathbf{x})$ is assumed to be independent of $Y_{l_{t+1}}(\mathbf{x}), \ldots, Y_{l_k}(\mathbf{x})$. The added benefit of the ρ parameters is the flexibility to scale each of the information sources. The combination of this framework with Kriging in the special case where only two sources are available is known as Co-Kriging [38] and is presented next. Note that it is similarly possible to work with RBFs, resulting wih the model known as Co-RBF [39, 40]. Unlike Co-Kriging however, no clear approach exists to assign appropriate values to the ρ parameter in such models.

It is also worth noting that these additive approaches do assume that each of the low-fidelity sources progressively decreases in accuracy. More recent approaches such as the one proposed by Eweis-Labolle et al. [41] bypass this assumption by pretending all the available data has been obtained from the same source. This is achieved by augmenting the set of inputs of the model with a categorical variable, and training a single model with $d+1$ dimensional data. The potential of this approach has recently been illustrated in the work of Toal [42], at least for the two industrial problems investigated in the study. The work in this thesis however will focus on the comparison of Kriging and Co-Kriging models, as despite having been developed more than 20 years ago, these techniques are still relevant today due to their strong underlying theory and its supplied uncertainty metric. Furthermore, many newer approaches are variations of or inspired by these techniques.

2.2.2.1 Co-Kriging

What follows is a brief summary of Co-Kriging; the full theoretical derivation is given in Appendix A. The training data used consists of a set of n_l locations $\mathbf{X}_l = \{\mathbf{x}_1^l, \mathbf{x}_2^l, \ldots, \mathbf{x}_{n_l}^l\}$ with known low-fidelity function values $\mathbf{y}_l = \{y_1^l, \ldots, y_{n_l}^l\} = \{f_l(\mathbf{x}_1), \ldots, f_l(\mathbf{x}_{n_l})\}$, and a set of n_h locations $\mathbf{X}_h = \{\mathbf{x}_1^h, \mathbf{x}_2^h, \ldots, \mathbf{x}_{n_h}^h\}$ with known high-fidelity function values $\mathbf{y}_h = \{y_1^h, \ldots, y_{n_h}^h\} = \{f_h(\mathbf{x}_1), \ldots, f_h(\mathbf{x}_{n_h})\}$. The Co-Kriging framework models the low- and high-fidelity responses as realisations of a multivariate random variable

$$\mathbf{Y} = (\mathbf{Y}_l(\mathbf{X}_l), \mathbf{Y}_h(\mathbf{X}_h)) = (Y_l(\mathbf{x}_1^l), \ldots, Y_l(\mathbf{x}_{n_l}^l), Y_h(\mathbf{x}_1^h), \ldots, Y_h(\mathbf{x}_{n_h}^h)).$$

The multivariate random variable $\mathbf{Y}_l(\mathbf{X}_l)$, that is the response of the cheap objective function, is treated as a multivariate normal random variable with distribution $\sim N\left(\mu_l, \sigma_l^2 \mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)\right)$. The multivariate random variable $\mathbf{Y}_h(\mathbf{X}_h)$, that is the response of the expensive objective function, is represented by a scaling of ρ of the response of the cheap expensive function $\mathbf{Y}_l(\mathbf{X}_h)$ plus a new Gaussian process $\mathbf{Y}_{diff}(\mathbf{X}_h)$ which models the difference between the cheap and expensive objective functions, that is

$$\mathbf{Y}_h(\mathbf{X}_h) = \rho \mathbf{Y}_l(\mathbf{X}_h) + \mathbf{Y}_{diff}(\mathbf{X}_h).$$

The random variable $\mathbf{Y}_{diff}$ is also treated as a multivariate normal random variable with distribution $\sim N\left(\mu_{diff}, \sigma_{diff}^2 \mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)\right)$. It is assumed that $\mathbf{Y}_l$ and $\mathbf{Y}_{diff}$ are independent. Thus the following covariance measures are given for $\mathbf{Y}_l$ and $\mathbf{Y}_h$

$$\begin{aligned}
\mathrm{Cov}(Y_l(\mathbf{x}_i^l), Y_l(\mathbf{x}_j^l)) &= \sigma_l^2 R_l(\mathbf{x}_i^l, \mathbf{x}_j^l), \\
\mathrm{Cov}(Y_h(\mathbf{x}_i^h), Y_l(\mathbf{x}_j^l)) &= \rho \sigma_l^2 R_l(\mathbf{x}_i^h, \mathbf{x}_j^l), \\
\mathrm{Cov}(Y_h(\mathbf{x}_i^h), Y_h(\mathbf{x}_j^h)) &= \rho^2 \sigma_l^2 R_l(\mathbf{x}_i^h, \mathbf{x}_j^h) + \sigma_{diff}^2 R_{diff}(\mathbf{x}_i^h, \mathbf{x}_j^h),
\end{aligned}$$

where

$$R_l(\mathbf{x}_1, \mathbf{x}_2) = \exp\left\{-\sum_{k=1}^{d} \theta_k^l |\mathbf{x}_1^{(k)} - \mathbf{x}_2^{(k)}|^{p_k^l}\right\}, \tag{2.3}$$

$$R_{diff}(\mathbf{x}_1, \mathbf{x}_2) = \exp\left\{-\sum_{k=1}^{d} \theta_k^{diff} |\mathbf{x}_1^{(k)} - \mathbf{x}_2^{(k)}|^{p_k^{diff}}\right\}. \tag{2.4}$$

The following matrices are now defined

$$\begin{aligned}
\mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)_{i,j} &= R_l(\mathbf{x}_i^l, \mathbf{x}_j^l) \qquad 1 \leq i, j \leq n_l, \\
\mathbf{R}_l(\mathbf{X}_h, \mathbf{X}_l)_{i,j} &= R_l(\mathbf{x}_i^h, \mathbf{x}_j^l) \qquad 1 \leq i \leq n_h \quad 1 \leq j \leq n_l, \\
\mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_h)_{i,j} &= R_l(\mathbf{x}_i^l, \mathbf{x}_j^h) \qquad 1 \leq i \leq n_l \quad 1 \leq j \leq n_h, \\
\mathbf{R}_l(\mathbf{X}_h, \mathbf{X}_h)_{i,j} &= R_l(\mathbf{x}_i^h, \mathbf{x}_j^h) \qquad 1 \leq i, j \leq n_h, \\
\mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)_{i,j} &= R_{diff}(\mathbf{x}_i^h, \mathbf{x}_j^h) \qquad 1 \leq i, j \leq n_h.
\end{aligned}$$

Training the hyperparameters of a Co-Kriging model is a two-step process. First, the MLEs $\hat{\mu}_l, \hat{\sigma}_l^2, \hat{\theta}_1^l, \ldots, \hat{\theta}_d^l, \hat{p}_1^l, \ldots, \hat{p}_d^l$ are found similarly to the training of a single-source Kriging model. That is, a numerical solution is found to the auxiliary optimisation problem

$$\max_{\theta_1^l, \ldots, \theta_d^l, p_1^l, \ldots, p_d^l} -\frac{n_l}{2}\log(\hat{\sigma}_l^2) - \frac{1}{2}\log(|\mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)|),$$

$$\text{where } \hat{\mu}_l = \frac{\mathbf{1}^T\mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)^{-1}\mathbf{y}_l}{\mathbf{1}^T\mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)^{-1}\mathbf{1}},$$
$$\hat{\sigma}_l^2 = \frac{(\mathbf{y}_l - \mathbf{1}\hat{\mu}_l)^T\mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)^{-1}(\mathbf{y}_l - \mathbf{1}\hat{\mu}_l)}{n_l}.$$

Once again, the implemented algorithms used for training Co-Kriging models are detailed in Appendix B. Next the vector $\mathbf{d} = \mathbf{y}_h - \rho\mathbf{y}_l(\mathbf{X}_h)$ is defined, where $\mathbf{y}_h = (f_h(\mathbf{x}_1^h), \ldots, f_h(\mathbf{x}_{n_h}^h))^\top$, and $\mathbf{y}_l(\mathbf{X}_h)_i$ is $f_l(\mathbf{x}_i^h)$ if this value is known, and otherwise it is $\hat{\mu}_l + \mathbf{r}_l(\mathbf{x})^\top\mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)^{-1}(\mathbf{y}_l - \mathbf{1}\hat{\mu}_l)$, with $\mathbf{r}_l(\mathbf{x}) = (R_l(\mathbf{x}, \mathbf{x}_1^l), \ldots, R_l(\mathbf{x}, \mathbf{x}_{n_l}^l))^\top$. That is, if a point has not been evaluated by f_l yet, the trained Kriging predictor of the low-fidelity source is used instead. The MLEs of the second set of hyperparameters are chosen to be the numerical solution of the auxiliary optimisation function

$$\max_{\rho, \theta_1^{diff}, \ldots, \theta_d^{diff}, p_1^{diff}, \ldots, p_d^{diff}} -\frac{n_h}{2}\log(\hat{\sigma}_{diff}^2) - \frac{1}{2}\log(|(\mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)|),$$

$$\text{where } \hat{\mu}_{diff} = \frac{\mathbf{1}^T\mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)^{-1}\mathbf{d}}{\mathbf{1}^T\mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)^{-1}\mathbf{1}},$$
$$\hat{\sigma}_{diff}^2 = \frac{(\mathbf{d} - \mathbf{1}\hat{\mu}_{diff})^T\mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)^{-1}(\mathbf{d} - \mathbf{1}\hat{\mu}_{diff})}{n_h}.$$

Similarly to Kriging, a trained Co-Kriging model treats a prediction at a location $\mathbf{x}$ as having the distribution $\sim N(s^{cokrig}(\mathbf{x}), v^{cokrig}(\mathbf{x}))$, providing both a prediction $s^{cokrig}(\mathbf{x})$ and an error metric $v^{cokrig}(\mathbf{x})$. These are defined as

$$s^{cokrig}(\mathbf{x}) = \hat{\mu} + \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}),$$
$$v^{cokrig}(\mathbf{x}) = \hat{\rho}\hat{\sigma}_l^2 + \hat{\sigma}_{diff}^2 - \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1}\mathbf{c}(\mathbf{x}),$$

with
$$\mathbf{C} = \begin{bmatrix} \hat{\sigma}_l^2 \mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l) & \hat{\rho}\hat{\sigma}_l^2 \mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_h) \\ \hat{\rho}\hat{\sigma}_l^2 \mathbf{R}_l(\mathbf{X}_h, \mathbf{X}_l) & \hat{\rho}^2\hat{\sigma}_l^2 \mathbf{R}_l(\mathbf{X}_h, \mathbf{X}_h) + \hat{\sigma}_{diff}^2 \mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h) \end{bmatrix}, \quad (2.5)$$

$$\mathbf{c}(\mathbf{x}) = \begin{bmatrix} \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}_1^l, \mathbf{x}) \\ \vdots \\ \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}_{n_l}^l, \mathbf{x}) \\ \hat{\rho}^2\hat{\sigma}_l^2 R_l(\mathbf{x}_1^h, \mathbf{x}) + \hat{\sigma}_{diff}^2 R_{diff}(\mathbf{x}_1^h, \mathbf{x}) \\ \vdots \\ \hat{\rho}^2\hat{\sigma}_l^2 R_l(\mathbf{x}_{n_h}^h, \mathbf{x}) + \hat{\sigma}_{diff}^2 R_{diff}(\mathbf{x}_{n_h}^h, \mathbf{x}) \end{bmatrix}, \quad (2.6)$$

$$\hat{\mu} = \frac{\mathbf{1}^\top \mathbf{C}^{-1}\mathbf{y}}{\mathbf{1}^\top \mathbf{C}^{-1}\mathbf{1}}.$$

As stated above, Co-Kriging can be viewed as the application of the framework proposed Kennedy and O'Hagan [37] (given in Equation 2.2) to the surrogate model Kriging and for the special case were only two sources are available (i.e. $k = 1$). Whilst this thesis has up until this point discussed Mf-EBB problems by assuming multiple low-fidelity sources may be available, the work in later chapters will solely focus on Bf-EBB problems, that is where only one low-fidelity source is available, via the comparison of Kriging, Co-Kriging and Co-Kriging inspired techniques.

2.2.3 Initial design of experiments

In the case where no initial data is given, building an initial model first requires the sampling of some or all of the available sources. Interestingly, no consensus exists in the literature on what proportion of the total budget should be spent on an initial sample, and how this sample should be further spread among different sources. A recent

2019 survey [4] on the use of Multi-fidelity models did not cover this question despite presenting different techniques to generate an initial sample plan. Forrester et al. [38] suggested a rule of thumb of $10d$ samples, to be further refined based on the total budget and the relative cost of the low-fidelity source. A very recent study [43] however which analysed (single-source) Kriging optimisation on a large variety of functions found that a much smaller initial sample of size $d + 4$ worked best. When optimising a Bi-fidelity EBB problem using RBF surrogate models, Müller [22] found that taking an initial low-fidelity sample of size $2(d + 1)$ and a high-fidelity sample of size $d + 1$ outperformed sampling both the high- and low- fidelity sources $2(d + 1)$ times. Finally, despite not providing a suggested proportion of the total budget to be spent on an initial sample, Toal [18] suggested spending between 10% and 80% of the initial sampling budget on low-fidelity samples. Generating guidelines for the generation of an initial sample is one of the open questions in this field.

Having decided on the number of samples to collect from each of the sources, different approaches exist on how to place these samples within the space. A very common approach is to rely on Latin Hypercube Sampling (LHS) [44]. Generating a sample of size n with this method consists of dividing each of the dimensions of the sample space into n intervals, and then placing n samples in the space so that for every dimension no two samples lie in the same interval. Whilst LHS in theory should provide a well-distributed sample within the space, simply relying on the generation of a random LHS plan can lead to a suboptimal design of experiments. Figure 2.1 in particular shows a sampling plan which despite satisfying the LHS definition, is far from evenly sampling the design space. It is beneficial to generate an LHS plan that satisfies some kind of local optimality criteria, such as the maximisation of the minimum Euclidean distance between any pair of points. This can be achieved by starting with a random LHS plan, and then iteratively taking two samples and swapping one of their coordinates as long as this increases the minimum distance between any pair of points. Once this can no longer be achieved, the obtained sample is locally optimal. The clear benefit of using such a procedure is shown in Figure 2.2. Note that this deterministic optimisation can be quite lengthy at high dimensions and for large sample sizes i.e. $d \geq 10$ and $n \geq 100$, in which case a heuristic approach such as Simulated Annealing [45] should be considered.

When generating a multi-fidelity sample, it is often desirable that the sample plan of a particular fidelity be a subset of the sample plan of the previous fidelity. Sample plans of

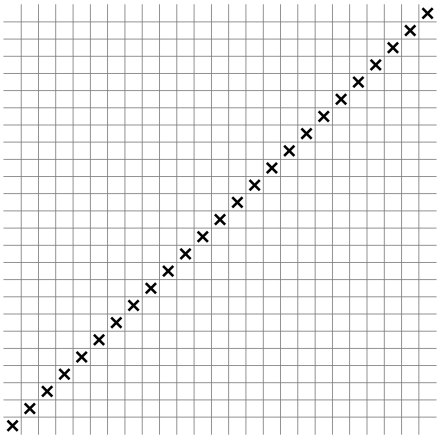
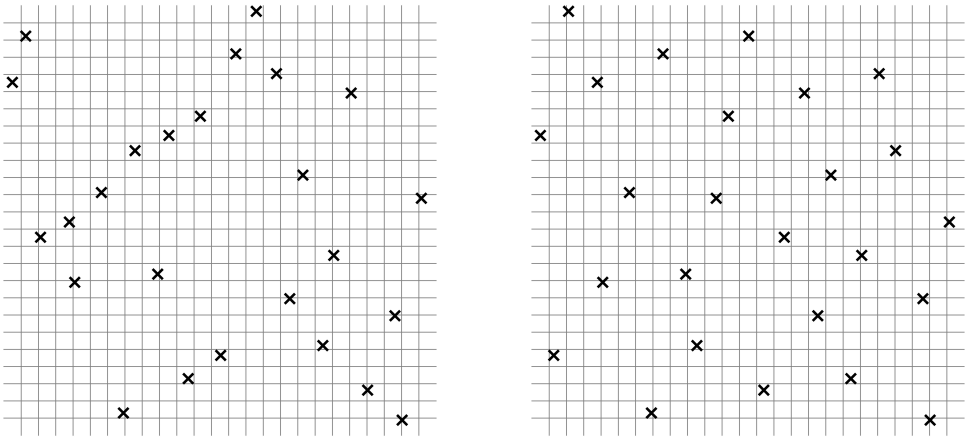


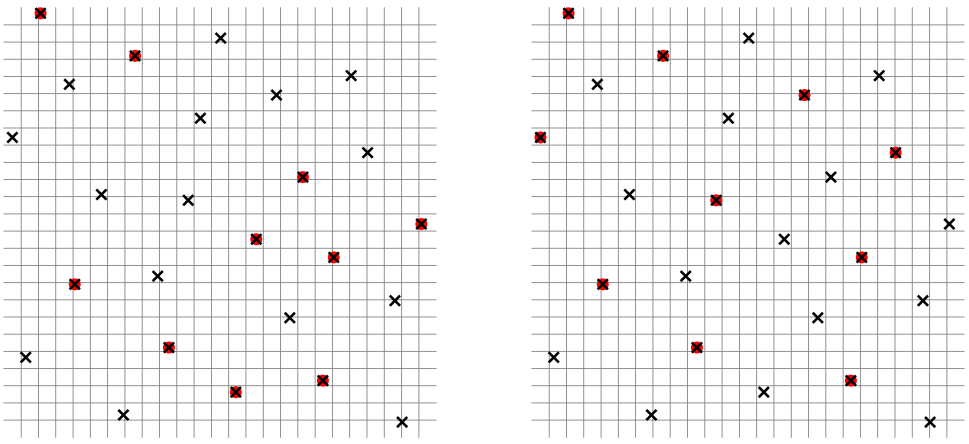
FIGURE 2.1: A 2-dimensional sampling plan of 25 samples with clearly undesirable properties. The thin grey lines indicate the division of each of the two dimensions into 25 sections. Each cross represents a sample. Note that as each row and column contains only a single sample, this satisfies the definition of an LHS plan.



(a) (b)

FIGURE 2.2: A randomly generated LHS sampling plan of 25 samples inside the 2-dimensional space $[0, 1]^2$ is shown in Subfigure a. The minimum Euclidean distance between a pair of points is 0.0754. Subfigure b shows the result of locally optimising the sampling plan by swapping coordinates between pairs of points. This minimum distance between a pair of points is 0.1602, and the sample is more evenly spread out in the space.

this type are known as nested sampling plans, and they tend to be preferred by models which train models of the discrepancy between sources [4]. Given a sample plan of size n_l, the creation of a subset of size n_h can be achieved once again by starting with a random subset and then swapping a point in the subset with a point outside of it as long as this increases the minimum distance between a pair of points. Once this can no longer be done, the set is once again locally optimal. Figure 2.3 shows the benefit of this



(a) (b)

FIGURE 2.3: Procedure when choosing a subset of an LHS plan. The crosses in both subfigures show an LHS plan of size 25 inside the 2-dimensional space $[0, 1]^2$. The red circles in Subfigure a represent a randomly chosen subset of size 10. The minimum distance between pairs of points is 0.1602. Subfigure b shows the results of locally optimising this subset by swapping points inside and outside of the set. The final subset is more evenly spread out, and the minimum distance between pairs of points is 0.2400.

approach over simply choosing a random subset. Another approach when generating a subset is to start with a second LHS sampling plan and to map each of the samples to the larger sample set [46]. Note that this is not a requirement when constructing a Co-Kriging model as outlined in the formulation above, as the trained low-fidelity model can be used to predict low-fidelity function values when these are unknown yet required to train the discrepancy model. Nevertheless, all sampling plans generated in this work are nested sampling plans. In the introduced methods of Chapter 5 it is possible that Co-Kriging models are trained using non-nested datasets; when this occurs the low-fidelity function value is estimated by the trained low-fidelity model as proposed by Forrester et al. [38] (and supplied in the above formulation).

2.2.4 Acquisition function

Once an initial model has been trained, an acquisition function is a means by which to evaluate the benefit of taking a further sample in a particular location. New samples are chosen by finding locations that are the most promising. The definition of "promising" of course depends on the aim of the current process. When the aim is to create as accurate a model as possible, further sampling should rely on the model's error prediction.

Kriging and Co-Kriging particularly shine in this scenario, as their error metric $v(\mathbf{x})$ is particularly well suited for searches of this type. An intuitive first option is to choose further sampling sites to be the location where the surrogate model is most unsure of its prediction, that is to find

$$\underset{\mathbf{x}\in\Omega}{\operatorname{argmax}} \quad v(\mathbf{x}).$$

This approach has the benefit of being intuitive and easy to evaluate. It has been shown however that a large number of samples tend to be gathered on the boundary of the space. These samples might therefore have a low impact on the overall accuracy of the model across the whole space. A second metric which takes this into account can also be considered, namely

$$\underset{\mathbf{x}'\in\Omega}{\operatorname{argmin}} \int_{\mathbf{x}\in\Omega} v_{\mathbf{x}'}(\mathbf{x}).$$

Here the term $v_{\mathbf{x}'}$ is the error prediction function of a surrogate model constructed with the available data with the addition of a sample at location $\mathbf{x}'$. The integral denotes the overall model error across the whole space if the point $\mathbf{x}$ were added to the dataset. This acquisition function takes a more global approach, however, traditionally this integral has been approximated using a technique such as Monte-Carlo integration [47]. Whilst the underlying assumption of Mf-EBB problems is that the computational time of the surrogate method is outweighed by sampling cost, it is worth stressing that this process can become computationally intensive. Promisingly, an analytical expression has been proposed for Kriging models with a reduced set of hyperparameters [48], which might lead to this sampling strategy being applicable for relatively larger sample sizes in the multi-fidelity setting. This approach is detailed in Chapter 5, and a novel adaptation to Co-Kriging is proposed.

When employing a surrogate model in optimisation, a perhaps intuitive first choice is to sample where the model's prediction $s(\mathbf{x})$ is maximised/minimised. That is, in the case of a minimisation problem, to sample at

$$\underset{\mathbf{x}\in\Omega}{\operatorname{argmin}} \quad s(\mathbf{x}).$$

This however tends to not be a good strategy. Samples gathered in this fashion can tend to cluster in the same region of the space, particularly for interpolating models. This is especially counterproductive with a limited sampling budget, as a lot of samples might be wasted finding a local minimum before restarting the search in some other region, as is often done by black-box optimisers. A strategy is needed which, at least part of the time, searches the space for yet-to-be-explored regions that might contain good values. Essentially a balance needs to be struck between exploration to find new promising regions, and further exploitation of regions with good solutions. Approaches such as Target Value sampling [3, 26] or Metric Stochastic Response Surface [27] employ some kind of parameter which specifies how much weight should be given to the exploration and exploitation qualities of new candidate samples, and aim to strike an overall balance between the two by cycling through a set of parameter values which goes from only caring about exploration to only caring about exploitation.

Once again, Kriging and Co-Kriging can rely on their model error prediction metric to provide a sampling strategy which does not require a user-defined parameter to balance exploration and exploitation. Assume the optimisation consists of minimising a function f, and that the best objective function value found so far had the value f^{min}. Recall that both Kriging and Co-Kriging provide a prediction of $f(\mathbf{x})$ at location $\mathbf{x}$ as a normally distributed random variable $\sim N(s(\mathbf{x}), v^2(\mathbf{x}))$. Then the probability that sampling at location $\mathbf{x}$ yields an improvement of $I \geq 0$ over the best point found so far is the probability that $Y(\mathbf{x}) = f^{min} - I$. For $I > 0$ this has the probability density function

$$\frac{1}{\sqrt{2\pi v^2(\mathbf{x})}} \exp\left\{-\frac{1}{2v^2(\mathbf{x})}(f^{min} - I - s(\mathbf{x}))^2\right\}.$$

An analytical expression for the expected improvement exists, and it is given by

$$\begin{aligned}\mathbb{E}[I(\mathbf{x})] &= \int_{I=0}^{I=\infty} I\left[\frac{1}{\sqrt{2\pi v^2(\mathbf{x})}} \exp\left\{-\frac{1}{2v^2(\mathbf{x})}(f^{min} - I - s(\mathbf{x}))^2\right\}\right] \\ &= \sqrt{v^2(\mathbf{x})}\left[\frac{f^{min} - s(\mathbf{x})}{\sqrt{v^2(\mathbf{x})}}\Phi\left(\frac{f^{min} - s(\mathbf{x})}{\sqrt{v^2(\mathbf{x})}}\right) + \phi\left(\frac{f^{min} - s(\mathbf{x})}{\sqrt{v^2(\mathbf{x})}}\right)\right],\end{aligned}$$

where Φ and ϕ are the cumulative density function and probability density function of the standard normal distribution, respectively. Choosing further samples by maximising this expression, that is

$$\underset{\mathbf{x}\in\Omega}{\operatorname{argmax}}\, \mathbb{E}[I(\mathbf{x})],$$

is known as Efficient Global Optimisation [49] and has many nice properties. The first is that it requires no tuning of optimisation parameters once training of the surrogate model is complete. The second is that it automatically balances exploration and exploitation. Note that both very low $s(\mathbf{x})$ values (i.e. locations with good predicted values) and very high $v(\mathbf{x})$ (i.e. locations with high prediction uncertainty) lead to high expected improvement values. The third reason is the fact that locations that have already been sampled have an expected improvement of 0, and therefore will never be selected again for sampling. Maximising this acquisition function can become challenging however, as the function's landscape may consist of a largely flat function with a few high peaks which are hard to find. Variations of this acquisition function have been proposed to circumvent this potential problem, and the interested reader is referred to the work of Forrester and Jones [50].

Choosing the location at which the next sample is collected consists of an auxiliary optimisation problem of the acquisition function of choice. Having chosen the next sampling site, however, the question remains of which of the sources should be sampled. Classical Co-Kriging simply samples both f_h and f_l at the chosen location. This simple approach is beneficial for the accuracy of further Co-Kriging models, as the training of the intermediate model of $f_h - \rho f_l$ can rely on exact values for both f_h and f_l wherever f_h is known. As shown in the formulation of Co-Kriging in Section 2.2.2.1 however, this is not a requirement for training the model. Intuitively, one can see that there might be a benefit in sometimes sampling only f_l if this source is highly accurate, or choosing to not sample f_l if it is not reliable enough.

The question of when to sample low-fidelity sources is another very active area of research for Mf-EBB problems. A potential approach is to take into account the fidelity cost by taking the ratio of the acquisition function at each of the fidelities and the fidelity cost ratio [51, 52]. This leads to the sampling of cheaper low-fidelity sources even if the potential gain is lower than sampling more expensive sources, as the cost-to-gain ratio is

larger. This approach however can suffer from the assumption that cost is a proxy for the quality of a source. Indeed, it is possible for a source that is cheap to be very accurate, and for a source that is relatively costly to be inaccurate. More recent approaches have taken this into account when choosing which source to sample, focusing on Bf-EBB problems. The work of van Rijn et al. [21] presents a framework that estimates the benefit of further sampling a single low-fidelity source when minimising the surrogate model error. This benefit is used to guide the division of the remaining available budget between high- and low-fidelity sources. Müller [22] proposes an optimisation framework in which two surrogate models are trained, the first with only high-fidelity data and the second with low-fidelity data. The high-fidelity model is used to choose further sample locations, whereas the low-fidelity model is used to conduct a preliminary assessment of a newly chosen sample site only when the low-fidelity source is deemed to be trustworthy. Whilst these approaches are relatively new, the need for techniques that conduct some kind of analysis on low-fidelity sources before choosing how to use them is clear. The potential downfalls of not doing so are illustrated in the next section.

2.3 Illustrative example

An example is now introduced to illustrate how techniques such as Co-Kriging which rely on low-fidelity sources can be affected by their quality. In this example, we assume we are map makers tasked with exploring the yet-unknown elevation of Australia. An elevation map of the country is shown in Figure 2.4. Here the only way to exactly measure the elevation at a particular location is to send a team of engineers to take an accurate measurement. Australian elevation can therefore be assumed here to be a two-dimensional EBB denoted f_h. Australian elevation makes for a hard black-box to model and optimise. As shown in the plot, the chosen coordinate window includes three very deep (i.e. objective function values lower than -4000 meters) regions at the bottom, top-left and top-right, a very flat land above sea level, and only a few regions of high altitude, with the highest point being Mt Kosciuszko in New South Wales at 2228m shown as the middle red triangle in the group of three on the bottom right of the map.

Two additional low-fidelity sources are available in this example to aid in both exploration and optimisation. The first is denoted f_l^{error} and is assumed to consist of existing inaccurate elevation records, generated here by adding a deterministic error to the true

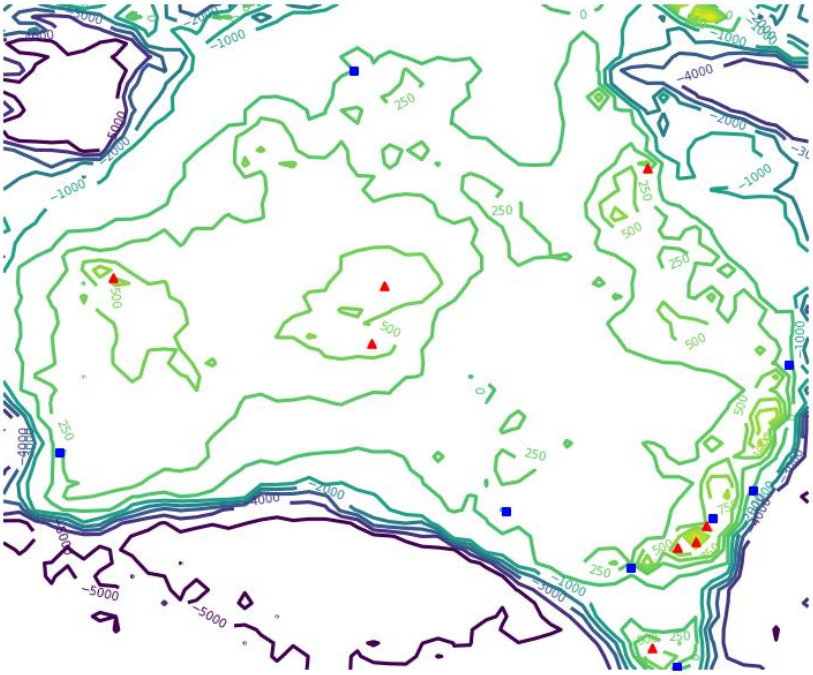


FIGURE 2.4: Elevation map of Australia. The blue squares mark the capital cities of the country, and the red triangles mark the highest point in each state and territory.

elevation and shown in Figure 2.5. Low-fidelity sources of this kind are often used in benchmarks in the literature. The second low-fidelity source is denoted f_l^{approx} and is generated with a simple procedure. For locations below sea-level (i.e. for $f_h(\mathbf{x}) < 0$), this approximation returns a value of 0 (i.e. $f_l^{approx}(\mathbf{x}) = 0$). Assuming that the elevation of the capital cities is known, at every location above sea-level the low-fidelity source provides an approximation of the elevation by taking a distance-based weighted sum of the elevation of the two nearest capital cities. This source is shown in Figure 2.6. For simplicity, here we assume that both low-fidelity sources are cheap to sample i.e. the cost ratio is $C_r = 0$ for both sources; the added effect of having a non-cheap source is discussed further down.

Let us now analyse different problem settings to illustrate the three problem types defined in Section 2.1. Assume first that the data has already been gathered i.e. engineering teams have already been sent to different locations in Australia, and queries have been made of the inaccurate records. Problem 1 defined in Section 2.1.1 corresponds here to the task of creating a model for the elevation of Australia based on this fixed sample. Two scenarios are compared. In the first one, 25 engineering teams have been sent to collect data, and 50 queries of the inexact records have been made. That is, 25 f_h samples and 50 f_l^{error} samples are available to train a model. In the second scenario, the

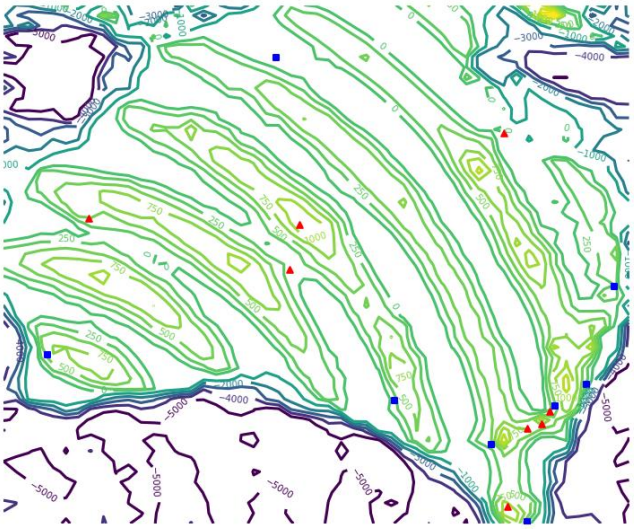


FIGURE 2.5: Low-fidelity source of the elevation of Australia, generated by adding a deterministic error to the true elevation. The blue squares mark the capital cities of the country, and the red triangles mark the highest point in each state and territory.

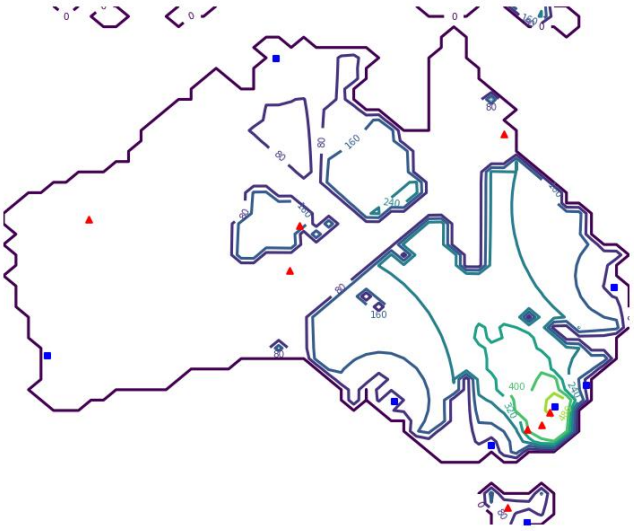


FIGURE 2.6: Low-fidelity source of the elevation of Australia, which is an approximation based on the known elevation of the capital cities. The blue squares mark the capital cities of the country, and the red triangles mark the highest point in each state and territory.

same amount of data of both kinds is available, but all f_l^{error} data comes from locations that are above sea level. This second scenario can be perhaps seen as more realistic if the source f_l^{error} is assumed to represent inaccurate records from city councils, which only exist on land. Figure 2.7 shows the locations at which these samples have been gathered for both scenarios. The accuracy of Kriging models trained only with high-fidelity data and Co-Kriging models trained with high- and low-fidelity data in both scenarios is shown in Table 2.1. In this example, it can be seen that many factors can impact the

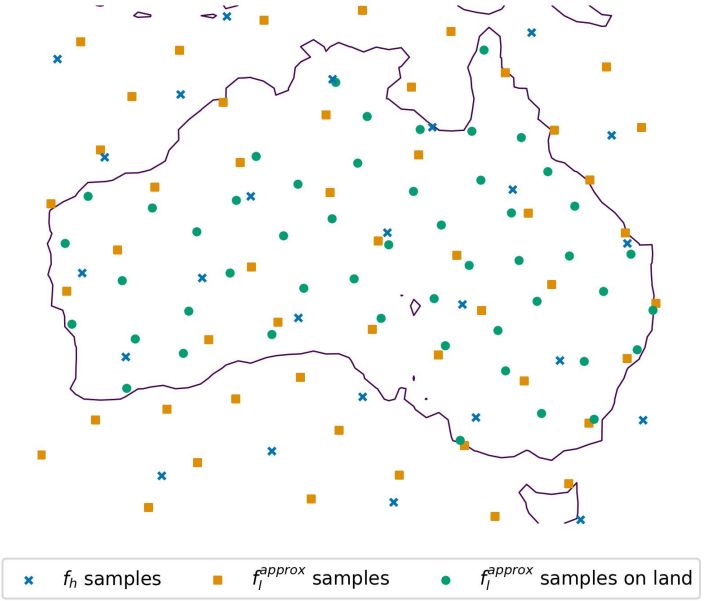


FIGURE 2.7: Locations at which samples have been gathered and made available to train a model. The blue crosses represent the 25 locations at which the high-fidelity source has been sampled. The orange squares represent 50 locations at which the low-fidelity source f_l^{approx} has been sampled, and are spread out across the whole space. The green circles represent 50 locations at which the low-fidelity source f_l^{approx} have been sampled; this last set of samples has been restricted to lie on locations above sea level, marked by the shown outline.

benefit of relying on low-fidelity sources. The fact that relying on data from f_l^{error} can be beneficial is represented by the decreased error of Co-Kriging models over Kriging models when this data has been collected evenly from the whole space. However, in the second scenario, Co-Kriging models have a larger error than Kriging models, despite the same amount of low-fidelity data having been collected from the same source. As the only difference in the second scenario is the fact that low-fidelity data is restricted to lie above sea level, it is clear that when choosing whether to rely on a low-fidelity source practitioners should carefully consider not only its quality but also the potential bias introduced by the sampling procedure.

Let us now assume that we have been given more freedom of action when generating a map of Australia, that is we are given the option to choose where to gather data from. The only given guideline is that we are restricted to sending a surveying team to at most 100 locations. This translates to Problem 2 defined in Section 2.1.2, namely model creation with a sample budget. To assess the value of the low-fidelity sources, three approaches are compared. The first is to rely only on high-fidelity data via the use of a Kriging model. The second is to work with a Co-Kriging model which relies on data

TABLE 2.1: Error of constructed models for 5 repetitions. The second column shows the error of Kriging models constructed using 25 high-fidelity samples spread out across the space. The third column shows the error of Co-Kriging models constructed with high-fidelity data and 50 samples from f_l^{error} spread out across the space. The fourth column shows the error of Co-Kriging models constructed with high-fidelity data and 50 samples from f_l^{error} spread out across locations above sea level.

Repetition	Kriging	Co-Kriging spread out	Co-Kriging above sea-level
1	1308.55	1180.69	1465.15
2	1294.4	1195.26	1378.99
3	1294.44	1129.29	1590.07
4	1552.9	1175.21	1611.1
5	1366.34	1183.36	1399.48

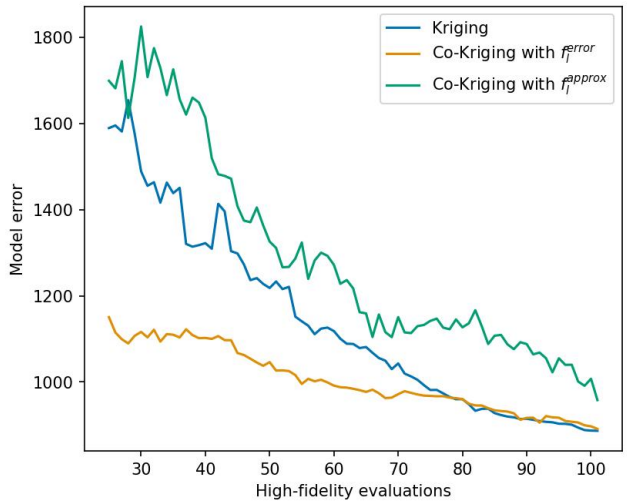


FIGURE 2.8: Average error of three surrogate models trying to model the elevation of Australia. The performance is shown for Kriging models which only use data from f_h (blue), Co-Kriging models which use data from f_h and f_l^{error} (orange), and Co-Kriging models which use data from f_h and f_l^{approx} (green)

from f_h and the low-fidelity source f_l^{error}. Finally, the third approach is to work with a Co-Kriging model which relies on data from f_h and the low-fidelity source f_l^{approx}. Following the design of experiments outlined in Section 2.2.3, a set of 50 locations is chosen via a locally optimised LHS plan at which f_l^{error} and f_l^{approx} are sampled, and a subset of 25 locations is chosen at which f_h is sampled. The relevant data is used to train the Kriging model and the two Co-Kriging models, and then further sampling locations are iteratively chosen by maximising the model uncertainty as described in Section 2.2.4. The process is stopped once the high-fidelity source has been sampled a total of 100 times. Repeating this approach 10 times and taking the average model error yields the results shown in Figure 2.8.

It can be seen here that the low-fidelity source f_l^{error} is an asset when attempting to

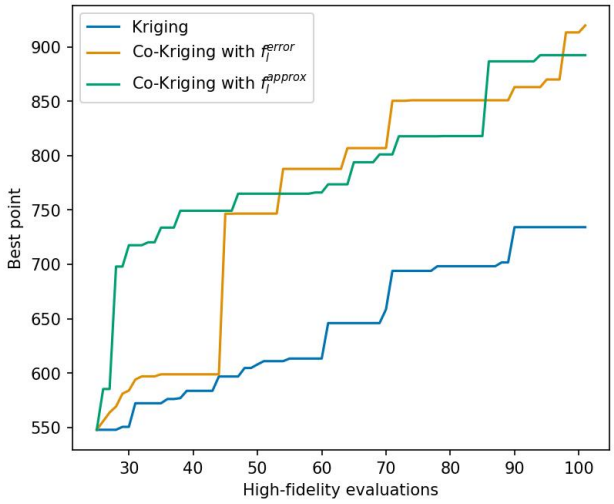


FIGURE 2.9: Average best point found during the optimisation of three surrogate modelling techniques. The performance is shown for Kriging models which only use data from f_h (blue), Co-Kriging models which use data from f_h and f_h^{error} (orange), and Co-Kriging models which use data from f_h and f_h^{approx} (green)

model the elevation of Australia, as Co-Kriging models that rely on this source are the most accurate on average. Classical examples of this type have often been used in the literature to motivate the development of multi-fidelity techniques such as Co-Kriging. It might be counter-productive to blindly rely on low-fidelity sources when they are available, however, as the same graph shows that Co-Kriging models which rely on the source f_l^{approx} perform worse than Kriging models which only rely on high-fidelity data. This can be unsurprising, given that the source f_l^{approx} only gives an extremely rough outline of the elevation. This example nonetheless demonstrates that one must be wary of always using data simply because it is available. It is also worth noting that when a lot of high-fidelity data is available (that is, towards the right of the graph), Kriging models and Co-Kriging models trained with f_l^{error} data show similar performance. The plot's horizontal axis is based only on high-fidelity evaluations since so far it has been assumed that the cost of sampling f_l^{error} was zero. It could be that this is not the case; assume for instance that obtaining the existing inaccurate elevation record of a particular location requires enquiring the archives, and this is roughly ten times cheaper than sending a team of engineers to the location. In this case, the cost ratio C_r of f_l^{error} would be equal to 0.1, and the total budget spent at any given moment is impacted by the number of low-fidelity samples taken. If this were the case, the orange line should be "shifted" to the right, meaning that for large budgets the best approach is to not rely on any low-fidelity data.

Let us now focus on an optimisation task. Assume that we have been tasked with finding the tallest point in Australia, once again with an overall budget of 100 samples. This translates to Problem 3 defined in Section 2.1.3, namely function optimisation with sample budget. Once again, the performance of Kriging models, Co-Kriging models trained with f_l^{error} data, and Co-Kriging models trained with f_l^{approx} data is compared using the same experimental setup. The only difference now is that further samples are iteratively chosen based on the expected improvement measure presented in Section 2.2.4. The results are shown in Figure 2.9. Perhaps unsurprisingly, relying on the source f_l^{error} is an improvement over using only high-fidelity data. Since this source helps generate more accurate models it seems a natural conclusion that it should help find better points during optimisation. What is perhaps unexpected is the fact that relying on the source f_l^{approx} leads to better points being found almost immediately, despite this source being counter-productive for surrogate model accuracy. This behaviour can be explained by the fact that the capital city Canberra is at an elevation of 554m. As this low-fidelity approximation is based on the elevation of the capital cities, and every other city is at sea level, the approximation leads to an almost flat objective function with only a high-elevation area in the bottom right of the map. This happens to coincide with the highest region of the true elevation of Australia. Because of this, Co-Kriging models trained with this data choose to sample in that region, leading to good points being found very quickly. Note that the low-fidelity source does not think that the area near Canberra is very high (as it thinks it has a maximum elevation of 500m), but it thinks it is the highest area, and this is sufficient. From a mathematical standpoint, this can be described as f_l^{approx} and f_h having a large difference, but being highly correlated in the highest regions of f_h. Once again, the fact that a low-fidelity source is harmful for model creation with a sample budget, but is beneficial for function optimisation with a sample budget, speaks to the complexity of characterising beneficial low-fidelity sources, and of creating algorithms that intelligently exploit or ignore these sources. Finally, note that none of these examples were the three sources used at the same time. The question of reliance on low-fidelity sources only grows in difficulty when more of them are made available at the same time.

2.4 Experimental studies

Despite the clear dangers of blindly trusting low-fidelity sources, most literature studies work with the assumption that multi-fidelity methods should always be applied to Mf-EBB problems, thus assuming a minimum threshold in the quality of f_l. In other words, if low-fidelity sources are available, they should be used, be it to train a surrogate model or when using optimisation algorithms. Toal [18] was one of the first to challenge this assumption by asking "when exactly should a designer select a multi-fidelity approach over a single fidelity approach and vice versa?" In his study, he focused on Bf-EBB problems by comparing the accuracy of (single-source) Kriging models with the accuracy of (two-source) Co-Kriging models for a variety of benchmarks (i.e. pairs of functions (f_l, f_h)) given a fixed sample.

As well as analysing the impact of the cost ratio and the relationship of the number of high- and low-fidelity function evaluations on model performance, he defined the correlation coefficient CC and error $RMSE$ measures

$$RMSE = \left[\frac{\sum_{i=1}^{n}(y_i^l - y_i^h)^2}{n}\right]^{1/2}, \tag{2.7}$$

$$CC = \left[\frac{1}{n-1}\left(\frac{\sum_{i=1}^{n}(y_i^l - \bar{y}_l)(y_i^h - \bar{y}_h)}{s_{y_l} s_{y_h}}\right)\right]^2, \tag{2.8}$$

$$\text{where} \qquad \bar{y}_l = \frac{1}{n}\sum_{i=1}^{n} y_i^l \qquad s_{y_l} = \left[\frac{\sum_{i=1}^{n}(y_i^l - \bar{y}_l)^2}{n-1}\right]^{1/2},$$

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

Both measures are calculated on a given set of sample points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \Omega$, with objective function values $\mathbf{y}_l = \{y_1^l, \ldots, y_n^l\}$ and $\mathbf{y}_h = \{y_1^h, \ldots, y_n^h\}$, with $y_i^l = f_l(\mathbf{x}_i)$ and $y_i^h = f_h(\mathbf{x}_i)$. The results of the experimental analysis conducted in this work led to the development of guidelines for the usage and sampling of low-fidelity sources. Given a total sampling budget of B and a cost ratio C_r, he defined the quantity $f_r = 1 - \frac{n_h}{B}$. As n_h represents the total number of high-fidelity samples, the value f_r denotes the proportion of the total high-fidelity budget devoted to sampling the low-fidelity sample. With this in mind, the provided guidelines include

1. The low-fidelity source should only be used if $CC \geq 0.9$.

2. At most 80% of the total budget should be spent sampling the low-fidelity source i.e. $f_r \leq 0.8$.

3. At least 10% of the total budget should be spent sampling the low-fidelity source i.e. $f_r \geq 0.1$.

4. There should always be more low-fidelity than high-fidelity samples, with a conservative bound of $f_r > \frac{1.75}{1+C_r}$.

Perhaps the first guideline has had the biggest impact in the literature, indicating that unless the low-fidelity source behaves similarly to the high-fidelity source, relying on it for surrogate model construction can be counter-productive. Note that crucially, in the study this measure is calculated with a large sample size irrespective of the actual sample used to train the models. This implies that the direct application of these guidelines in an industrial setting might be limited. As discussed in the beginning of this chapter, however, insights provided by studies of this kind can directly result in adaptive techniques which can be directly applied to industrial problems [22].

As stated in the Introduction, Toal's work highlights two important and interrelated questions pertaining to Bf-EBB problems in particular and Mf-EBB problems in general:

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

2. How can we develop an algorithm or rules to choose when to use a bi-fidelity or single source model?

Despite Toal's work bringing attention to these important questions, analysis of the performance of algorithms within the context of variations in low-fidelity function quality has only gained traction in very recent years. A survey by Giselle et al. [4] which focused on the topic of building multi-fidelity surrogate models for optimisation did not discuss the impact of function quality; nor did another survey by Liu et al. [53] which focused on sampling strategies leading to accurate multi-fidelity surrogate models. Giselle et al. [4] stated that "more research into the choice between high-fidelity alone surrogate and multi-fidelity surrogate is clearly called for". Whilst most studies properly justify the

choice of test benchmarks, an analysis of their diversity is often missing [38, 54–62]. This can lead to a choice of benchmarks for which the multi-fidelity algorithm performance is unsurprising and consistent with Toal's findings [18]: either performing poorly when f_l is chosen as an unrelated function [55], or performing better since f_l functions are frequently either chosen as highly correlated industry test problems [63] or created via a small perturbation of f_h [54].

Some very recent studies however extend the work of Toal when performing algorithm analysis. These studies analyse both the qualities of their chosen test benchmarks by measuring the correlation between f_h and f_l, and the overall discrepancies between the two, to assess whether a wide set of benchmarks have been chosen to conduct a thorough algorithm analysis [20, 22, 40, 64–66]. Some studies also provide a plot of the high-fidelity function values against the low-fidelity function values [67, 68] to analyse how well the low-fidelity source represents the high-fidelity source. The choice of testing benchmarks is an important decision in all experimental studies, particularly as the drawn conclusions could be affected by the way new methods are tested. Conducting an assessment that is both varied and bias-free is key to ensure any new insights are valid and applicable to other untested problems.

2.5 Conclusion

This chapter has provided an introduction to surrogate modelling techniques for Mf-EBB problems. The focus has been on the definition of different problem variations, and the use of traditional approaches such as training Co-Kriging models and using expected improvement as the acquisition function. As alluded to throughout the chapter, many new techniques have been created since the multi-fidelity data fusion of Kennedy and O'Hagan [37] was first proposed over 20 years ago. A key aspect of methods for Mf-EBB problems however which still seems to lack a satisfying level of understanding is how the quality of low-fidelity sources can impact multi-fidelity techniques, and how this impact can be mitigated. The elevation example in particular has helped to highlight that characterising and exploiting helpful low-fidelity sources is far from trivial. Indeed, relying on beneficial low-fidelity sources can become a sub-optimal choice if their domain

is restricted to certain regions in the space or their cost is too great, and harmful low-fidelity sources when constructing accurate surrogate models can be very beneficial when used during an optimisation process.

Strategic questions such as how to divide a total budget among an initial design of experiments, and how to further divide the sample among sources also still lack a consensus from the literature. These questions are particularly important for new techniques that seek to learn whether a source is helpful or harmful before relying on them, as it is crucial to strike the right balance between spending long enough correctly characterising a source and being left with enough budget to exploit this characterisation. The recent work of Toal [18] on the analysis of low-fidelity source quality impact on Co-Kriging model performance has shown this area to require further study, particularly as it can provide insights into the creation of new adaptive techniques. The remainder of this work seeks to further explore this topic. The next chapter will begin this exploration by analysing the choice of benchmarks made by studies in the literature, and whether performing the same analysis with different benchmarks might lead to different results.

Chapter 3

Showcasing the need for new test instances

As stated in the preface, the content of this chapter constitutes an adaptation of the published paper "Bi-fidelity Surrogate Modelling: Showcasing the Need for New Test Instances". The introduction and conclusion have been adapted from this paper to fit the thesis format.

The previous chapter has highlighted both the risks of blindly trusting low-fidelity sources without conducting a prior assessment, and the importance of the chosen testing benchmarks (henceforth referred to as instances) when analysing new methods. This chapter argues the need for further work in both instance creation and instance characterisation in the field of Bf-EBB. Both are important prerequisites before analysing for what types of instances can a two-source algorithm be used, and to develop prediction techniques which can correctly choose which method to use. As such, their importance will be showcased through the analysis of single-source vs two-source surrogate model accuracy in the form of Kriging vs Co-Kriging, similar to the work of Toal [18]. Performing this analysis twice using different instance test suites will reveal the need for new instances and new instance measures, or features, both of which are put forward in this work.

The remainder of this chapter is structured as follows. Section 3.1 presents the existing types of instances and features in the literature, chooses over 200 instances to create a literature test suite, and conducts a performance analysis of Kriging and Co-Kriging

algorithms using this test suite. Section 3.2 then proposes new features and instances which are used to create an augmented test suite. This new test suite is used for a second algorithm performance analysis, leading to results in direct disagreement with those found in Section 3.1. Section 3.3 presents a discussion of the differences in results obtained based on the chosen test suite, and highlights the importance of the presented features and instance creation procedure. Section 3.4 concludes the chapter with some closing remarks. In order to make the results easily reproducible, both the code and the data used in the research at the time of publishing are available on the Informs Journal On Computing GitHub site [69].

3.1 Existing test suites

As presented in Chapter 2, Bf-EBB problem instances are defined in part by a pair of functions (f_l, f_h). Many such pairs exist in the literature, as studies presenting new techniques often perform their experimental analysis using newly created instances. The standard approach in creating such a pair of functions is the use of a well-known test function such as the Branin or Hartmann functions being assigned to f_h, with f_l being defined as a modification of f_h via the addition of some extra terms or the modification of the coefficients of f_h. Three types of literature instances, namely *fixed*, *parameter-based* and *error-based* instances, are implemented in this study to create a literature test suite. All of the implemented instances are provided in Appendix C.

The term fixed can be used to describe the most common type of instance found in the literature. Each of these instances is the result of some modification being applied to f_h resulting in a unique (f_l, f_h) pair, as opposed to a "family" of instances as is the case in the other two classes described below. How this modification is added can greatly vary from study to study, from adding a function to f_h to represent a small error term [54] to assigning an unrelated function to f_l [55]. A total of 43 distinct functions of this type with varying dimension ($d \in \{1, 2, 3, 4, 5, 6, 8, 10, 20\}$) from ten different studies [54–56, 58, 60, 64, 70–73] are implemented here.

The set of parameter-based instances is of the kind first presented by Toal [18], which is followed in the work of Song et al. [40]. This type of instances is defined as a "family" of instances by defining a fixed f_h function and a parametrised f_l function. By varying

the single parameter A of f_l, new (f_l, f_h) pairs are created with varying f_l quality. An example is Toal's Branin function, defined for $\Omega = [-5, 10] \times [0, 15]$,

$$f_h(\mathbf{x}) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10,$$
$$f_l(\mathbf{x}) = f_h(\mathbf{x}) - (A + 0.5)(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 \qquad A \in [0, 1].$$

By varying the parameter A with the values $\{0, 0.1, \ldots, 0.9, 1.0\}$, 11 different instances can be generated from a single definition. A further 77 such instances with different dimensions ($d \in \{1, 2, 4, 10\}$) are added to the literature test suite (7 "families" of instances with 11 different A values each).

Finally, another set of instances defined by Wang et al. [20] are also added to the literature test suite. While studying an evolutionary algorithm for multi-fidelity optimisation, the authors argued the need for new test instances for these types of algorithms, and provided a systematic way to create a (f_l, f_h) pair by adding so-called stochastic, instability or resolution errors to f_h. Neither stochastic nor instability errors are implemented in this study, as the former define non-deterministic functions, and the latter define functions that might fail to evaluate. On the other hand, resolution errors are deterministic and therefore highly relevant to this study. Resolution errors are used to create a framework for instance creation based on industry problems where the low-fidelity function f_l is taken to represent a simulation model, with its accuracy and cost able to be varied by changing the resolution of the model. As such, given a function f_h, f_l is created by adding one of four resolution errors with parameter $\phi \in [0, 10000]$. The authors showcase their method with the Rastrigin function, for which a desired function degree d can be specified. Setting the dimension to 5 and 10 in combination with all four resolution errors and for ϕ values $\{0, 1000, \ldots, 9000, 10000\}$, another 88 instances are added to the literature test suite, ultimately containing a total of 208 instances already studied in the literature.

3.1.1 Existing features

As discussed in Chapter 2, the work of Toal [18] was one of the first to bring attention to the impact that different instance qualities can have on the performance of a given algorithm. His work presented two measures (here denoted as *instance features*) for the analysis of pairs of functions (f_l, f_h), namely CC and RMSE, both of which are defined in Equations 2.8 and 2.7, respectively. Note that these two measures are calculated after heavily sampling both f_h and f_l. In this study, a sample of size $1000d$ is chosen via LHS. Taking such a large sample cannot be done in practice, however, as the sample budget is limited. The aim here is to study the effect of different instance features on algorithm performance in an artificial setting, where the analytical expressions for both f_h and f_l are known and sampling them can be done at little to no cost. Once the features have been measured, these functions are treated as expensive black-box functions when assessing algorithm performance.

Both features are calculated on a given set of sample points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \Omega$, evaluated both at f_l and f_h. Therefore, the two available sets are $\mathbf{y}_l = \{y_1^l, \ldots, y_n^l\}$ and $\mathbf{y}_h = \{y_1^h, \ldots, y_n^h\}$, with $y_i^l = f_l(\mathbf{x}_i)$ and $y_i^h = f_h(\mathbf{x}_i)$. Note that Toal's CC feature is defined as the square of the sample correlation of $\mathbf{y}_l$ and $\mathbf{y}_h$. As such, $0 \leq CC \leq 1$, and a high CC value indicates f_l is correlated with f_h and therefore behaves similarly. Note also that the $RMSE$ feature can vary greatly between instances as it is affected by the range of f_h and f_l. As such, in this study the Relative $RMSE$ ($RRMSE$) feature will be used instead to allow comparison between instances

$$RRMSE = \frac{RMSE}{\max\{\mathbf{y}_h\} - \min\{\mathbf{y}_h\}}. \tag{3.1}$$

Figure 3.1 shows a plot of all 208 instances of the literature test suite based on their CC and $RRMSE$ feature values.

3.1.2 Algorithm performance analysis on literature test suite

The created literature test suite can be used to assess the comparative performance of Kriging and Co-Kriging (see Appendix A) using an experimental setup similar to that of Toal [18]. For a given instance of dimension d, a total sample budget of $5d$ is used

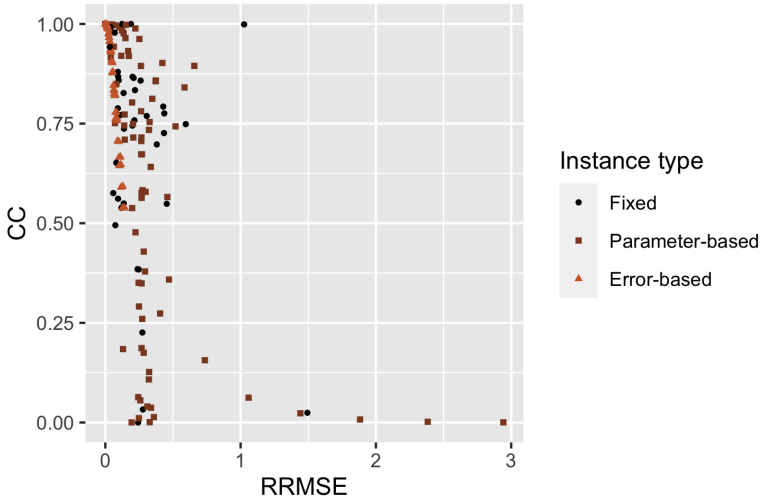


FIGURE 3.1: Instances in the literature test suite plotted based on their Correlation Coefficient (CC, y-axis) and Relative Root Mean Squared Error ($RRMSE$, $\mathbf{x}$-axis) values. Each plot point represents an instance i.e. a pair of functions (f_l, f_h), and a set of $1000d$ points is used to calculate the features of each instance, where $1 \leq d \leq 20$

for both Kriging and Co-Kriging. In the case of Kriging, the whole budget is used to sample f_h; that is, a sample of f_h of size $5d$ is created using LHS and then used to fit the surrogate model. In the case of Co-Kriging, a budget of d samples is used to sample f_l, with the remaining $4d$ used to sample f_h. These samples are chosen using first LHS, and then a subset f_h sample is chosen using the procedure provided in Section 2.2.3. Toal analyses algorithm performance for different cost ratio C_r values; in this study, however, a value $C_r = 0.1$ is fixed, leaving further analysis of the impact of this constant to future work. Therefore, for an instance of dimension 5, a Kriging model is fitted using 25 samples of f_h, and a Co-Kriging model is fitted using 20 samples of f_h and 50 samples of f_l.

Note here that this experimental setup does not fit the problem definitions of Model creation with fixed sample or Model creation with sample budget given in Section 2.1. In a sense, the comparison being carried out is that of two hypothetical scenarios, each differentiated by the user's choice of model despite having no prior knowledge of the information sources. In the first, a Kriging model is trained and thus the whole sampling budget is spent gathering high-fidelity data. In the second, a Co-Kriging model is trained and thus the sampling budget is divided between the collection of high- and low-fidelity data. We believe that if the user has this amount of freedom, there should be room for the use of an acquisition function as is explored in Chapter 5. If this freedom does not exist and the data has already been collected, the same amount of high-fidelity data

should be used when training both the Kriging and Co-Kriging models, as is explored in Chapter 4. Nonetheless, as this chapter does not aim to develop new guidelines but simply to illustrate the need for new instances and features, the setup of Toal is closely followed here.

The performance of the two techniques on a given instance is compared by assessing the statistical performance of 20 runs per algorithm. All experiments are run using the Spartan computer cluster (see Preface). The performance of an algorithm on a single run is measured in the form of the error between the constructed surrogate model and the function f_h. This is measured by taking $1000d$ samples of both the surrogate model and f_h, and calculating the $RRMSE$ value between the two samples. The 20 runs of each algorithm are then compared using a two-tailed Wilcoxon test [74], with the null hypothesis being that the two sets of errors are not statistically significantly different from one another. The two algorithms are said to perform equally well if the null hypothesis cannot be rejected with higher than 95% accuracy. If the null hypothesis is rejected, the algorithm with the smaller median error is said to perform best. A plot representing which algorithm performs best for each instance is shown in Figure 3.2, which seems to indicate that for $CC \leq 0.5$, Co-Kriging is ill-advised. This supports the findings of Toal [18] regarding the strong impact the feature CC has on algorithm performance, despite a discrepancy on the critical CC value at which Co-Kriging is discouraged. Toal concluded that Co-Kriging should be used for $CC \geq 0.9$ whereas Figure 3.2 seems to indicate that Co-kriging could be used for $CC \geq 0.5$, although this difference may be due to the choice of C_r and budget in this study.

To further evaluate the impact of different features on algorithm performance, a machine learning method is used to infer when each algorithm is expected to perform better than the other. The question being answered is when to ignore the low-fidelity source of information as harmful, and use Kriging instead of Co-Kriging. Therefore, the prediction accuracy is assessed in terms of the accuracy predicting for which instances Co-Kriging can be used (i.e. Co-Kriging performs no worse than Kriging, shown as green triangles and yellow circles in the plots), and for which it cannot be used (i.e. Kriging outperforms Co-Kriging, red squares in the plots). The literature test suite is randomly divided into a training and a testing set, the former containing 80% and the latter 20% of the data. A decision tree is trained, with its accuracy assessed on the testing set. This cross-validation procedure is repeated 100 times for each technique and the average accuracy

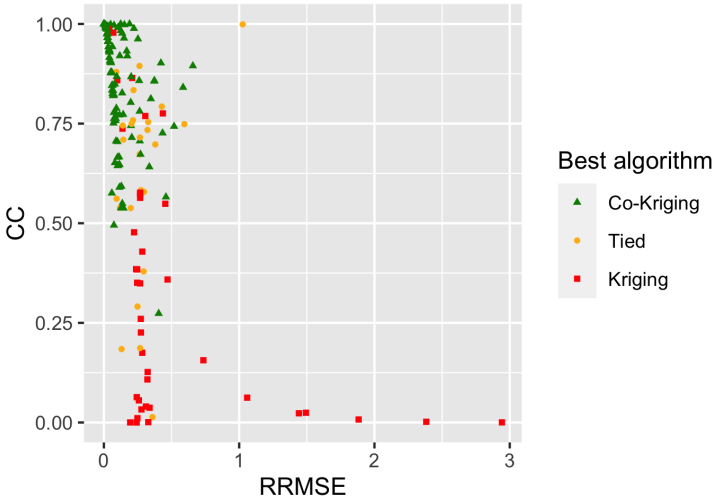


FIGURE 3.2: Comparative performance of Kriging and Co-Kriging on the literature test suite. Each plot point represents an instance i.e. a pair of functions (f_l, f_h) and its position is based on the features Correlation Coefficient (CC, y-axis) and Relative Root Mean Squared Error ($RRMSE$, x-axis). The colour/shape represents which technique performed best, either Kriging (red square), Co-Kriging (green triangle) or statistically equal performance (yellow circle).

is calculated.

Two basic prediction rules are used as baseline comparisons. The first is the simple rule to always use Co-Kriging. The second is a majority rule; that is, the same algorithm is always used based on the one that has superior accuracy on the training data. For example, if Co-Kriging should only be used in 30% of the instances in the training set, the majority rule chooses to always use Kriging on the testing set. These two baselines are compared to a set of decision trees built using varying sets of features. These are constructed using the rpart package in R, with parameters $cp = 0$ and $minbucket = 0.05$. This allows the constructed trees to keep growing as long as prediction error is reduced, while restricting the leaves of the tree to contain at least 5% of the instances in the training set. Two sets of prediction trees are compared, the first built using only the feature CC, and the second using all the literature features, namely CC, $RRMSE$, problem dimension and problem budget. The resulting prediction accuracies are shown in Table 3.1. Interestingly, creating a decision tree with only the CC feature or with all literature features results in almost equally accurate trees.

Finally, a decision tree is built using all literature features with the whole literature test suite as the training set, as shown in Figure 3.3, to further analyse the impact of different features on accuracy. This final tree states that Co-Kriging should be used only when

TABLE 3.1: Cross-validated accuracies of different techniques predicting when to use Co-Kriging on the literature test suite. The techniques include always using Co-Kriging, a majority rule, and the accuracies of constructing a decision tree using only the CC feature, or using all of the literature features, that is $RRMSE$, CC, problem dimension and problem budget.

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

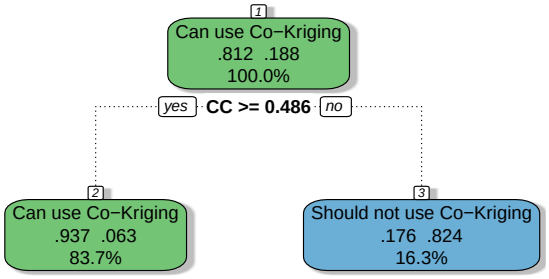


FIGURE 3.3: Decision tree constructed using the literature features and the performance of Co-Kriging and Kriging on the literature test suite. The label "Can use Co-Kriging" means the tree predicts Co-Kriging performs no worse than Kriging. The label "Should not use Co-Kriging" means the tree predicts Co-Kriging performs worse than Kriging. The percentages indicate the instances present in the node, and the proportions indicate the proportion of the instances in the node for which Co-Kriging can be used (left) and for which it should not be used (right).

$CC \geq 0.486$ (93.7% true positive rate, 82.4% true negative rate, 91.9% accuracy), which would be an improvement over the choice to always use Co-Kriging (81.5% accuracy). Despite being allowed to grow unrestricted (by setting $cp = 0$), the tree is quite small. It is however possible that features other than CC could have been used to produce a different tree with similar accuracy. In order to get a sense of the relevance of each of the features, the "variable importance" value of the decision tree is inspected. This value indicates the importance of each of the features when constructing the tree, as it indicates the reduction in prediction error obtained when splitting nodes in the tree using a particular feature. The variable importance is measured in each of the 100 runs when constructing a decision tree using all literature features, and the average is shown in Figure 3.4.

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

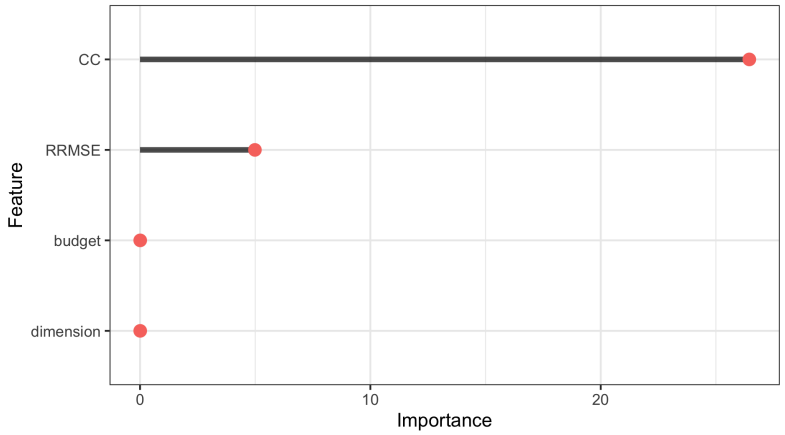


FIGURE 3.4: Average variable (feature) importance of 100 runs when constructing a decision tree using all the literature features on a training dataset chosen from the literature test suite.

- The constructed literature test suite appears to be varied enough, as seen in Figure 3.1, and therefore they present a good test of algorithm performance.
- Co-Kriging is overall a "safe" technique when applied to Bf-EBB problems, as it performed no worse than Kriging in 81.5% of the instances analysed, as shown in Table 3.1.
- The feature CC is sufficient to accurately predict when Co-Kriging can be used, and the use of other features only results in a marginal improvement on this accuracy, as shown in Table 3.1. Figure 3.4 shows that this feature has a much higher importance than the other features when constructing a decision tree.
- For problems with a total budget of $5d$ and a cost ratio $C_r = 0.1$, it is beneficial to use f_l (through Co-Kriging) for $CC \geq 0.486$, as shown in Figure 3.3.

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

3.2 Creation of new test suites

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

3.2.1 New features

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

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

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

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

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

Note that if $w_i = 1$ for all i, $WCC = CC$. Given the samples $\mathbf{y}_l$ and $\mathbf{y}_h$, we define the Local Correlation Coefficient (LCC) at a point $\mathbf{x}$ with radius r as

$$\begin{aligned} LCC^r(\mathbf{x}) &= WCC(\mathbf{w}), \\ \text{where} \qquad w_i &= \max\{0, 1 - \frac{\|\mathbf{x} - \mathbf{x}_i\|}{r\|\mathbf{x}^\top - \mathbf{x}^\perp\|}\}. \end{aligned}$$

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

$$LCC_p^r = \frac{|\mathcal{L}_p^r|}{|\mathcal{L}^r|}. \tag{3.2}$$

Here $\mathcal{L}_p^r = \{LCC^r(\mathbf{x}) \in \mathcal{L}^r \quad | \quad LCC^r(\mathbf{x}) \geq p\}$ is the set of local correlations with a value no lower than p. The family of features LCC_p^r provides an approximation of the probability that a randomly chosen point in Ω has a local correlation no lower than p. The set $\mathcal{L}^r$ can be used to calculate two additional features which represent the variability in local correlation. These are the sample standard deviation LCC_{sd}^r and coefficient of variance LCC_{coeff}^r of $\mathcal{L}^r$ and are given by the formula

$$LCC_{sd}^r = \sqrt{\frac{\sum_{i=1}^n \left[LCC^r(\mathbf{x}_i) - \overline{LCC}^r\right]^2}{n-1}}, \tag{3.3}$$

$$LCC_{coeff}^r = \frac{LCC_{sd}^r}{\overline{LCC}^r}, \tag{3.4}$$

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

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

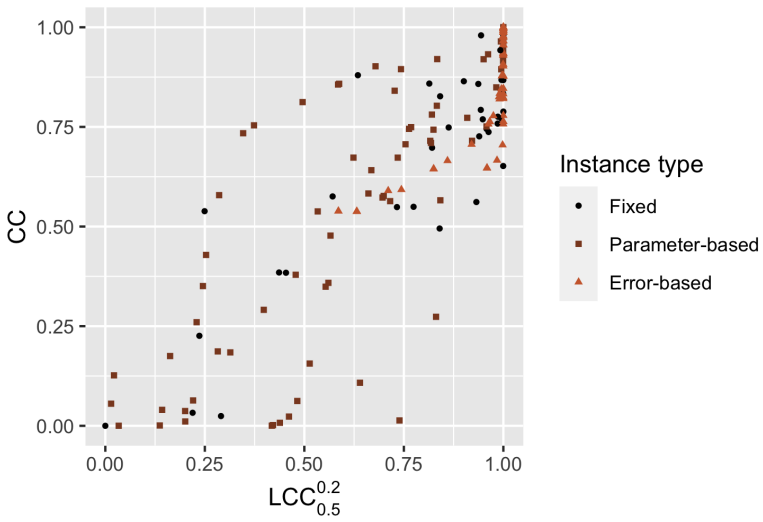


FIGURE 3.5: Literature instances plotted using a feature from the literature (CC, y-axis) and a proposed new feature ($LCC_{0.5}^{0.2}$, x-axis). Each plot point represents an instance i.e. a pair of functions (f_l, f_h). Note that the instances have similar values for both features (the relationship is strongly linear), and the top left and bottom right quadrants are relatively empty.

The previous section demonstrated that for a budget of $5d$ and cost ratio $C_r = 0.1$, the correlation threshold for which Co-Kriging can be used appears to be $CC > 0.486$. Therefore, the feature $LCC_{0.5}^{0.2}$ is of particular interest as it uses a threshold of 0.5 when assessing whether a local correlation is high enough to be considered "good". This feature is used to replot the literature test suite instances, resulting in Figures 3.5 and 3.6. As can be seen in both figures, despite containing a large set of instances, the literature test suite only occupies limited regions of the sample space, with most instances having very similar CC and $LCC_{0.5}^{0.2}$ values. Figure 3.5 shows that error-based instances suffer the most from this, as most of them have very high CC and $LCC_{0.5}^{0.2}$ values. Parameter-based instances provide some variation within the test suite, although most instances still lie in the top-right and bottom-left regions. Figure 3.6 shows why this could be a problem. If the space is divided into four quadrants, the top-right is mainly filled with instances where Co-Kriging performs well, whereas the bottom-left region is filled with instances where it does not. As the other two regions are relatively empty, however, it is hard to assess whether Co-Kriging performs well for instances with a high CC value or for instances with a high $LCC_{0.5}^{0.2}$, and how it will perform with instances which lie in the empty regions. To analyse the causes for the observed results, new instances that further fill the space are required.

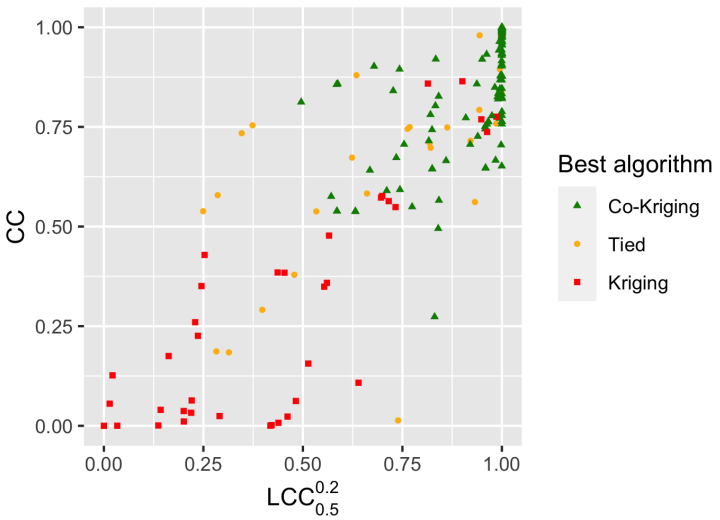


FIGURE 3.6: Comparative performance of Kriging and Co-Kriging on the literature test suite, plotted using a feature from the literature (CC, y-axis) and a proposed new feature ($LCC_{0.5}^{0.2}$, x-axis). Each plot point represents an instance i.e. a pair of functions (f_l, f_h), and the colour/shape represents which technique performed best, either Kriging (red square), Co-Kriging (green triangle) or statistically equal performance (yellow circle). Note that the top-right quadrant is filled with instances where Co-Kriging performs well, the bottom-left is filled with instances where it does not, and the top-left and bottom-right regions are almost empty.

3.2.2 Generating new instances

A new instance generating procedure is presented here. The aim is two-fold. Firstly, the procedure must generate artificial instances which can represent industry problems. This is needed as algorithms being tested will eventually be applied to real-world problems, and therefore conclusions being drawn from the use of a test suite must be applicable to industry. Secondly, to fill the space shown in Figure 3.5, instances are needed for which the quality of f_l changes throughout the domain Ω. This differs from existing literature instances, which have a f_l function with a fixed quality throughout the whole domain.

The proposed procedure for constructing the low-fidelity function f_l consists of two steps. A "basic" disturbance is first defined which can be added to the high-fidelity function, similar to the error term used by Wang et al. [20]. A set of modifications are then defined which can be applied to a basic disturbance so that its impact on the function varies throughout Ω. A single basic disturbance is used in this work, namely

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

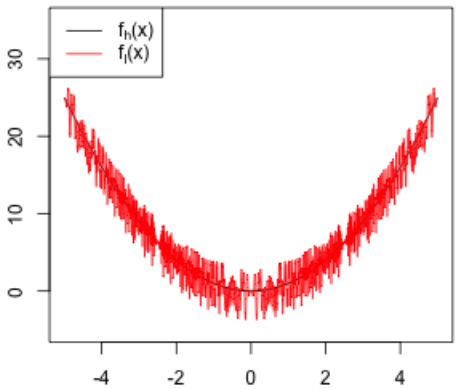


FIGURE 3.7: Basic disturbance when added to the function $f_h(x) = x^2$ (black) with $\alpha = 0.3$ and $\nu = 10$ to create a $f_l(x) = f_h(x) + \text{dist}(x, \alpha, \nu)$ function (red).

The disturbance is defined relative to a given high-fidelity function, with $f_h : \Omega \to [f_{min}, f_{max}] \subset \mathbb{R}$ being bounded by the hypercube $\Omega = [x_1^{\perp}, x_1^{\top}] \times \cdots \times [x_d^{\perp}, x_d^{\top}]$. The constant α modulates the amplitude of the disturbance, and ν the frequency of the trigonometric functions. Figure 3.7 shows the creation of a function f_l via the addition of the disturbance to f_h. It is important to note that a single basic disturbance is used in this study to showcase how basic disturbances can be modified to generate novel instances that vary greatly from one another. The modifications presented below can be applied to any basic disturbance, including stochastic disturbances such as White Noise, if the creation of a stochastic function is desired.

3.2.2.1 Height-based disturbance

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

$$\text{Dist}_1^h(f(\mathbf{x}), h, r) = M \cdot dist,$$
$$\text{with } M = \max\{0, 1 - \frac{|\frac{f(\mathbf{x}) - f_{min}}{f_{max} - f_{min}} - h|}{r}\}.$$

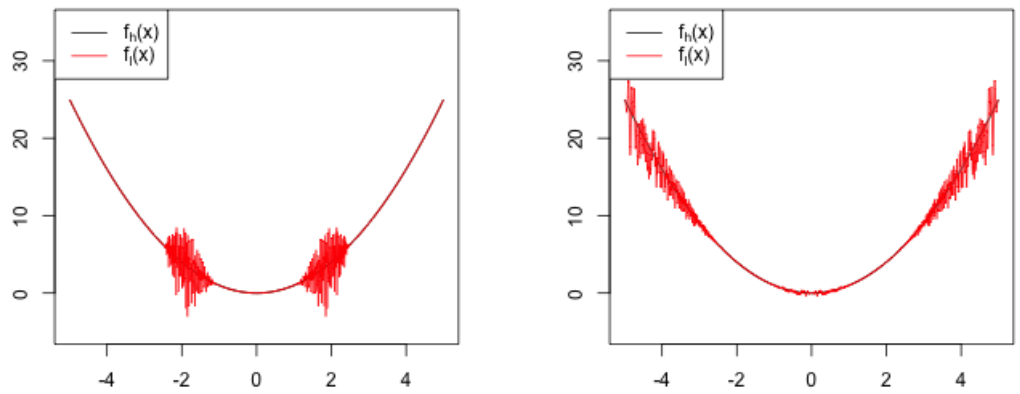


FIGURE 3.8: Height based disturbances added to the function $f_h(x) = x^2$ (black) to create a function $f_l = f_h(x) + \text{Dist}_i^h(f_h(x), h, r)$ (red). The modification Dist_1^h (left) adds the disturbance around a particular height, whereas the modification Dist_2^h (right) removes the disturbance around a particular height. The parameters used are $h = 0.15$ and $r = 0.1$, with a basic disturbance defined with $\alpha = 0.3$ and $\nu = 10$.

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

$$\text{Dist}_2^h(f(\mathbf{x}), h, r) = M \cdot dist,$$
$$\text{where } M = \max\{0, 1 - \frac{1 - |\frac{f(\mathbf{x}) - f_{min}}{f_{max} - f_{min}} - h|}{1 - r}\}.$$

Here r and h are similarly defined. This modification can be used to create a function f_l representing a model constructed for a particular output range in which the output is accurate, beyond which it starts to lose accuracy. Both of these modifications can be added to a f_h function to create a f_l function as shown in Figure 3.8.

3.2.2.2 Location-based disturbance

Similarly to height-based disturbances, location-based disturbances are a way to modulate the disturbance based on the sample location of f_h, with locations defining sources

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

$$\begin{aligned} \mathrm{Dist}_1^s(\mathbf{x}, L, r) &= M \cdot dist, \\ \text{with } M &= \max\{0, 1 - \frac{1 - \frac{\min\{\|\mathbf{x}-\mathbf{x}_i^s\|\}}{\|\mathbf{x}^\top - \mathbf{x}^\perp\|}}{1-r}\}. \end{aligned}$$

The second location based disturbance is defined as a disturbance near the locations, with the disturbance disappearing after a certain radius

$$\begin{aligned} \mathrm{Dist}_2^s(\mathbf{x}, L, r) &= M \cdot dist, \\ \text{with } M &= \max\{0, 1 - \frac{\min\{\|\mathbf{x} - \mathbf{x}_i^s\|\}}{r\|\mathbf{x}^\top - \mathbf{x}^\perp\|}\}. \end{aligned}$$

Here $r \in [0, 1]$ again denotes the relative radius of the disturbance. Both of these modifications can also be added to a f_h function to create a f_l function as shown in Figure 3.9.

3.2.3 Augmented test suite construction

The newly defined instance generating procedure is used to create a large set of instances from which an augmented test suite can be constructed. The COCO test suite [75] (defined in Appendix C), which is composed of 24 generating functions that can be instantiated through translation and rotation, is used for this purpose. Therefore, 75 f_h functions are created by taking the first instantiation from functions $\{1, \ldots, 20\}$ with dimensions 1 (except for functions 8, 9, 17, 18 and 19 which are defined for $d \geq 2$), 2, 5 and 10. Each f_h function is then used to construct a set of (f_l, f_h) pairs via the addition of disturbances $\mathrm{Dist}_1^h, \mathrm{Dist}_2^h, \mathrm{Dist}_1^s$ or Dist_2^s, using values $h \in \{0, 0.25, 0.5, 0.75, 1.0\}$, $r \in \{0.025, 0.05, 0.1, 0.15, 0.2, 0.25\}$, a set L with 1, 3, 6 or 9 randomly chosen locations, and a basic disturbance with $\alpha \in \{0.1, 0.5, 1.0, 1.5, 2.0\}$ and $\nu \in \{10, 100\}$. This leads to the construction of 81000 instances, of which a subset is chosen that possesses a variety of

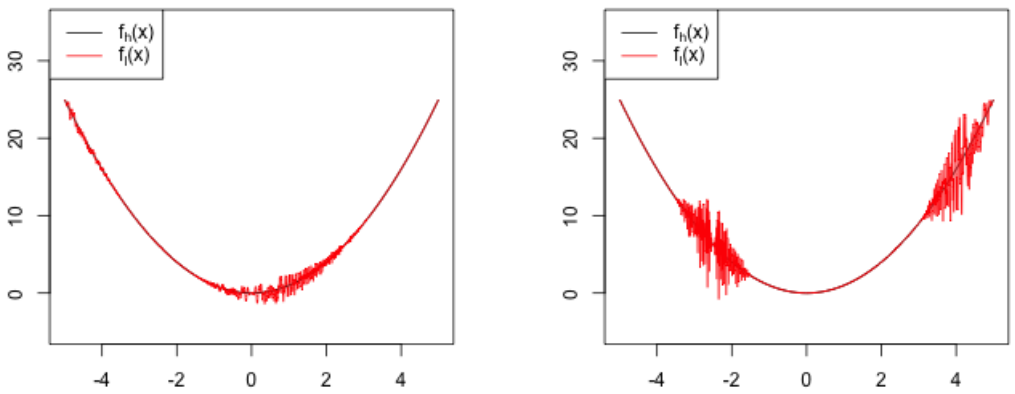


FIGURE 3.9: Location based disturbances added to the function $f_h(x) = x^2$ (black) to create a function $f_l = f_h(x) + \text{Dist}_i^h(x, L, r)$ (red). The modification Dist_1^s (left) removes the disturbance around the specified locations, and the modification Dist_2^h (right) adds the disturbance around the locations. The parameters used are $r = 0.1$ and $L = \{(-2.5), (4.1)\}$, with a basic disturbance with $\alpha = 0.3$ and $\nu = 10$.

CC and $LCC_{0.5}^{0.2}$ values. This is achieved by choosing, when possible, a single instance i of dimension d satisfying $0.05n - 0.05 \leq CC(i) \leq 0.05n$ and $0.05m - 0.05 \leq LCC_{0.5}^{0.2}(i) \leq 0.05m$ for $n, m \in \{1, 2, \ldots, 20\}$ and $d \in \{1, 2, 5, 10\}$. The chosen subset of instances is added to the literature test suite to construct the augmented test suite.

The set of 81000 instances as well as the augmented test suite are shown in Figure 3.10. As can be seen on the left plot, despite using a single basic disturbance, the use of the different modifications allows for the creation of instances with a large variation in CC and $LCC_{0.5}^{0.2}$ values, which showcases the procedure's effectiveness in more diverse instance creation. It is also worth noting that despite generating many more instances in the top-left and bottom-right quadrants than the literature test suite, there are still some regions with no instances. This makes sense as they represent extreme regions where instances have either very large CC and very low $LCC_{0.5}^{0.2}$ values, or very low CC and very large $LCC_{0.5}^{0.2}$ values. Whilst such instances could be created, they are not particularly practical, and we consider the selected augmented test suite to be sufficiently diverse for this study.

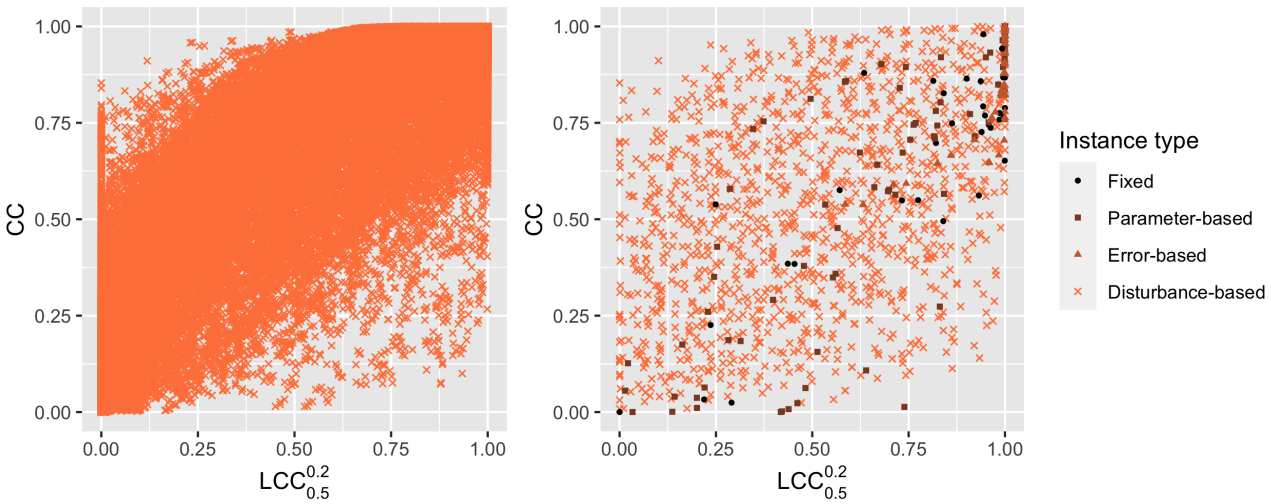


FIGURE 3.10: Set of 81000 newly generated instances (left) and selected augmented test suite (right), plotted using a feature from the literature (CC, y-axis) and a proposed new feature ($LCC_{0.5}^{0.2}$, x-axis). Each plot point represents an instance i.e. a pair of functions (f_l, f_h). The addition of the chosen subset to create the augmented test suite results in a much more varied test suite.

3.2.4 Updated algorithm performance analysis

The algorithm performance analysis presented in Section 3.1.2 is repeated with the augmented suite and additional features. Both Kriging and Co-Kriging are again used to construct surrogate models a total of 20 times each for each instance, and their comparative performance is assessed using a two-tailed Wilcoxon test. The top-performing algorithm for each of the instances is shown in Figure 3.11. Interestingly, it appears at first glance that the feature CC no longer provides an accurate means to decide when Co-Kriging will perform no worse than Kriging as there appears to be no horizontal line that could be drawn to separate the instances shown as red squares from the instances shown as green triangles and yellow circles.

Further analysis is conducted to assess which features have the biggest impact on algorithm performance. The same prediction techniques are used to predict when Co-Kriging will perform no worse than Kriging, including the baselines "always use Co-Kriging" and a majority rule. Three sets of decision trees are built, the first using only the CC feature, the second using all literature features (CC, $RRMSE$, budget, problem dimension), and the third using literature and new features (CC, $RRMSE$, budget, problem dimension, $LCC_{sd}^{0.2}, LCC_{coeff}^{0.2}, LCC_{0.1}^{0.2}, \ldots, LCC_{0.9}^{0.2}, LCC_{0.95}^{0.2}, LCC_{0.975}^{0.2}$). The resulting cross-validated accuracies, averaged over 100 runs, are shown in Table 3.2 for both the literature and the augmented test suites. The second column presents a strong contrast

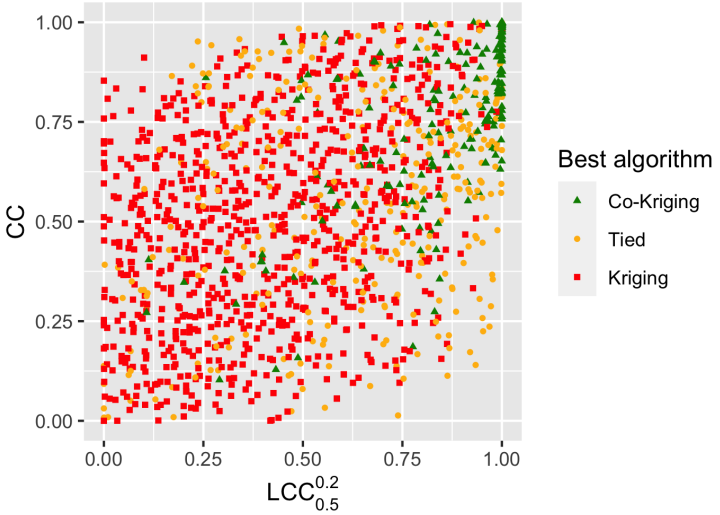


FIGURE 3.11: Comparative performance of Kriging and Co-Kriging on the augmented test suite. Each plot point represents an instance i.e. a pair of functions (f_l, f_h) and its position is based on the features Correlation Coefficient (CC, y-axis) and Local Correlation Coefficient with threshold 0.5 ($LCC_{0.5}^{0.2}$, x-axis). The colour/shape represents which technique performed best, either Kriging (red square), Co-Kriging (green triangle) or statistically equal performance (yellow circle). Note that $LCC_{0.5}^{0.2}$ appears to have a bigger impact on performance than CC.

TABLE 3.2: Accuracies of different techniques when predicting when to use Co-Kriging on the literature test suite, and the augmented test suite. The techniques include always using Co-Kriging, a majority rule, and the accuracies of constructing a decision tree using only the CC feature, using all literature features (CC, $RRMSE$, budget, problem dimension), or using literature and newly proposed features (CC, $RRMSE$, budget, problem dimension, $LCC_{sd}^{0.2}, LCC_{coeff}^{0.2}, LCC_{0.1}^{0.2}, \ldots, LCC_{0.9}^{0.2}, LCC_{0.95}^{0.2}, LCC_{0.975}^{0.2}$).

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

to the conclusions drawn in Section 3.1.2. Unlike with the literature test suite, with the augmented test suite it is best to choose always to use Kriging (the choice of the majority rule), as choosing to use Co-Kriging chooses the right technique only 38.8% of the time. It also appears that the feature CC no longer provides a very accurate indication of which technique will perform best. The decision trees constructed using only this feature performed remarkably worse than the ones constructed using all literature features, and a tree constructed using all features provides a further improvement in accuracy.

The importance of variables in the decision trees is again analysed to examine the impact

of each feature on tree accuracy. Figure 3.12 shows a final decision tree constructed using the whole augmented test suite, and Figure 3.13 shows the averaged variable importance of trees constructed using all features over the 100 runs using a training set. The decision tree makes use of the newly defined feature $LCC_{0.3}^{0.2}$ and the literature feature $RRMSE$, although interestingly it makes no use of the feature CC. Figure 3.13 further shows the importance of the newly presented features, as the dominant features in terms of importance do not include any of the literature features. The results obtained through the analysis of the augmented test suite lead to the following conclusions:

- While Co-Kriging may be recommended under special conditions, such as a high CC, when we consider a more comprehensive set of instances we find that Co-Kriging is not likely to perform better than Kriging for many types of instances.
- Choosing to always use Co-Kriging leads to only 38.8% accuracy in predicting which algorithm to use, with the majority rule choosing to always use Kriging instead with an accuracy of 61.2% (Table 3.2). It is therefore more likely that Co-Kriging will perform worse than Kriging except under some very specific conditions.
- The feature CC only gives a rough indication of when Co-Kriging can be used, as the accuracy of a decision tree constructed with this feature presents a small improvement over the majority rule (Table 3.2) and the feature importance of CC is very small relative to other features (Figure 3.13).
- On the other hand, measures of local correlation (i.e. how f_l behaves locally, and how this behaviour changes throughout Ω) give a good indication of when Co-Kriging can be used (Figure 3.13).
- For problems with a total budget of $5d$ and a cost ratio $C_r = 0.1$, it is beneficial to use f_l (through the use of Co-Kriging) only if f_l is often somewhat locally correlated to f_h ($LCC_{0.3}^{0.2} \geq 0.875$), or if the error between f_h and f_l is very small ($RRMSE < 0.0843$).

These conclusions represent a significant divergence from the assumptions presented in the literature, and the conclusions drawn in Section 3.1 are based on a more limited set

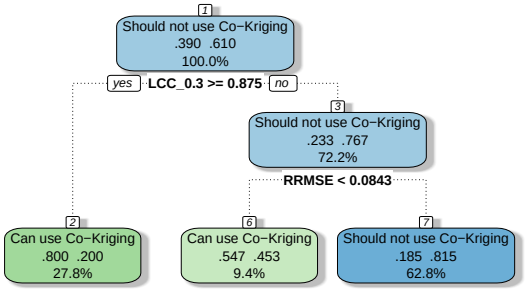


FIGURE 3.12: Decision tree constructed using new and literature features and the performance of Co-Kriging and Kriging on the augmented test suite. The label "Can use Co-Kriging" means the tree predicts Co-Kriging performs no worse than Kriging. The label "Should not use Co-Kriging" means the tree predicts Co-Kriging performs worse than Kriging. The percentages indicate the instances present in the node, and the proportions indicate the proportion of the instances in the node for which Co-Kriging can be used (left) and for which it should not be used (right).

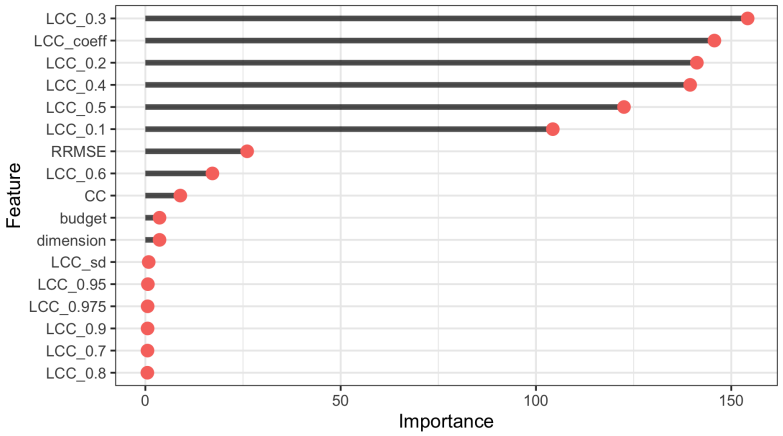


FIGURE 3.13: Average variable (feature) importance of 100 runs when constructing a decision tree using new and literature features on a training dataset chosen from the augmented test suite. Notice that the importance of the literature features is quite small relative to the newly defined features.

of test instances studied in the existing literature. These findings show that the choice of test instances matters, for reasons we discuss further in the next section.

3.3 Discussion

Sections 3.1 and 3.2 presented a comparative analysis of the performance of Kriging and Co-Kriging models through the use of two different instance test suites. Despite the analysis procedure being identical in both cases, the resulting conclusions strongly contradict one another. It is clear from these differences that the choice of test instances

can have a significant impact on the resulting algorithm analysis, and therefore on the directions taken in the field of multi-fidelity surrogate modelling and optimisation.

3.3.1 Impact of new features

The work of Toal [18] highlighted the need for some measure of quality of f_l when assessing whether a bi-fidelity technique such as Co-Kriging can be used when constructing a surrogate model. His work showcased the usefulness of the feature CC, which has led to subsequent studies [20, 22, 40, 64–66] to take this feature into account when performing algorithm analysis and instance creation. Despite its usefulness, the work presented here shows the potential shortfalls of this feature, in particular its global nature. Defining new features that measure the change in local correlation between f_l and f_h rather than overall correlation has revealed the potential bias present in existing literature instances. Section 3.1 presents what seems to be a varied literature test suite, however, Section 3.2 demonstrates that most instances in this suite have very similar CC and $LCC_{0.5}^{0.2}$ values, indicating that the overall quality of f_l and its local quality are similar throughout Ω in most literature instances. Many real-world problems however show variation in local f_l quality, due to model assumptions breaking down or device measurements having non-uniform errors, among others.

The dangers incurred by the bias of the literature test suite is demonstrated in the accuracy of the different classification techniques. Table 3.2 shows that for this test suite Co-Kriging performs no worse than Kriging most of the time, and therefore in general if a function f_l is available in an EBB problem it should be used when constructing a surrogate model. Figure 3.6 however indicates that this is likely the result of most instances having a large $LCC_{0.5}^{0.2}$ value, despite the whole test suite being varied in terms of the CC feature. When constructing the augmented test suite with instances with varying CC and $LCC_{0.5}^{0.2}$ values, Co-Kriging performs no worse than Kriging only 38.3% of the time. This does not mean Co-Kriging is not a good technique; rather, as Toal points out, one should be cautious and selective about when to use it.

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

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

The usefulness of the newly presented features is also clearly shown in Table 3.2. Despite the literature consensus being that the feature CC is sufficient to estimate whether Co-Kriging can be used, classification trees constructed using all features achieved a 77.3% accuracy on the augmented test suite, which is a significant improvement over the 65.3% accuracy of trees constructed using only the CC feature. It is worth mentioning however that this accuracy is still far from the 91.0% accuracy achieved on the more homogeneous literature test suite alone. This highlights the need for further work in this area, and in particular, the development of new features, to further assess the intricacies and differences between heterogeneous instances.

3.3.2 Impact of new instances

The results shown in this work also highlight the usefulness of the proposed instance creation framework. Both Toal [18] and Wang et al. [20] rightly emphasised the need for new instances in the field of Mf-EBB surrogate modelling and optimisation, and their work proposed instance creation techniques which led to a varied set of instances. Their work has been influenced however by the fact that the definition of "varied" has been based solely on the CC feature. The instance creation technique proposed by Wang et al. in particular leads to instances with a variety of CC values and therefore a variety of overall f_l quality. However, these instances suffer from having a very high $LCC^{0.2}_{0.5}$ value, indicating that nowhere in Ω does the quality of f_l worsen significantly. It is however possible in industry problems that the low-fidelity source of information worsens only in certain regions due to physical assumptions no longer holding, a lack of training data on a simulation model, or the readout of a measuring device moving away from its operating range.

The instance generating procedure presented in this work has shown its ability to create synthetic Bf-EBB instances with these more real-world characteristics. Despite only working with a single "basic" disturbance, the four modifications (two height based and two location based) can create instances with a large variety of CC and $LCC_{0.5}^{0.2}$ values. These instances show the promise of defining f_l functions with their quality being affected by either the output of f_h or the region of Ω being sampled. These types of modifications can be further adapted to suit the needs of the instance being created. The basic disturbance could be modified to closely resemble an error term found in a particular industry, such as the creation of stochastic instances via the use of white noise as the basic disturbance. The impact of the modification could also be altered, for instance defining the modification to "flatline" f_h in certain regions when creating f_l, to represent a loss in the ability of f_l to represent the intricate behaviour of f_h. The procedure could also be modified to construct Mf-EBB instances with more than 2 levels of fidelity by modulating the impact of the added disturbance based on the fidelity level. This could be achieved by modulating the amplitude of the basic disturbance, or by increasing/decreasing the radius of the final disturbance.

3.4 Conclusion

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

The work presented here constitutes a proof of concept rather than a complete work, however, and has some limitations. Despite the computational setup being very similar to that of Toal [18], the analysis in this chapter is more limited in the sense that both the

sample budget and the cost ratio have been fixed to $5d$ and $C_r = 0.1$, respectively. Due to this, it is important to place the obtained results in perspective. Namely, the decision tree presented at the end of Section 3.2 stating that Co-Kriging can be used for instances with $LCC_{0.3}^{0.2} \geq 0.875$ or $RRMSE < 0.0843$ should be taken as an indication of the importance of these features, rather than as a set of rules to be followed. Furthermore, whilst it is quite likely that the newly presented features will have a significant impact on algorithm performance for other budgets and cost ratios, the generalisation of these findings to other experimental settings remains to be conducted. This chapter has also been restricted by not allowing further samples to be gathered. It is not clear whether a technique which will create the most accurate model given an initial sample will do so if given a further sampling budget, and whether it will also perform best in function optimisation. Within this context, the usefulness of the new instances and features presented here remains to be investigated.

Finally, the increase in scope of future analysis in terms of varying instance budgets and cost ratios will require the use of more sophisticated classification and prediction techniques. In this chapter, a relatively simple machine learning method in the form of decision trees has been used. This choice can be justified as the aim here has not been to obtain a set of rules that further researchers should follow, but rather to showcase the importance of the presented instances and features. The following chapter will tackle the increased complexity of deriving industrial guidelines on low-fidelity source reliance which can be trusted by practitioners. This will require the analysis to be conducted with the limited available data used to train the models, to be simple enough to be easy to use, and to derive them in an intuitive manner that generates trust for the practitioners in the field.

Chapter 4

Characterising harmful data sources

As stated in the preface, the content of this chapter constitutes an adaptation of the submitted paper "Characterising harmful data sources when constructing multi-fidelity surrogate models". The introduction and conclusion have been adapted from this paper to fit the thesis format.

Chapter 3 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, it has 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. The newly proposed features and instance creation procedure have aimed to remedy this shortcoming. Chapter 3 has also shown 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 [18] and the results provided in Chapter 3 is their 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 [76] 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 [77–79]. Approaches of this type seek to understand what key instance 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 [80] 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 chapter can be seen as the logical extension of the work presented by Toal [18], enhanced by the instances and features presented in Chapter 3. Its focus is still the comparison of the accuracy of Kriging and Co-Kriging models in Bf-EBB problems with given high- and low-fidelity data sets. The two main contributions are the construction of an unbiased set of high- and low-fidelity function pairs for future algorithm testing, and the characterisation of harmful low-fidelity data sources, at least when constructing Co-Kriging models, using only the limited data available. This is highly relevant to the literature, as despite many new techniques having been proposed, both Kriging and Co-Kriging are very much still the surrogate models of choice in single-source and two-source EBB problems. The remainder of this chapter is structured as follows. Section 4.1 provides an introduction to ISA and a definition of the instances, features and algorithms used to perform the analysis. Section 4.2.1 presents the creation

of a set of objectively varied function pairs which are then used to generate a set of instances for ISA. Section 4.2.2 uses these instances to generate the instance space and identify the regions for which Kriging and Co-Kriging should be given precedence. Section 4.2.3 combines the identification of these regions and the feature value trends to explain why a low-fidelity source is harmful or beneficial. Section 4.2.4 provides a simplified single-feature analysis to provide some easy-to-use guidelines for practitioners in the field. Finally, Section 4.3 concludes the chapter with some closing remarks.

4.1 Preliminaries

4.1.1 Instance Space Analysis

Instance Space Analysis (ISA) [80] is based both on the work of Rice [76], which proposed using instance features to predict algorithmic performance, and the No-Free Lunch theorems proposed by Wolpert and Macready [81], 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.

Figure 4.1 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

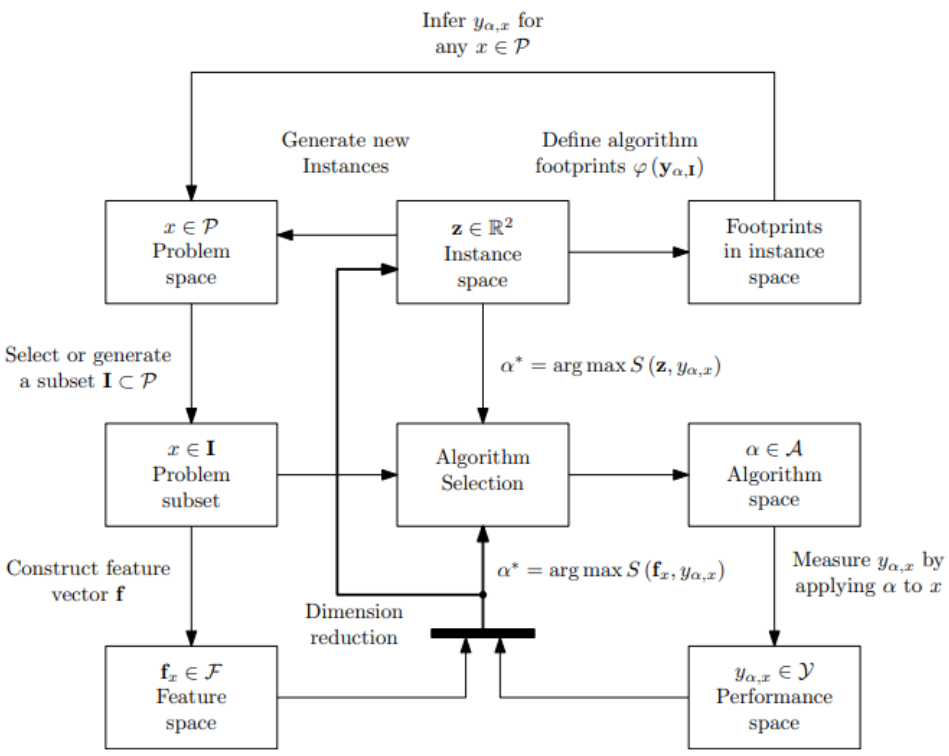


FIGURE 4.1: ISA framework [80]

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 [82] is publicly available, and the online tool Melbourne Algorithm Test Instance Library with Data Analytics (MATILDA) [83] 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 is conditional on the user's need, namely what kind of analysis is required. Furthermore, a large set of features can initially be used, as the implemented toolkit provides an automatic selection

procedure that only retains the most relevant features when creating the instance space. Choosing a subset $\mathbf{I} \subset \mathcal{P}$ however can be a harder problem, as recently shown by Alipour et al. [84]. 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 kth 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 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 $\mathfrak{u}_{D_\theta}$ is introduced, with

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

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 $\mathfrak{u}_{\mathcal{D}_\theta}$ so that its values lie between 0 and 1, and

choosing the smallest θ for which $\mathfrak{u}_{\mathcal{D}_\theta} \geq 0.5$. 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. [84].

4.1.2 Instances, Features and Algorithms

4.1.2.1 Instances

Recall that as presented in Chapter 2, the predominant characteristic of Bf-EBB problems is the existence of two black boxes that can be queried for information. They are defined as

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

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. In this chapter, 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. This is defined in Section 2.1 as Model creation with fixed sample. 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 $n_h \leq n_l$. 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 Section 4.2.1.

Note that contrary to the previous chapter, here both Kriging and Co-Kriging models will be trained with the same amount of high-fidelity data. This seems like a more natural approach when the user does not have a say on how to gather the data, and can only choose what to do with the data available to them. Note also that instead of

defining an instance based on the high- and low-fidelity sample available, it is defined on the high- and low- fidelity sample *size*. This allows for several repetitions with similarly gathered samples of the same size, which in turn allows for a statistical comparison of the performance of the models.

4.1.2.2 Algorithms and Performance

This study aims 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, once again the two well-established models Kriging [3, 23, 24] and Co-Kriging [38] are used. For an in-depth mathematical derivation, refer to Appendix A. 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 of "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

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

are constructed, with $|\mathbf{X}_l| = n_l$ and $|\mathbf{X}_h| = n_h$. The sets $\mathbf{X}_l$ and $\mathbf{X}_h$ correspond to the locations at which the low- and high-fidelity sources are sampled, respectively. They are generated following the procedure given in Section 2.2.3. That is $\mathbf{X}_l$ is obtained by creating a random Latin Hypercube Sampling (LHS) plan of Ω, locally optimised to be as evenly spread out as possible. 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. 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 [85].

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

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

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

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

$$\text{where} \qquad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} f_h(\mathbf{x}_i),$$

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

$$\bar{S} = \frac{1}{N}\sum_{i=1}^{N} s(\mathbf{x}_i),$$

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

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. 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 [74]. The performance value of Kriging on a particular instance is the p-value of a single-tailed Wilcoxon test with the null hypothesis that the correlation of the Kriging model with f_h is statistically no less than 0.001 than the correlation of the Co-Kriging model with f_h. In essence, the p-value of this test gives the

probability that the Kriging model accuracy is either higher or almost identical (within a tolerance of 0.001) to the accuracy of the Co-Kriging model. If this p-value is larger than 0.5, the performance of the Kriging model is assumed to be good. An identical procedure is followed to obtain the p-value of the Co-Kriging model. It is worth noting that both p-values can be large (i.e. the performance is "good" for both) if the performance of both models is statistically similar.

4.1.2.3 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 [18] which indicate the overall quality of f_l relative to f_h; these are defined in this thesis in Equations 2.8 and 3.1, respectively. The Local Correlation Coefficient presented in Chapter 3 are used as well, namely LCC_p^r, LCC_{sd}^r and LCC_{coeff}^r defined in Equations 3.2, 3.3, 3.4, respectively. Recall these features 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 $p \in \{0.1, 0.2, \ldots, 0.9, 0.95, 0.975\}$, 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 landscape features are taken from the R package flacco [86], which is made up of 17 feature sets, such as classic Exploratory Landscape Analysis (ELA) or Information Content (IC) 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. All flacco features used in this work are defined in Appendix D. 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 4.1, 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 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) 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.

4.2 ISA of Bf-EBB surrogate modelling with fixed sample

4.2.1 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 includes the 208 literature function pairs used in Chapter 3, which are gathered from a variety of studies [54–56, 58, 60, 64, 70–73] and include function pairs of dimensions $d \in \{1, 2, 3, 4, 5, 6, 8, 10, 20\}$. These are supplemented by over 80,000 function pairs generated by the instance-generating procedure proposed in Chapter 3. Recall that these

Method	Feature name	Description	Range
Overall quality of f_l [18]	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 (Chapter 3)	LCC_p^r	Proportion of time the local correlation between f_h and f_l is above a given $p \in [0, 1]$	[0,1]
	$LCC_{sd}^{0.2}$	Standard deviation of the local correlation between f_h and f_l	$[0, 1]$
	$LCC_{coeff}^{0.2}$	Coefficient of variation of the local correlation between f_h and f_l	$[0, \infty)$
Distribution (ELA distr) [87]	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) [87]	$MMCE_{lda}^q$	Mean misclassification error of a predictive linear model with data quantile split $q \in \{0.1, 0.25, 0.5\}$	$[0, 1]$
Meta modeling (ELA meta) [87]	$\bar{R}_L^2$	Adjusted R^2 of a linear model without interactions	$[0, 1]$
	$\bar{R}_{LI}^2$	Adjusted R^2 of a linear model with interactions	$[0, 1]$
	$\bar{R}_Q^2$	Adjusted R^2 of a quadratic model without interactions	$[0, 1]$
	$\bar{R}_{QI}^2$	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[88]	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 [89]	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 [90]	$\overline{DISP}_\epsilon$	Ratio of the mean distance between all points, and the mean distance between the ϵ% best points, $\epsilon \in \{2, 5, 10\}$	$[0, \infty)$
	$\widetilde{DISP}_\epsilon$	Ratio of the median distance between all points, and the median distance between the ϵ% best points, $\epsilon \in \{2, 5, 10\}$	$[0, \infty)$
Data budget	$B_h = n_h$	Number of high-fidelity samples available	$[2, 400]$
	$B_l = n_l$	Number of low-fidelity samples available	$[2, 400]$
	$B_h^r = n_h/d$	Relative number of high-fidelity samples available	$[2, 20]$
	$B_l^r = n_l/d$	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		$[1, 20]$

TABLE 4.1: A short description of the features used in this chapter, as well as the range of feature values for each feature.

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. Similarly to Chapter 3, in this chapter, the disturbance-based functions are generated using all 24 generating functions of the COCO test suite [75] with $d \in \{2, 4, 6, 8, 10\}$. The high-fidelity functions from the literature test suite are used as well however to generate disturbance-based functions. A slight variation of the disturbance parameters are used in this chapter. The values for frequency $\nu \in \{2, 5\}$, amplitude $\alpha \in \{0.1, 0.5, 1, 2\}$, radius $r \in \{0.025, 0.05, 0.1, 0.2\}$, height $h \in \{0, 0.25, 0.5, 0.75, 1\}$ and number of disturbance centers $\{1, 3, 5\}$ are used. Most of these parameters are quite similar to the previous chapter, with the notable exception of the frequency parameter ν, which takes much smaller values in this chapter. This leads to an added disturbance which oscillates less frequently, and therefore will look more realistic. As it will be seen in the results of this chapter, however, lowering this parameter still leads to varied instances which are different from any of the test instances currently available in the literature. Note once again that the definitions of the literature instances and the COCO functions are given in Appendix C.

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. To prevent a certain type of feature from having an unwanted weight 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 [91] 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. Further details about this simulation engine are given in Appendix C.

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. [84]. 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.

4.2.2 Algorithm performance analysis via ISA

Using the automated feature selection procedure of the MATILDA toolkit 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. To visualise this space, a projection onto 2-dimensional space is achieved via the projection matrix given in Equation 4.1. Recall that the features $LCC_{sd}^{0.2}$, $LCC_{0.4}^{0.2^{1/d}}$, $LCC_{0.95}^{0.2^{1/d}}$ 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 4.2.3. 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.

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} -0.4916 & -0.0889 \\ -0.3167 & -0.2321 \\ -0.1506 & 0.372 \\ -0.0568 & 0.4394 \\ 0.1777 & -0.4154 \\ 0.3696 & 0.0989 \\ 0.4362 & 0.0526 \\ 0.177 & 0.3381 \\ 0.4031 & 0.2545 \end{bmatrix}^\top \begin{bmatrix} B^r \\ LCC_{sd}^{0.2} \\ LCC_{0.4}^{0.2^{1/d}} \\ LCC_{0.95}^{0.2^{1/d}} \\ RRMSE \\ f_h,\ MMCE_{lda}^{0.5} \\ f_h,\ H_0 \\ f_h - f_l,\ \bar{R}_L^2 \\ f_h - f_l,\ \bar{R}_{LI}^2 \end{bmatrix} \tag{4.1}$$

Figure 4.2 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 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. However, the SOLAR instances lie towards the centre of the space, indicating that some synthetic instances that lie towards the border of the instance space might not represent realistic examples. This implies that whilst conclusions obtained from analysing algorithmic performance with synthetic instances will cover practical industrial cases, some extreme benchmarks might not be truly applicable to an applied setting.

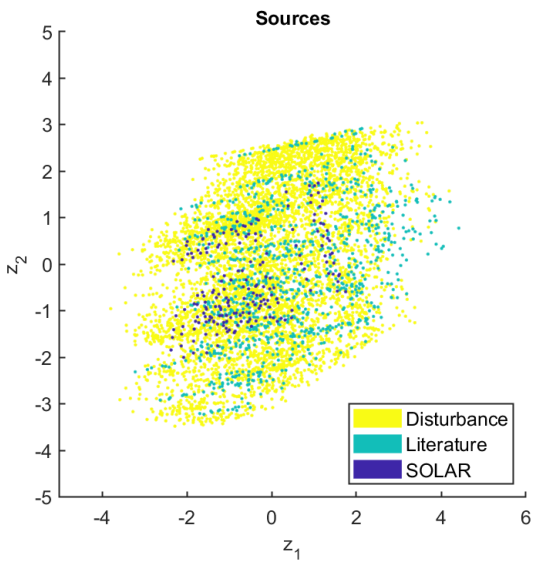
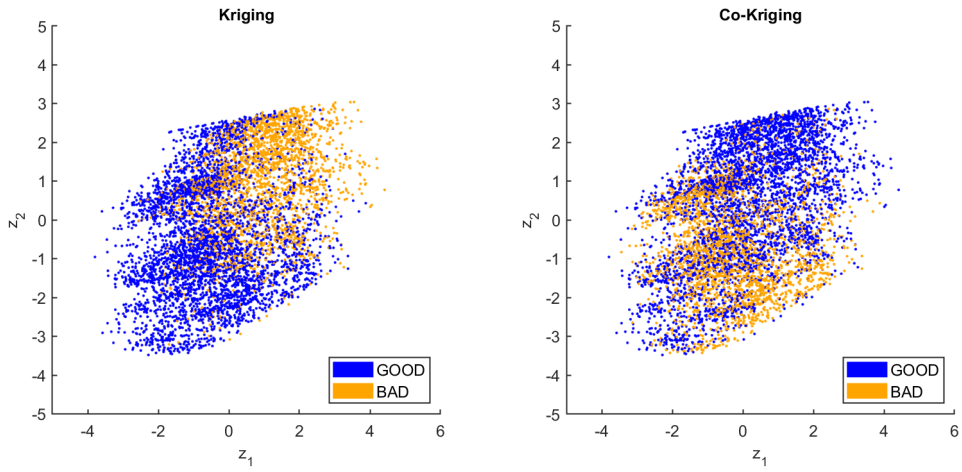


FIGURE 4.2: Sources of the function pairs used in each of the instances, where the dark blue points represent instances created from the SOLAR simulation, the light blue points represent classical literature instances, and the yellow points represent disturbance-based instances.

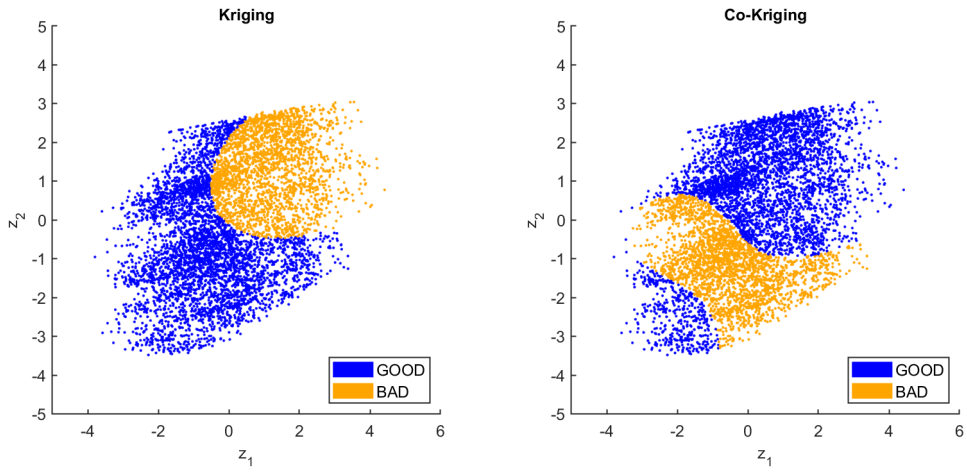
Figures 4.3a and 4.3b show the binary performance of the Kriging and Co-Kriging models respectively. Recall that a model is said to have good performance if it has a probability of at least 0.5 of having a higher or similar (within a tolerance of 0.001) correlation with f_h than its competitor. Figure 4.3a clearly shows a top-right region where Kriging has bad performance, indicating that ignoring the low-fidelity data is counter-productive. On the other hand, Figure 4.3b shows that, whilst Co-Kriging performs well in the top part of the space, moving towards the bottom left leads to a region containing a lot of instances where Co-Kriging performs badly. It is worth stressing once again that both models could perform well in the same instances, leading to blue regions in both plots. Therefore, despite the bottom 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 4.2.3.

Given the good/bad labels of both techniques, Support Vector Machines (SVMs) are trained to predict when both Kriging and Co-Kriging can be used, as well as a selector for which method should be used for each of the regions of the space. The SVMs



(a) (b)

FIGURE 4.3: 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.



(a) (b)

FIGURE 4.4: 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.

predictions for Kriging and Co-Kriging performance are shown in Figures 4.4a and 4.4b, respectively. It can be seen that in both the top-left and bottom-left regions both Kriging and Co-Kriging are predicted to be good, whereas only Co-Kriging is predicted to perform well in the top-right of the space and only Kriging is predicted to perform well in the bottom-right of the space. The best method predictions of the trained selector are shown in Figure 4.5. Perhaps unsurprisingly, the selector is almost identical to the SVM which predicts when Kriging performance is good or bad. Intuitively, this is the

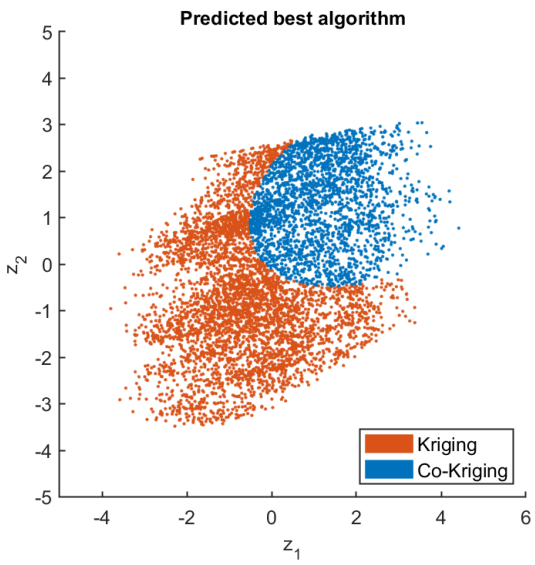


FIGURE 4.5: SVM selector which predicts whether Kriging or Co-Kriging should be used. The red dots indicate instances for which Kriging is predicted to be best, and the blue dots indicate instances for which Co-Kriging is predicted to be best.

TABLE 4.2: 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	0.610	77.4%	80.0%	84.0%
Co-Kriging	0.618	70.2%	76.0%	75.6%
Selector	0.806	80.6%	-	-

result of the division being much clearer for Kriging performance than for Co-Kriging performance. This is further reflected by Table 4.2, which presents the performance of all three SVMs. 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. In particular, a recall of 84.0% indicates that the Kriging SVM is very accurate when predicting when Kriging will perform badly. As such, the selector chooses Co-Kriging when Kriging is predicted to perform poorly and Kriging the rest of the time. This implies it is a lot easier to know when a low-fidelity source will be beneficial than when it will be harmful, indicating that a low-fidelity source should only be used if one can assert it will be an asset.

Finally, it is worth remarking that after filtering the metadata, the probability that

Kriging and Co-Kriging will perform well is roughly the same i.e. 0.610 vs. 0.618, respectively. Therefore, simply choosing to ignore or always use the low-fidelity source is the correct strategy only slightly over 60% of the time. The trained selector however correctly chooses a good model 80.6% of the time, despite basing its decision only on the limited sample used to train the models. This accuracy is also remarkable because 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.

4.2.3 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 0.12, 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 0.483 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 0.176 with algorithm performance. The feature B_h^r however

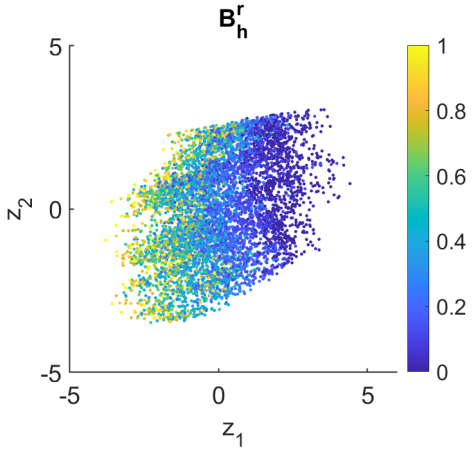


FIGURE 4.6: Feature distribution of the budget feature B_h^r. This feature indicates that going from the right to the left of the space means going through instances with the lowest to the highest relative amount of high-fidelity data. Note the colour grading is relative, with a value of 1 indicating a real feature value of 20, and a value of 0 indicating a real feature value of 2.

measures how much high-fidelity data is available relative to the problem dimension and has a correlation of 0.406, 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. As shown in Figure 4.6, 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 left of the space where both Kriging and Co-Kriging performs 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.

Figure 4.7 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 $LCC_{0.4}^{0.2^{1/d}}$ and $LCC_{0.95}^{0.2^{1/d}}$ measure the probability that f_l and f_h have a local correlation of 0.4 and 0.95, respectively, using a relative radius of $0.2^{1/d}$. The feature $RRMSE$ measures the error between f_h and f_l. The features $\bar{R}_L^2$ and $\bar{R}_{LI}^2$ are used to characterise the difference $f_h - f_l$ between sources, measuring how easy it is to model this difference using a simple linear model and a linear model with interactions, respectively. Finally,

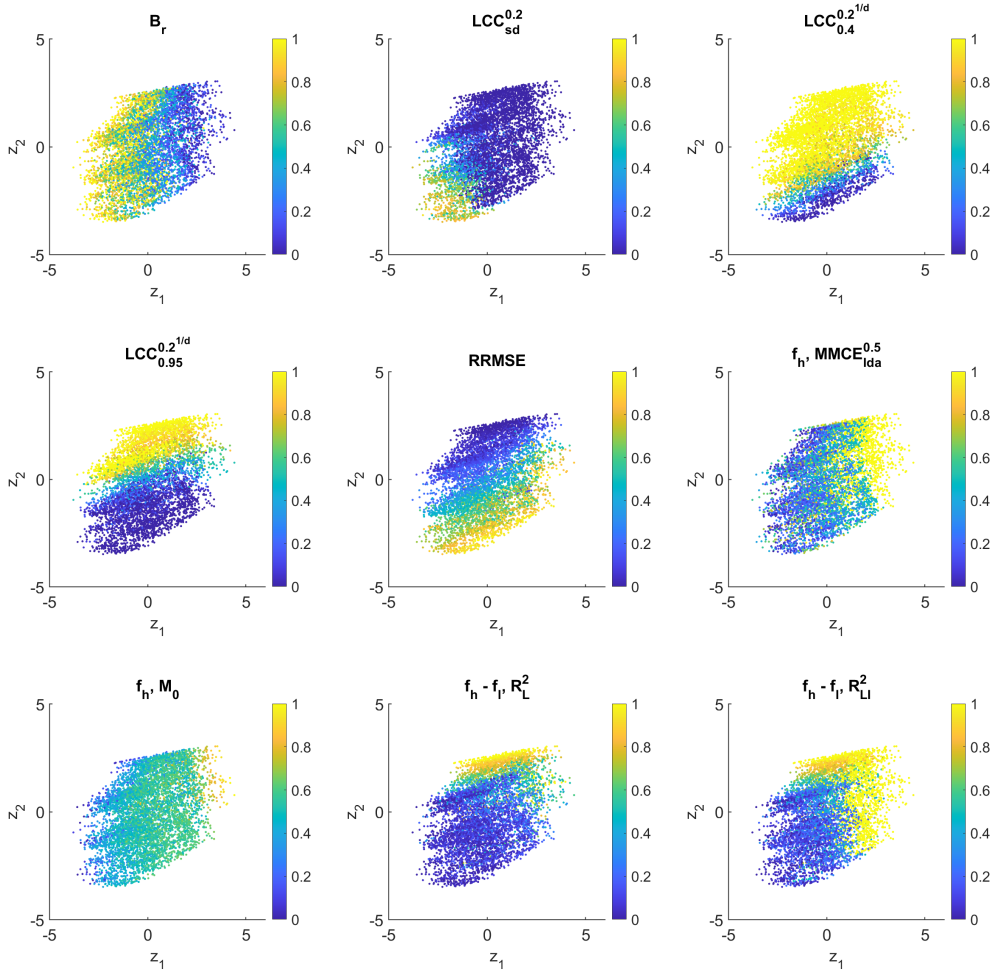


FIGURE 4.7: Feature distributions of the 9 chosen features. Note that the colour gradient is relative; for the range of each of the feature values consult Table 4.1.

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 here 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 $LCC_{0.4}^{0.2^{1/d}}$ and $LCC_{0.95}^{0.2^{1/d}}$ divides the space into a top, middle and bottom region. In the top region, $LCC_{0.95}^{0.2^{1/d}}$ 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 0.95 locally). This corresponds to a region where Co-Kriging can be used, and therefore the low-fidelity source is valuable. In the bottom region,

$LCC_{0.4}^{0.2^{1/d}}$ 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 0.4 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 given in Chapter 3 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 $LCC_{sd}^{0.2}$ plot in Figure 4.7 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 $\bar{R}_L^2$ and $\bar{R}_{LI}^2$ 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 easy to model. 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 $\bar{R}_L^2$ should be relied upon rather than the feature $\bar{R}_{LI}^2$, as 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 $\bar{R}_{LI}^2$ value is 1, but Co-Kriging performs worse than Kriging. Finally, note that a low $\bar{R}_L^2$ or $\bar{R}_{LI}^2$ value does not necessarily mean Co-Kriging cannot be used.

Not all selected features are necessarily helpful. The landscape features $MMCE_{lda}^{0.5}$ 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_h^r. 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

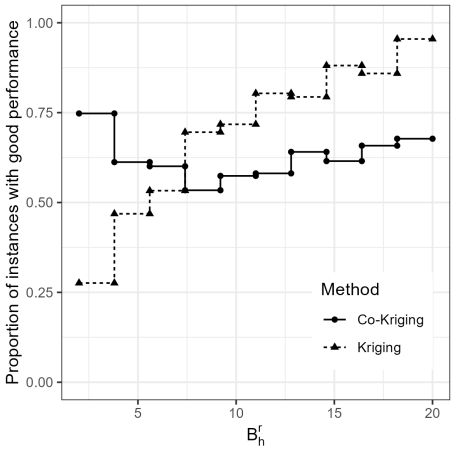


FIGURE 4.8: Proportion of instances for which Kriging and Co-Kriging perform well for different B_h^r values, the amount of high-fidelity data relative to problem dimension.

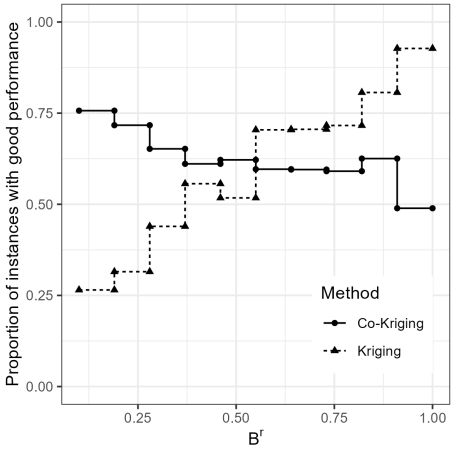


FIGURE 4.9: Proportion of instances for which Kriging and Co-Kriging perform well for different B^r values, the amount of high-fidelity data available relative to low-fidelity data available.

analysed. As such, it is also not recommended to use this feature when predicting algorithm performance.

4.2.4 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 to derive some simple rules that industrial practitioners can follow.

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 Figure 4.8. 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 almost guaranteed 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 between 18 and 20 times the dimension of the problem, Kriging is a good choice for 95% of the instances. Further analysis of harmful data sources should therefore be restricted to cases where at most $20d$ high-fidelity samples are available. Figure 4.9 indicates that 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 93% of the instances. Based on this graph, it is recommended to have at least 2.5 times as much low-fidelity as high-fidelity data when using Co-Kriging.

Figures 4.10, 4.11 and 4.12 provide good cut-off points for three feature values when choosing which model to use. In particular, Figure 4.10 shows that when $LCC_{0.4}^{0.2^{1/d}} \leq 0.7$, Kriging performs well for at least 76% of the instances in each of the feature bins. Similarly, Figure 4.11 shows that when $LCC_{0.95}^{0.2^{1/d}} \geq 0.7$, Co-Kriging performs well for at least 73% of the instances in each of the feature bins. Finally, Figure 4.12 shows that when $\bar{R}_L^2 \geq 0.4$, Co-Kriging also performs well for at least 73% 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 [18] 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, Figure 4.13 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

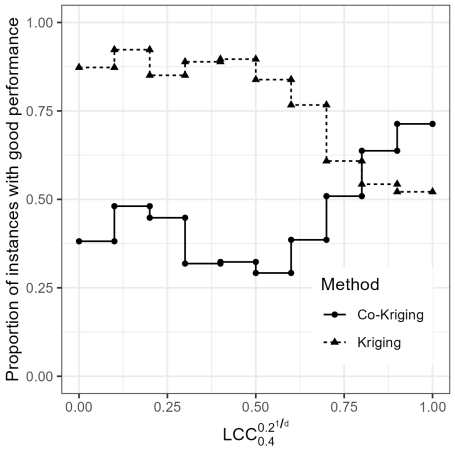


FIGURE 4.10: Proportion of instances for which Kriging and Co-Kriging perform well for different $LCC_{0.4}^{0.2^{1/d}}$ values.

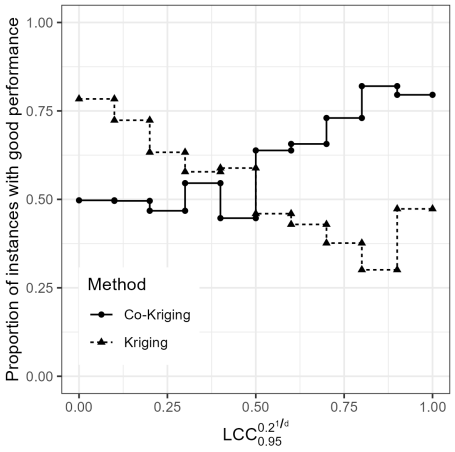


FIGURE 4.11: Proportion of instances for which Kriging and Co-Kriging perform well for different $LCC_{0.95}^{0.2^{1/d}}$ values.

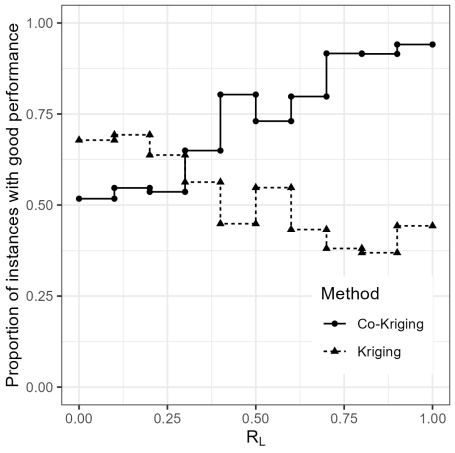


FIGURE 4.12: Proportion of instances for which Kriging and Co-Kriging perform well for different $\bar{R}_L^2$.

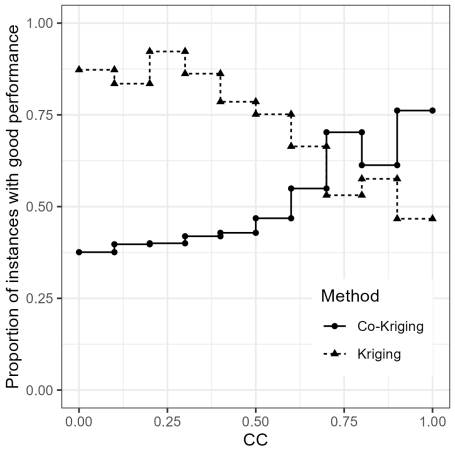


FIGURE 4.13: Proportion of instances for which Kriging and Co-Kriging perform well for different CC values, the global correlation between f_l and f_h.

75.0% 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 $B_h^r \geq 18$, $B^r = 1$ or $LCC_{0.4}^{0.2^{1/d}} \leq 0.7$, Kriging should be used.

2. For instances which do not satisfy the above conditions, but for which $LCC_{0.95}^{0.2^{1/d}} \geq 0.5$, or $\bar{R}_L^2 \geq 0.4$, Co-Kriging should be used.

3. For instances which are not covered by the previous two sets of rules, Co-Kriging should be used when $B_h^r \leq 5$, 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 of f_l and f_h data is available, 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.

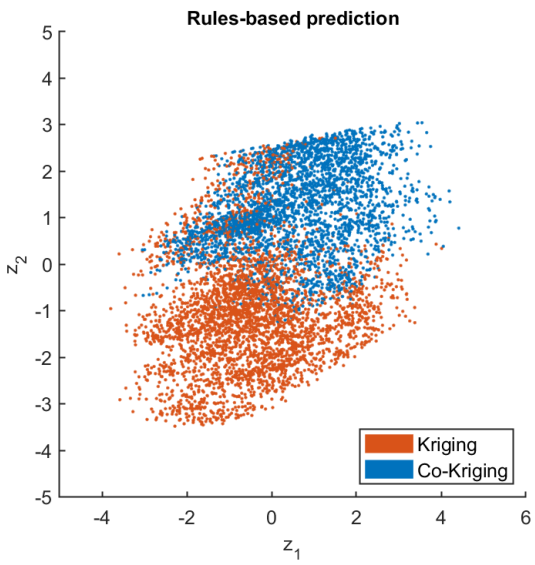


FIGURE 4.14: Selector which predicts whether Kriging or Co-Kriging should be used based on the simple rules given in this section. The red dots indicate instances for which Kriging is predicted to be best, and the blue dots indicate instances for which Co-Kriging is predicted to be best. Note that contrary to the predictor shown in Figure 4.5, no clear regions arise from this prediction.

Applying these guidelines to the filtered benchmark set containing 6500 instances yields very high accuracies. In particular, 2477 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 87.8% of the time. A total of 2503 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 81.6% 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 highly accurate recommendation. The accuracy of applying the full set of rules to the entire filtered benchmark set leads to a good algorithm being chosen 81.4% of the time, which is an improvement over the already highly 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 Figure 4.14. 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 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.

4.3 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 an 80.6% 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 only about 61% 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 in model prediction. Despite being based on easy to calculate features approximated with limited data, these rules lead to a good model being chosen 81.6% 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. It is important to note however that analysis of ISA is iterative. Further instances and features can be developed, 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 so far has focused on the simplest variant of Bf-EBB problems, namely model creation with a fixed sample as defined in Section 2.1.1. In this case, the only decision available to the practitioner is which model to train. The next chapter will shift its attention to the case where a sampling budget is supplied rather than an already collected data set. The findings presented in this chapter can likely be used to develop adaptive techniques that dynamically choose when to rely on low-fidelity

sources. Techniques of this type will likely outperform existing models that choose to always or never use a low-fidelity source.

Chapter 5

Adapting to budget constraints

The focus of the work conducted in this thesis so far has solely been on deciding whether a single-source or a two-source model should be trained given a set of data samples. It is very common however, particularly in an industrial setting, to begin the process of surrogate model building at a stage where no data has yet been gathered. Instead of being supplied with a dataset, a total sampling budget B is given which denotes the maximum number of times that a high-fidelity source f_h can be sampled. Furthermore, for every additional low-fidelity source that is available for sampling, a cost ratio $0 \leq C_r \leq 1$ is provided which denotes the cost of sampling the low-fidelity source relative to the cost of sampling f_h. Here the aim is to gather information by sampling the available sources, with the end-goal of modelling the high-fidelity source as accurately as possible once the overall sampling budget has been exhausted. In this thesis, this problem is denoted as model creation with a given budget and is formally defined in Section 2.1.2. In addition to deciding how to train a model given the data collected so far, here methods can be used to guide the data gathering process to collect further data where it is most needed. Furthermore, methods that accurately predict the cost-benefit trade-off of sampling each of the available sources can greatly increase their overall accuracy by judiciously allocating a portion of the available budget to each of the sources in some optimal way.

This chapter aims to investigate techniques that improve upon Kriging and Co-Kriging by assessing the quality of the low-fidelity source during the data-gathering process. Two mechanisms are considered that act on this assessment. The first considers whether

incorporating low-fidelity data into the model construction process improves the overall accuracy. If this is not the case, this source is no longer sampled and the data collected is not used when constructing the surrogate model. The second approach selects only one source to sample at each iteration based on the predicted improvement to the model's accuracy and the cost of the source being sampled. This may lead to repeatedly sampling from the low-fidelity source and only sampling the high-fidelity source at a reduced rate when it seems beneficial to do so, as well as sampling the low-fidelity source very little if the predicted accuracy gain from doing so is very small, even relative to the reduced sampling cost.

Few techniques in the literature offer solutions to both mechanisms. The classical Co-Kriging approach as presented by Forrester et al. [38] suggests choosing a location at which to sample both high- and low-fidelity sources at each iteration. In this sense, Co-Kriging can be viewed as a technique that always chooses to gather further low-fidelity data, whereas Kriging chooses to never gather any. Newer techniques have been proposed which aim to strike a balance between the two. Perhaps the most notable example is the work of van Rijn et al. [21]. Their research explored how to divide the remaining sampling budget between a high- and low-fidelity source. This division is based on an empirical assessment of the accuracy of models trained with subsets of the currently available data. This approach leads to a prediction of the improvement in model accuracy from further sampling each source. In this work, the approach taken chooses to rely instead on the prediction of model uncertainty reduction given by the trained models. New expressions are derived which predict the reduction in model uncertainty when sampling either only the high-fidelity source, or only the low-fidelity source, at a particular location. Comparing this predicted uncertainty reduction with the cost of sampling each of the sources gives rise to an approach that assesses the sampling trade-off. As it will be shown in the experimental study, this leads to an improvement over traditional Kriging and Co-Kriging methods, but is likely to be best used in conjunction with a prior analysis of the quality of the low-fidelity data source.

In terms of choosing whether to use or ignore the collected low-fidelity data, remarkably few techniques currently exist that base this decision on the characteristics of the available data. In theory, it is possible for a Co-Kriging model to make predictions that are independent of the low-fidelity data if, when training the hyperparameters, the parameter ρ is set to 0. This however is extremely unlikely, and therefore classical Co-Kriging

models are almost always impacted by all the data used to train them. The recent work of Eweis-Labolle et al. [41] consists of treating all available data as coming from a single information source, with a categorical variable indicating which fidelity level the sample comes from. The authors state that this allows the training of a Kriging model which can decide to limit the impact of low-fidelity data on future model predictions. This technique also relies on the underlying probabilistic theory of Kriging models to make the distinction between beneficial and harmful data. In this chapter, the empirically derived guidelines of the previous chapter will be used instead to predict whether the low-fidelity source is beneficial or harmful to the model's accuracy. As it will be seen in the results section, the insights gleaned during the characterisation of harmful data sources can lead to adaptive techniques that are reliably more accurate than the basic Kriging and Co-Kriging approaches.

The remainder of this chapter is structured as follows. Section 5.1 presents the surrogate modelling framework used to generate an initial design of experiments and then gather further data samples until the whole sampling budget is spent. Sections 5.1.2 and 5.1.3 in particular present novel approaches for both model selection and data acquisition. Section 5.2 presents a computational study that compares the performance of Kriging, Co-Kriging and three new techniques for a variety of sampling budgets, cost ratios and function pairs. The obtained results will show that in almost all investigated scenarios, using one of the newly developed techniques is a much safer approach than using Kriging or Co-Kriging. Finally, Section 5.3 closes the chapter with some concluding remarks.

5.1 Surrogate modelling framework

As discussed in the introduction, the problem being tackled in this chapter consists of sampling two information sources with a limited budget to try to model the high-fidelity source as accurately as possible. Therefore here an instance is formally defined by the tuple (f_h, f_l, B, C_r), where f_h denotes the high-fidelity source which needs to be modelled as accurately as possible, f_l denotes the low-fidelity source which may aid in surrogate modelling construction, B denotes the overall sampling budget, and C_r denotes the cost of sampling f_l relative to f_h. Note that as is the case throughout the thesis, both sources share the same hypercube domain $\Omega = [x_1^{\perp}, x_1^{\top}] \times \cdots \times [x_d^{\perp}, x_d^{\top}] \subset \mathbb{R}^d$. Methods which solve this problem do so by generating a dataset $(\mathbf{X}_h, \mathbf{X}_l, \mathbf{y}_h, \mathbf{y}_l)$ with $|\mathbf{X}_h| + C_r|\mathbf{X}_l| \leq B$,

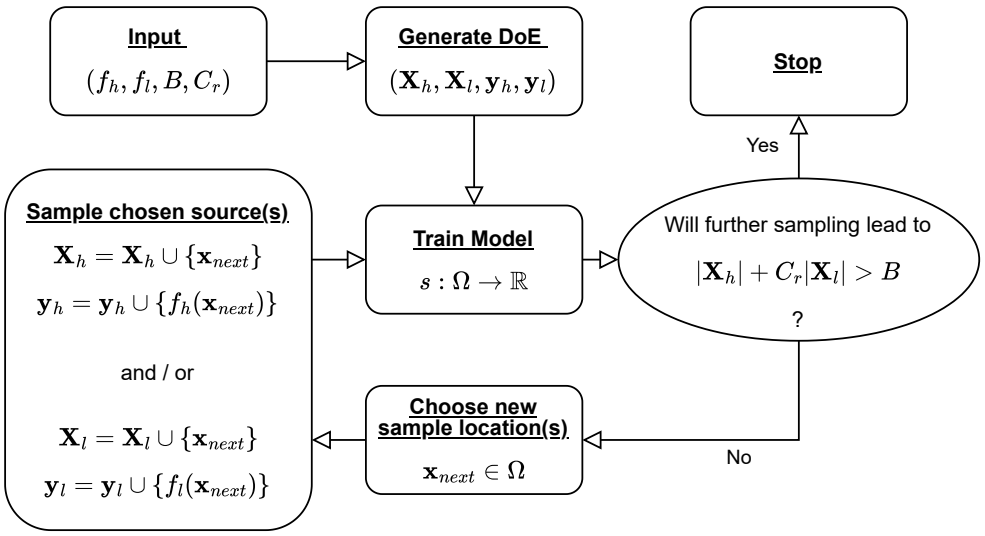


FIGURE 5.1: Surrogate modelling framework given an instance defined by two sources f_h and f_l, a sampling budget B and a cost ratio C_r.

where $\mathbf{X}_h$ and $\mathbf{X}_l$ denote the set of locations at which f_h and f_l have been sampled, respectively, and $\mathbf{y}_h$ and $\mathbf{y}_l$ denote the values of f_h and f_l at those locations, respectively.

Whilst it is possible to generate this data via a one-shot procedure, a more fine-tuned approach consists of spending a portion of the overall sampling budget in an initial sample (also known as Design of Experiments (DoE)), and then using the constructed model to guide further sampling of both sources. As such, each technique is uniquely identified by four key decisions, including how to construct the initial DoE, what model to train at each iteration, how to choose further sample locations, and how to choose which source(s) to sample at each iteration. The framework which illustrates these decisions is shown in Figure 5.1. The different approaches to each decision are presented next, the combination of which gives rise to a total of 5 techniques.

5.1.1 Design of Experiments

The first phase of the surrogate modelling procedure includes deciding what proportion of the overall budget should be spent in an initial DoE, how this allocated budget should be distributed between sources, and how the samples themselves should be collected. As the focus of this chapter is the performance of different data assessment and acquisition techniques, for comparative purposes it is best to take the same approach with all the techniques being investigated. This leads to every technique starting with the same

dataset when run on the same instance, and therefore the final performance is only the result of other components of each of the techniques. As discussed in Section 2.2.3 however, there seems to be no consensus in the literature in terms of best practices concerning the division of the budget between sources in the initial sample gathering. Therefore the initial DoE is generated according to the two following requirements

1. Half of the budget should be spent in the DoE, i.e. $|\mathbf{X}_h| + C_r|\mathbf{X}_l| = \frac{1}{2}B$, and
2. The low-fidelity source should be sampled twice as many times as the high-fidelity source, i.e. $|\mathbf{X}_l| = 2|\mathbf{X}_h|$.

The first requirement is simply taken as an intuitive approach, where half the budget is spent on a spread-out sample to better understand the characteristics of the sources, and the remainder of the sampling is guided by what is hoped to be a well-informed model. The second requirement is the result of the insights from both Chapter 4 and the work of Toal [18], both of which suggest that it is important to have more low- than high-fidelity data for multi-fidelity models to perform well. The analysis of whether these starting requirements are optimal or whether better approaches exist and are technique-dependent is left for future research.

As both C_r and B are given as part of the input, the two requirements given above lead to the following starting sample sizes for both sources

$$|\mathbf{X}_l| = \frac{B}{2C_r + 1},$$
$$|\mathbf{X}_h| = \frac{B}{2(2C_r + 1)}.$$

Having chosen the initial sizes for both data sets, the actual samples are placed using the locally optimised LHS approach presented in Section 2.2.3. Namely, the low-fidelity samples are chosen by first generating a random LHS plan which is then locally optimised so that the distance between any pair of points is as large as possible. A subset sample of high-fidelity points is then chosen by again maximising the minimum distance between any pair of points.

5.1.2 Choice of model

Unlike many studies in the literature, in this chapter, no new data fusion models are presented. Instead, in addition to Kriging and Co-Kriging, here the so-called "Rules-based Co-Kriging" is proposed. This method consists of making a choice, at each iteration, to train either a Kriging model with only $(\mathbf{X}_h, \mathbf{y}_h)$, or a Co-Kriging model with $(\mathbf{X}_h, \mathbf{X}_l, \mathbf{y}_h, \mathbf{y}_l)$. This decision is based on an adaptation of the guidelines proposed in Chapter 4 to the problem of model creation with a sample budget. That is, at each iteration, the budget features $B_h^r = \frac{|\mathbf{X}_h|}{d}$ and $B^r = \frac{|\mathbf{X}_h|}{|\mathbf{X}_l|}$, the relationship features $LCC_{0.4}^{0.2^{1/d}}$ and $LCC_{0.95}^{0.2^{1/d}}$ defined in Equation 3.2, and the landscape feature $\bar{R}_L^2$ of the difference $f_h - f_l$ defined in Equation D.1, are approximated using the data $(\mathbf{X}_h, \mathbf{X}_l, \mathbf{y}_h, \mathbf{y}_l)$ obtained so far. Then the following rules are followed to decide what model to train

1. If $B^r \geq 1$, $LCC_{0.4}^{0.2^{1/d}} \leq 0.7$, train a Kriging model.
2. Otherwise, if $LCC_{0.95}^{0.2^{1/d}} \geq 0.5$ or $\bar{R}_L^2 \geq 0.4$, train a Co-Kriging model.
3. Otherwise, train the same model as the previous iteration.

Note these rules constitute a slight variation on the guidelines proposed in Chapter 4. The main difference is the removal of the reliance on the budget feature B_h^r. Indeed, as the original guidelines were not developed with an iterative process in mind, they do not take into consideration the cost of having gathered low-fidelity data. Therefore, a transition between iterations from Co-Kriging to Kriging is only triggered when the low-fidelity data is deemed harmful i.e. condition one. Similarly, a transition from Kriging to Co-Kriging is only triggered when the low-fidelity data is deemed to be definitively beneficial i.e. condition two. Otherwise, the chosen model is the same as the previous iteration. When implementing this technique, in the first iteration it is assumed that a Kriging model has been chosen, indicating that a transition to using the low-fidelity data should only happen if it appears to be beneficial to the model's accuracy.

In the experimental section, the performance of Rules-based Co-Kriging will be compared to Kriging which always chooses to ignore the low-fidelity data, and Co-Kriging, which is trained with all available data and may gather further low-fidelity samples. The approach used to choose the locations of further samples is presented next.

5.1.3 Data acquisition

As shown in Figure 5.1, after training the chosen surrogate model further samples may be collected. It is possible to simultaneously choose multiple locations at which to sample the sources, although doing so can become computationally challenging. In this chapter, the data acquisition approaches considered only choose a single location at which one or more of the sources are sampled, before re-training the model and choosing further sampling sites. As discussed in Section 2.2.4, choosing further sample locations by maximising the model error can lead to a lot of samples being spent on the boundary of the space. A more global approach aims to gather samples that minimise the overall model uncertainty, and has been shown to produce more accurate models than when sampling via maximisation of model error [47], at least for single-source EBB models. The acquisition function which defines this approach is known as the Integrated Mean Square Prediction Error (IMSPE), and is defined as

$$IMSPE = \int_{\mathbf{x}\in\Omega} v(\mathbf{x})d\mathbf{x},$$

where $v(\mathbf{x})$ is the predicted model error at point $\mathbf{x}$. The acquisition function can then choose further samples which maximise the reduction in model error, namely

$$\underset{\mathbf{x}_{next}\in\Omega}{\operatorname{argmax}} \quad \Delta_{\mathbf{x}_{next}} IMSPE = \underset{\mathbf{x}_{next}\in\Omega}{\operatorname{argmax}} \quad \int_{\mathbf{x}\in\Omega} v(\mathbf{x})d\mathbf{x} - \int_{\mathbf{x}\in\Omega} v_{\mathbf{x}_{next}}(\mathbf{x})d\mathbf{x},$$

where $v_{\mathbf{x}_{next}}(\mathbf{x})$ is the predicted model error at point $\mathbf{x}$ after sampling at the location $\mathbf{x}_{next}$, and $\Delta_{\mathbf{x}_{next}} IMSPE$ is the (positive) reduction in $IMSPE$ after sampling $\mathbf{x}_{next}$. Traditionally this quantity has been approximated via Monte-Carlo integration [47], which can be very computationally intensive, especially with problems of medium to large dimension. Recently however a closed form expression of $\Delta_{\mathbf{x}_{next}} IMSPE$ has been proposed by Binois et al. [48] for certain surrogate models. The proposed closed form expression can be used in Kriging models with slightly simplified hyperparameters and is given next.

5.1.3.1 IMSPE for Kriging

What follows is a simplified derivation of the closed form expression of $\Delta_{\mathbf{x}_{next}} IMSPE$ for a Kriging model trained on a set of samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$; the full derivation is provided in Appendix E. Note that this derivation assumes that the hyperparameters $p_1, \ldots, p_k$ are set to 2 i.e. $p_1 = \cdots = p_k = 2$, and that $\Omega = [0, 1]^d$. The second assumption is only used to generate a simpler expression and is not strictly required; note however that it can be beneficial to linearly scale the input data to lie in the unit hypercube before training the surrogate model of choice. The first assumption is necessary to obtain the closed form expression, and may restrict the Kriging modelling capabilities. As the hyperparameters $\theta_1, \ldots, \theta_d$ can still be freely optimised however, this simplified Kriging model still has a strong modelling capability in the sense that each of the inputs can still be independently scaled internally by the model.

To obtain a closed form expression of $\Delta_{\mathbf{x}_{next}} IMSPE$ for a Kriging model, Binois et al. [48] first note the following

$$\begin{aligned}
\int_{\mathbf{x}\in\Omega} v^{krig}(\mathbf{x}) d\mathbf{x} &= \int_{\mathbf{x}\in\Omega} \hat{\sigma}^2 - \hat{\sigma}^2 \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}) d\mathbf{x}, \\
&= \hat{\sigma}^2 \int_{\mathbf{x}\in\Omega} 1 d\mathbf{x} - \hat{\sigma}^2 \int_{\mathbf{x}\in\Omega} \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}) d\mathbf{x}, \\
&= \hat{\sigma}^2 \int_{\mathbf{x}\in\Omega} 1 d\mathbf{x} - \hat{\sigma}^2 \mathbb{E}\left[\mathbf{r}(X)^\top \mathbf{R}^{-1} \mathbf{r}(X)\right],
\end{aligned}$$

with X being a random variable uniformly sampled in Ω, $\mathbf{R}$ is the trained covariance matrix of the Kriging model, and $\mathbf{r}(\cdot)$ is the Kriging vector defined in Equation 2.1. Lemma 1 of Binois et al. shows that

$$\mathbb{E}\left[\mathbf{r}(X)^\top \mathbf{R}^{-1} \mathbf{r}(X)\right] = tr(\mathbf{R}^{-1} \mathbf{W}^{krig}),$$

where $tr(\cdot)$ denotes the trace of the matrix, and the $n \times n$ matrix $\mathbf{W}^{krig}$ is defined as

$$\mathbf{W}^{krig}_{i,j} = W(\mathbf{x}_i, \mathbf{x}_j) = \int_{\mathbf{x}\in\Omega} \mathbf{r}(\mathbf{x})_i \mathbf{r}(\mathbf{x})_j d\mathbf{x}. \tag{5.1}$$

Here $\mathbf{r}(\mathbf{x})_i$ denotes the ith entry of the $\mathbf{r}(\mathbf{x})$ vector. An expression is then derived for each of the entries of the $\mathbf{W}^{krig}$ matrix which is fast to calculate, namely

$$W(\mathbf{x}_i, \mathbf{x}_j) = \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k, \theta_k),$$

$$\text{where } w(x_1, x_2, \theta_1, \theta_2) = \frac{\sqrt{\pi}}{2\sqrt{\theta_1+\theta_2}} \exp\left\{\frac{-\theta_1\theta_2(x_1-x_2)^2}{\theta_1+\theta_2}\right\} \times \left[\operatorname{erf}\left(\frac{\theta_1+\theta_2-x_1\theta_1-x_2\theta_2}{\sqrt{\theta_1+\theta_2}}\right) + \operatorname{erf}\left(\frac{x_1\theta_1+x_2\theta_2}{\sqrt{\theta_1+\theta_2}}\right)\right]. \tag{5.2}$$

Here $\operatorname{erf}(\cdot)$ stands for the Gauss error function, and $\mathbf{x}^{(k)}$ denotes the kth entry of the $\mathbf{x}$ vector. The details of this derivation are provided in Appendix E.1. Finally, let $\mathbf{R}_{\mathbf{x}_{next}}$ and $\mathbf{W}^{krig}_{\mathbf{x}_{next}}$ denote the updated $\mathbf{R}$ and $\mathbf{W}^{krig}$ matrices of the Kriging model after f_h has been sampled at $\mathbf{x}_{next}$. Then

$$\begin{aligned}
\Delta_{\mathbf{x}_{next}} IMSPE &= \int_{\mathbf{x}\in\Omega} v^{krig}(\mathbf{x})d\mathbf{x} - \int_{\mathbf{x}\in\Omega} v^{krig}_{\mathbf{x}_{next}}(\mathbf{x})d\mathbf{x}, \\
&= -\hat{\sigma}^2 \left[tr(\mathbf{R}^{-1}\mathbf{W}^{krig}) - tr(\mathbf{R}^{-1}_{\mathbf{x}_{next}}\mathbf{W}^{krig}_{\mathbf{x}_{next}})\right], \\
&= \frac{\hat{\sigma}^2}{1-\mathbf{r}(\mathbf{x}_{next})^\top\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_{next})}\Big[\mathbf{r}(\mathbf{x}_{next})^\top\mathbf{R}^{-1}\mathbf{W}^{krig}\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_{next}) \\
&\qquad - 2\mathbf{w}^{krig}(\mathbf{x}_{next})^\top\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_{next}) + W(\mathbf{x}_{next}, \mathbf{x}_{next})\Big],
\end{aligned} \tag{5.3}$$

where

$$\mathbf{w}^{krig}(\mathbf{x}) = [W(\mathbf{x}, \mathbf{x}_1) \dots W(\mathbf{x}, \mathbf{x}_n)]^\top. \tag{5.4}$$

The full derivation is given in Appendix E.2. Note that after pre-calculating the matrices $\mathbf{R}^{-1}$ and $\mathbf{W}^{krig}$, this expression can be evaluated in $O(n^2)$ which is very fast in practice, leading to a tractable acquisition function even for large problem dimensions and sample sizes.

5.1.3.2 IMSPE for Co-Kriging

Next, we use the same approach to derive a novel expression for the change in IMSPE of a Co-Kriging model when gathering further samples; once again the full derivation is

given in Appendix E. As this model is trained using samples $\mathbf{X}_h = \{\mathbf{x}_1^h, \ldots, \mathbf{x}_{n_h}^h\}$ and $\mathbf{X}_l = \{\mathbf{x}_1^l, \ldots, \mathbf{x}_{n_l}^l\}$ from f_h and f_l respectively, it is possible when gathering further data to sample both sources simultaneously, or to choose instead to sample only one of the sources. Therefore define the following notation

$$\begin{aligned}
\Delta^l_{\mathbf{x}_{next}} IMSPE = &\ \text{Change in } IMSPE \text{ if } f_l \text{ is sampled at } \mathbf{x}_{next}, \\
\Delta^h_{\mathbf{x}_{next}} IMSPE = &\ \text{Change in } IMSPE \text{ if } f_h \text{ is sampled at } \mathbf{x}_{next}, \\
\Delta^b_{\mathbf{x}_{next}} IMSPE = &\ \text{Change in } IMSPE \text{ if both } f_h \text{ and } f_l \text{ are sampled at } \mathbf{x}_{next}.
\end{aligned}$$

Similarly to Kriging, the first step is to note that

$$\begin{aligned}
\int_{\mathbf{x}\in\Omega} v^{cokrig}(\mathbf{x})d\mathbf{x} &= \int_{\mathbf{x}\in\Omega} \hat{\rho}\hat{\sigma}_l^2 + \hat{\sigma}_{diff}^2 - \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1}\mathbf{c}(\mathbf{x})d\mathbf{x}, \\
&= \left(\hat{\rho}\hat{\sigma}_l^2 + \hat{\sigma}_{diff}^2\right) \int_{\mathbf{x}\in\Omega} 1 d\mathbf{x} - \int_{\mathbf{x}\in\Omega} \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1}\mathbf{c}(\mathbf{x})d\mathbf{x}, \\
&= \left(\hat{\rho}\hat{\sigma}_l^2 + \hat{\sigma}_{diff}^2\right) \int_{\mathbf{x}\in\Omega} 1 d\mathbf{x} - \mathbb{E}\left[\mathbf{c}(X)^\top \mathbf{C}^{-1}\mathbf{c}(X)\right].
\end{aligned}$$

with X being a random variable uniformly sampled in Ω, $\mathbf{C}$ is the trained covariance matrix of the Co-Kriging model, and $\mathbf{c}(\cdot)$ is to Co-Kriging vector defined in Equation 2.6. Once again using the same Lemma we obtain

$$\mathbb{E}\left[\mathbf{c}(X)^\top \mathbf{C}^{-1}\mathbf{c}(X)\right] = tr(\mathbf{C}^{-1}\mathbf{W}^{cokrig}).$$

Here the $(n_l + n_h) \times (n_l + n_h)$ matrix $\mathbf{W}^{cokrig}$ is defined as

$$\mathbf{W}_{i,j}^{cokrig} = \int_{\mathbf{x}\in\Omega} \mathbf{c}(\mathbf{x})_i \mathbf{c}(\mathbf{x})_j d\mathbf{x}. \tag{5.5}$$

Here $\mathbf{c}(\mathbf{x})_i$ denotes the ith entry of the $\mathbf{c}(\mathbf{x})$ vector. By assuming that $\Omega = [0,1]^d$ and $p_1^l = \cdots = p_d^l = p_1^{diff} = p_n^{diff} = 2$, each of the entries of the $\mathbf{W}^{cokrig}$ matrix can be

calculated quickly, and are given by

$$\mathbf{W}_{i,j}^{cokrig} = \begin{cases} W^1(\mathbf{x}_i^l, \mathbf{x}_j^l) & \text{for } 1 \leq i,j \leq n_l, \\ W^2(\mathbf{x}_i^l, \mathbf{x}_{j-n_l}^h) & \text{for } 1 \leq i \leq n_l < j \leq n_l + n_h, \\ W^2(\mathbf{x}_j^l, \mathbf{x}_{i-n_l}^h) & \text{for } 1 \leq j \leq n_l < i \leq n_l + n_h, \\ W^3(\mathbf{x}_{i-n_l}^h, \mathbf{x}_{j-n_l}^h) & \text{for } n_l < i,j \leq n_l + n_h. \end{cases}$$

The functions W^1, W^2 and W^3 are defined as

$$\begin{aligned} W^1(\mathbf{x}_i, \mathbf{x}_j) &= \left(\hat{\rho}\hat{\sigma}_l^2\right)^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^l, \theta_k^l), \\ W^2(\mathbf{x}_i, \mathbf{x}_j) &= \hat{\rho}^3 \left(\hat{\sigma}_l^2\right)^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^l, \theta_k^l) \\ &\quad + \hat{\rho}\hat{\sigma}_l^2 \hat{\sigma}_{diff}^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^l, \theta_k^{diff}), \\ W^3(\mathbf{x}_i, \mathbf{x}_j) &= \left(\hat{\rho}^2\hat{\sigma}_l^2\right)^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^l, \theta_k^l) \\ &\quad + \hat{\rho}^2 \hat{\sigma}_l^2 \hat{\sigma}_{diff}^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^l, \theta_k^{diff}) \\ &\quad + \hat{\rho}^2 \hat{\sigma}_l^2 \hat{\sigma}_{diff}^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^{diff}, \theta_k^l) \\ &\quad + \left(\hat{\sigma}_{diff}^2\right)^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^{diff}, \theta_k^{diff}). \end{aligned}$$

The derivation of these expressions is given in Appendix E.1. Let $\mathbf{C}_{(l,\mathbf{x}_{next})}$, $\mathbf{W}_{(l,\mathbf{x}_{next})}^{cokrig}$ and $v_{(l,\mathbf{x}_{next})}^{cokrig}$ denote respectively the updated $\mathbf{C}$ and $\mathbf{W}^{cokrig}$ matrices and error function v^{cokrig} of the Co-Kriging model after f_l has been sampled at $\mathbf{x}_{next}$. Then

$$\begin{aligned} \Delta_{\mathbf{x}_{next}}^l IMSPE &= \int_{\mathbf{x}\in\Omega} v^{cokrig}(\mathbf{x})d\mathbf{x} - \int_{\mathbf{x}\in\Omega} v_{(l,\mathbf{x}_{next})}^{cokrig}(\mathbf{x})d\mathbf{x}, \\ &= -tr(\mathbf{C}^{-1}\mathbf{W}^{cokrig}) + tr\left(\mathbf{C}_{(l,\mathbf{x}_{next})}^{-1}\mathbf{W}_{(l,\mathbf{x}_{next})}^{cokrig}\right), \\ &= \frac{1}{\hat{\sigma}_l^2 - \mathbf{c}_l(\mathbf{x}_{next})^\top \mathbf{C}^{-1}\mathbf{c}_l(\mathbf{x}_{next})} \Big[\mathbf{c}_l^\top(\mathbf{x}_{next})\mathbf{C}^{-1}\mathbf{W}^{cokrig}\mathbf{C}^{-1}\mathbf{c}_l(\mathbf{x}_{next}) \\ &\qquad - 2\mathbf{w}_l^{cokrig}(\mathbf{x}_{next})^\top \mathbf{C}^{-1}\mathbf{c}_l(\mathbf{x}_{next}) + W^1(\mathbf{x}_{next}, \mathbf{x}_{next})\Big], \end{aligned} \tag{5.6}$$

where

$$\mathbf{w}_l^{cokrig}(\mathbf{x}) = [W^1(\mathbf{x}, \mathbf{x}_1^l) \dots W^1(\mathbf{x}, \mathbf{x}_{n_l}^l) \quad W^2(\mathbf{x}, \mathbf{x}_1^h) \dots W^2(\mathbf{x}, \mathbf{x}_{n_h}^h)]^\top, \tag{5.7}$$

$$\mathbf{c}_l(\mathbf{x}) = [\hat{\sigma}_l^2 R_l(\mathbf{x}_1^l, \mathbf{x}) \dots \hat{\sigma}_l^2 R_l(\mathbf{x}_{n_l}^l, \mathbf{x}) \quad \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}_1^h, \mathbf{x}) \dots \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}_{n_h}^h, \mathbf{x})]^\top. \tag{5.8}$$

Here the R_l function is as defined in Equation 2.3, and the full derivation is given in Appendix E.3. Similarly, let $\mathbf{C}_{(h,\mathbf{x}_{next})}$, $\mathbf{W}_{(h,\mathbf{x}_{next})}^{cokrig}$ and $v_{(h,\mathbf{x}_{next})}^{cokrig}$ denote respectively the updated $\mathbf{C}$ and $\mathbf{W}^{cokrig}$ matrices and error function v^{cokrig} of the Co-Kriging model after f_h has been sampled at $\mathbf{x}_{next}$. Then

$$\begin{aligned}
\Delta_{\mathbf{x}_{next}}^h IMSPE &= \int_{\mathbf{x}\in\Omega} v^{cokrig}(\mathbf{x})d\mathbf{x} - \int_{\mathbf{x}\in\Omega} v_{(h,\mathbf{x}_{next})}^{cokrig}(\mathbf{x})d\mathbf{x}, \\
&= -tr(\mathbf{C}^{-1}\mathbf{W}^{cokrig}) + tr\left(\mathbf{C}_{(h,\mathbf{x}_{next})}^{-1}\mathbf{W}_{(h,\mathbf{x}_{next})}^{cokrig}\right), \\
&= \frac{1}{\hat{\rho}^2\hat{\sigma}_l^2 + \hat{\sigma}_{diff}^2 - \mathbf{c}(\mathbf{x}_{next})^\top\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_{next})}\Big[\mathbf{c}(\mathbf{x}_{next})^\top\mathbf{C}^{-1}\mathbf{W}^{cokrig}\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_{next}) \\
&\qquad - 2\mathbf{w}_h^{cokrig}(\mathbf{x}_{next})^\top\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_{next}) + W^3(\mathbf{x}_{next}, \mathbf{x}_{next})\Big],
\end{aligned} \tag{5.9}$$

where

$$\mathbf{w}_h^{cokrig}(\mathbf{x}) = [W^2(\mathbf{x}, \mathbf{x}_1^l) \dots W^2(\mathbf{x}, \mathbf{x}_{n_l}^l) \quad W^3(\mathbf{x}, \mathbf{x}_1^h) \dots W^3(\mathbf{x}, \mathbf{x}_{n_h}^h)]^\top. \tag{5.10}$$

Here the $\mathbf{c}(\cdot)$ vector is as defined in Equation 2.6, and once again the full derivation is given in Appendix E.3. Finally, let $\mathbf{C}_{(b,\mathbf{x}_{next})}$, $\mathbf{W}_{(b,\mathbf{x}_{next})}^{cokrig}$ and $v_{(b,\mathbf{x}_{next})}^{cokrig}$ denote respectively the updated $\mathbf{C}$ and $\mathbf{W}^{cokrig}$ matrices and error function v^{cokrig} of the Co-Kriging model after both f_h and f_l have been sampled at $\mathbf{x}_{next}$. Then

$$\begin{aligned}
\Delta_{\mathbf{x}_{next}}^b IMSPE &= \int_{\mathbf{x}\in\Omega} v^{cokrig}(\mathbf{x})d\mathbf{x} - \int_{\mathbf{x}\in\Omega} v_{(b,\mathbf{x}_{next})}^{cokrig}(\mathbf{x})d\mathbf{x}, \\
&= -tr(\mathbf{C}^{-1}\mathbf{W}^{cokrig}) + tr\left(\mathbf{C}_{(b,\mathbf{x}_{next})}^{-1}\mathbf{W}_{(b,\mathbf{x}_{next})}^{cokrig}\right), \\
&= \mathbf{1}^\top\left(\mathbf{C}^{-1}\mathbf{c}_b(\mathbf{x}_{next})\mathbf{A}_b\mathbf{c}_b(\mathbf{x}_{next})^\top\mathbf{C}^{-1} \circ \mathbf{W}^{cokrig}\right)\mathbf{1} \\
&\quad - \mathbf{1}^\top\left(\mathbf{C}^{-1}\mathbf{c}_b(\mathbf{x}_{next})\mathbf{A}_b \circ \mathbf{w}_b^{cokrig}(\mathbf{x}_{next})\right)\mathbf{1} \\
&\quad - \mathbf{1}^\top\left(\mathbf{A}_b\mathbf{c}_b(\mathbf{x}_{next})^\top\mathbf{C}^{-1} \circ \mathbf{w}_b^{cokrig}(\mathbf{x}_{next})^\top\right)\mathbf{1} \\
&\quad + \mathbf{1}^\top\left(\mathbf{A}_b \circ \mathbf{W}_b^{cokrig}\right)\mathbf{1}
\end{aligned}. \tag{5.11}$$

Here $\circ$ stands for the element-wise matrix product, and

$$\mathbf{w}_b^{cokrig}(\mathbf{x}) = \begin{bmatrix} \mathbf{w}_l^{cokrig}(\mathbf{x}) & \mathbf{w}_h^{cokrig}(\mathbf{x}) \end{bmatrix}, \tag{5.12}$$

$$\mathbf{W}_b^{cokrig} = \begin{bmatrix} W^1(\mathbf{x}_{next}^b, \mathbf{x}_{next}^b) & W^2(\mathbf{x}_{next}^b, \mathbf{x}_{next}^b) \\ W^2(\mathbf{x}_{next}^b, \mathbf{x}_{next}^b) & W^3(\mathbf{x}_{next}^b, \mathbf{x}_{next}^b) \end{bmatrix}, \tag{5.13}$$

$$\mathbf{c}_b(\mathbf{x}) = \begin{bmatrix} \mathbf{c}_l(\mathbf{x}) & \mathbf{c}(\mathbf{x}) \end{bmatrix}, \tag{5.14}$$

$$\mathbf{A}_b = \left(\Sigma - \mathbf{c}_b(\mathbf{x}_{next})^\top \mathbf{C}^{-1} \mathbf{c}_b(\mathbf{x}_{next})\right)^{-1}, \tag{5.15}$$

$$\Sigma = \begin{bmatrix} \hat{\sigma}_l^2 & \hat{\rho}\hat{\sigma}_l^2 \\ \hat{\rho}\hat{\sigma}_l^2 & \hat{\rho}^2\hat{\sigma}_l^2 + \hat{\sigma}_{diff}^2 \end{bmatrix}. \tag{5.16}$$

The full derivation is again available in Appendix E.3. Note that similarly to the Kriging case, if the matrices $\mathbf{C}^{-1}$ and $\mathbf{W}^{cokrig}$ are available, the changes in $IMSPE$ provided in Equations 5.6, 5.9 and 5.11 can be calculated in $O((n_h + n_l)^2)$, which is once again very fast in practice.

5.1.3.3 Acquisition functions

Having derived expressions for the change in $IMSPE$ which are fast to calculate, what remains is to define the auxiliary optimisation problem used to obtain new samples. In the case of Kriging, this is simply defined as

$$\underset{\mathbf{x}_{next} \in \Omega}{\text{argmax}} \quad \Delta_{\mathbf{x}_{next}} IMSPE.$$

As Kriging only uses high-fidelity data, there is no decision to be made in terms of which source to sample. Having chosen the next sample location $\mathbf{x}_{next}$, only f_h is sampled. Similarly, traditional Co-Kriging simply finds a location at which to sample both sources. Therefore the acquisition function for Co-Kriging is

$$\underset{\mathbf{x}_{next} \in \Omega}{\text{argmax}} \quad \Delta_{\mathbf{x}_{next}}^b IMSPE.$$

Having chosen $\mathbf{x}_{next}$, both sources are sampled. As alluded to throughout this chapter however, a more judicious approach consists of only choosing one source to sample based on the trade-off between cost and uncertainty reduction. Therefore, the denoted "Adaptive Co-Kriging" consists of choosing the next sample site via

$$\underset{\mathbf{x}_{next}\in\Omega}{\text{argmax}} \quad \max\{\frac{1}{C_r}\cdot\Delta^l_{\mathbf{x}_{next}}IMSPE, \quad \Delta^h_{\mathbf{x}_{next}}IMSPE\}.$$

Once $\mathbf{x}_{next}$ is chosen, f_l is sampled if $\frac{1}{C_r}\cdot\Delta^l_{\mathbf{x}_{next}}IMSPE \quad \geq \quad \Delta^h_{\mathbf{x}_{next}}IMSPE$, and f_h is sampled otherwise. This can lead to sampling the low-fidelity source despite the decrease in model uncertainty being lower than when sampling the high-fidelity source, as long as the decrease relative to the cost makes this a better choice.

In terms of Rules-based Co-Kriging, the acquisition function is the same as for Kriging and traditional Co-Kriging depending on which of the two models have been chosen based on the guidelines. The final technique is denoted "Rules-based Adaptive Co-Kriging" and follows the same guidelines given in Section 5.1.2 to choose a Kriging or Co-Kriging model. In the case that a Co-Kriging model is trained, the acquisition function used is the same as for Adaptive Co-Kriging.

5.2 Computational study

5.2.1 Experimental setup

As stated in Section 5.1, an instance of model creation with a given budget is defined by the tuple (f_h, f_l, B, C_r). To test the performance of the five chosen techniques, the set of 321 function pairs chosen for the computational study of Chapter 4 is used here. Recall that these function pairs are chosen to be as diverse and therefore as unbiased as possible. For each function pair, the budget values $B \in \{5d, 10d, 15d, 20d\}$ and cost ratios $C_r \in \{0.5, 0.1, 0.025, 0.01\}$ are used, with d denoting the dimension of f_h and f_l. This leads to a total of 5136 instances being analysed. Note the budget is taken relative to the problem dimension and with an upper bound of $20d$ as suggested by the findings of Chapter 4. The chosen cost ratios denote low-fidelity sources which are 2, 10, 40 and 100 times cheaper than the high-fidelity source, respectively.

The type of problem considered here entails an increase in computational requirement over the problems considered in previous chapters. Therefore certain decisions have been made to allow for results to be gathered for such a large number of instances. First, a maximum number of $50d$ samples can be gathered from the low-fidelity source. This limit is placed on the Adaptive Co-Kriging and Rules-based Adaptive Co-Kriging techniques to prevent potentially thousands (for $C_r = 0.01$) of low-fidelity points from being gathered. Furthermore, as training the hyperparameters for Kriging and Co-Kriging models can become quite computationally taxing for larger training datasets, under certain conditions the rate at which the hyperparameters are trained is decreased. Specifically, when 100 or less points are available, the hyperparameters are optimised at every iteration. For samples larger than 100, 200 and 500 points the hyperparameters are optimised every 10, 50 and 100 iterations, respectively. All hyperparameters are optimised in the last iteration however. This means that the effect of fewer optimisations can lead to sub-optimal points being chosen. This approach is deemed to not impact the obtained results however as once again the interest is in the comparative rather than absolute performance between techniques, each of which follows this routine.

In terms of performance, the two measures of accuracy used are the Relative Root Mean Squared Error ($RRMSE$) and Pearson's correlation (P_{corr}) between the trained model s_h and f_h. These are calculated at a set of $N = 1000d$ locations and are given by

$$RRMSE = \frac{1}{\max_{i=1,\dots,N}\{f(\mathbf{x}_i)\} - \min_{i=1,\dots,N}\{f(\mathbf{x}_i)\}} \sqrt{\frac{\sum_{i=1}^{N}(s_h(\mathbf{x}_i) - f_h(\mathbf{x}_i))^2}{n}},$$

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

where

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

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

$$\bar{S} = \frac{1}{N}\sum_{i=1}^{N} s(\mathbf{x}_i),$$

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

In Chapters 3 and 4 a statistical comparison of accuracy was used between Kriging and

Co-Kriging. This is not possible here as techniques like Rules-based Co-Kriging can have identical performance to Kriging or Co-Kriging depending on the instance. Therefore comparison methods such as the Wilcoxon test cannot be used. Furthermore, only 5 repetitions with different initial sampling plans are conducted for every technique and instance due to the high computational resources needed for these problems. Instead of a statistical comparison, the next section shows the distribution of the accuracies of each of the techniques based on the available budget and cost ratios.

5.2.2 Results

Chapter 4 aimed to assess whether Kriging and Co-Kriging would perform best given the characteristics of the available data. In this chapter, the focus has changed to a setting where the only available information to base any technique selection is the available budget B and the cost ratio C_r. Based on these two pieces of information, a decision needs to be made on which technique to use. The hope is that the technique itself will learn how to adapt based on the characteristics of the function pair as more data is obtained. Therefore the instances being analysed are separated into 16 groups based on B and C_r, and the distribution of the performance of each of the techniques is shown via box-plots. Figures 5.2 and 5.3 show this performance in terms of correlation and error, respectively. Each box-plot shows the distribution of a technique over 5 repetitions for 321 function pairs, totalling 1605 data points. Rather than showing the absolute performance for each technique, for each instance and repetition, the accuracy of the best performing technique is determined, and the difference between this accuracy to each of the other technique's accuracy is calculated. This difference is shown in the y-axis of both figures. Reliably accurate techniques are therefore identified by box-plots with smaller spreads. Similarly, a median relative performance (shown by the thick black line) close to zero denotes that a technique that achieves a near-best performance among its competitors a large proportion of the time. Note also that a technique could be often near optimal a large proportion of the time but also have a large spread, indicating that the technique is not always reliable.

Let us first discuss the performance in terms of correlation. Note that this measure indicates how well the constructed models replicate the behaviour of f_h. Recall that Rules-based Co-Kriging uses feature-based guidelines to choose whether to ignore the

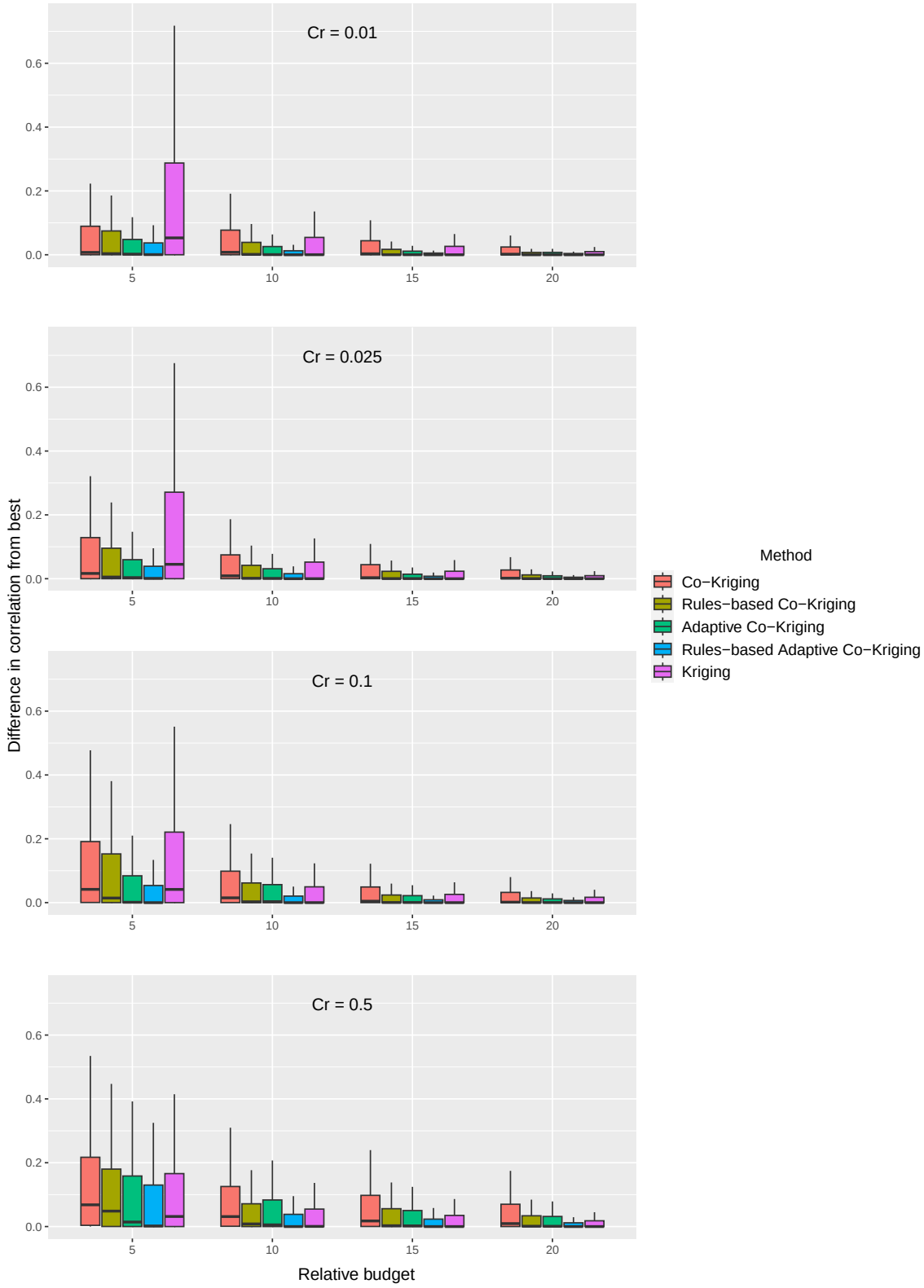


FIGURE 5.2: Distribution of the performance of each of the five techniques in terms of correlation for varying cost ratios and relative budget. The y-axis denotes the difference from the best performing technique for each repetition of each instance.

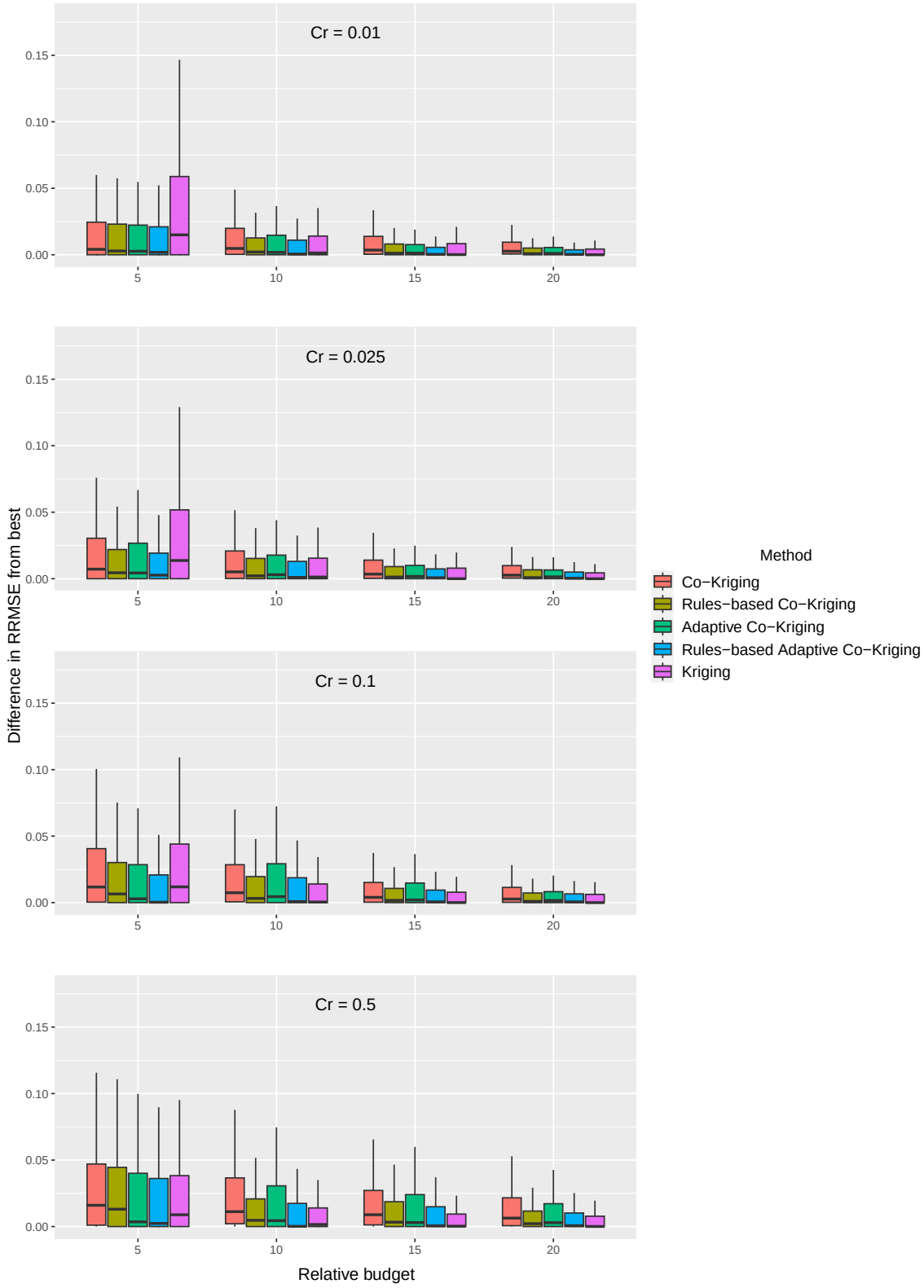


FIGURE 5.3: Distribution of the performance of each of the five techniques in terms of error for varying cost ratios and relative budgets. The y-axis denotes the difference from the best performing technique for each repetition of each instance.

low-fidelity source, Adaptive Co-Kriging chooses only one source to sample at each iteration, and Rules-based Adaptive Co-Kriging employs both techniques. It appears that both the usage of the guidelines and the adaptive sampling consistently lead to an improvement over classical Co-Kriging. The plot indicates that if only one can be chosen, it is best to use adaptive sampling. As shown by the box-plots however, the combination of both approaches in the form of Rules-based Adaptive Co-Kriging outperforms every other approach, including Kriging, for every cost ratio and budget pair. In particular, for $B \geq 15$ and $C_r \leq 10$, the distribution of this technique's relative performance is very close to optimal. As the relative budget decreases this technique struggles to remain consistent in choosing the best model. This is the result of having less data to characterise both sources, and therefore making a model choice based only on an approximated characterisation. Even in this situation, however, the performance distribution is superior to every other technique.

Looking next at the error-based performance, the conclusions that can be drawn from the displayed data vary slightly. Unlike when assessing performance via correlation, here two techniques outperform all others, namely Rules-based Adaptive Co-Kriging and traditional Kriging. Recall that this technique simply chooses to always ignore the low-fidelity source. It appears this is the best approach when f_l is not very cheap (i.e. $C_r = 0.5$), except for when very little high-fidelity data is available (i.e. $B = 5d$). At the same time, Rules-based Adaptive Co-Kriging seems to be the best choice when the low-fidelity data is very cheap (i.e. $C_r = 0.01$) or very little high-fidelity data is available (i.e. $B = 5d$). In every other case, it appears both techniques have a similar performance. Unlike with correlation as the performance measure, however, the spread of both of these techniques is not as narrow, indicating that for both techniques there is a noticeable amount of instances for which the performance is not extremely close to optimal. For Kriging, this indicates that there are situations where ignoring the low-fidelity source is counter-productive. For Rules-based Adaptive Co-Kriging, this indicates that the choice of model is not always accurate. This is perhaps unsurprising, however, as the guidelines this technique is based on were developed by analysing the performance of Kriging and Co-Kriging in terms of correlation, and not error. More accurate guidelines in terms of model error can likely be developed if a second characterisation of harmful data sources is conducted using error as the performance measure.

Finally, it is important to also discuss the impact of the initial DoE on Kriging performance. As discussed in Section 5.1.1, for every instance the same initial high- and low-fidelity sample is collected for all techniques. This is done so that, among others, Kriging, Rules-based Co-Kriging and Rules-based Adaptive Co-Kriging will have the same performance if the guidelines deem the low-fidelity source to be harmful. This allows for the comparison of the model choice mechanic, rather than the overall technique. For Kriging however this initial DoE is not an optimal decision, as some of the budget is spent on gathering low-fidelity data which is then not used. The alternative when using Kriging would be to never collect any low-fidelity data including in the DoE phase, and therefore to blindly assume that the low-fidelity source should not be used. Doing so would indicate some prior knowledge on the practitioner's behalf of the characteristics of the available information sources. When this knowledge is not available, it seems a much safer choice in terms of model correlation is to use Rules-based Adaptive Co-Kriging; this will also likely be the case in terms of model error if more appropriate guidelines are first devised.

5.3 Conclusion

This chapter has shifted the focus of previous chapters away from a simple model choice based on available data. More complex techniques bound only by budget constraints have instead been investigated. The techniques that tackle instances in this chapter are given the freedom to choose how to allocate the given budget between an initial DoE, how to query a constructed model to select further sampling locations, and how to select which source to sample. By employing both a newly proposed acquisition function and the newly developed guidelines from Chapter 4, the technique Rules-based Adaptive Co-Kriging has been developed which is much safer than traditional Kriging or Co-Kriging as it shows a much more narrow performance spread, particularly in terms of model correlation with the high-fidelity source.

The key decisions to be made when generating an initial DoE however have not been investigated in this chapter, and are left to future work. The proportion of the total budget to be spent in the initial sample, and how this proportion should be spent among sources must have an impact on the overall performance of a technique. Finding the right balance however is far from trivial, as a larger initial sample can lead to more accurate

models and therefore more accurate model guided sampling, but a smaller initial sample leaves a higher remaining budget to be guided by what may be a much less accurate model. This is also likely to affect the performance of the newly proposed Rules-based Adaptive Co-Kriging, as the choice of model is based on the characterisation of the low-fidelity source. Not having enough data for this characterisation might lead to the wrong choice of model, but an early accurate diagnostic of a source as harmful can lead to a smaller proportion of the budget being wasted on low-fidelity data if it will not be used when training a model.

The performance of the newly developed techniques has highlighted the strong potential benefits of better understanding the impact of information source characteristics on overall model accuracy. This is a markedly different approach to almost all current techniques in the field which aim to develop techniques to internally modulate the impact of each of the information sources based on their quality. The hope with these techniques is that for harmful data sources, the impact of "bad" low-fidelity sources is reduced internally by the model. Traditional Co-Kriging can do this via the hyper-parameter ρ which internally scales the low-fidelity data. Here the proposed approach is to decouple this decision from the model by simply analysing the features of the dataset itself. The results of Rules-based Adaptive Co-Kriging speak to the benefits of this approach.

Finally, it is worth stressing that the techniques explored here do not represent the end-goal of this avenue of research. This chapter illustrates the benefits of the insights gleaned in this thesis, and the power of analysing information sources rather than over-relying on surrogate models to incorporate all available information without prior assessment. The potential for future improvements in this regard is further discussed in the closing chapter.

Chapter 6

Concluding remarks

The high cost of assessing the performance of certain industrial products, coupled with the existence of multiple information sources of varying cost and accuracy, provide excellent grounds for surrogate modelling techniques to excel. These techniques however are susceptible to under-performing when blindly combining all available information into a single model, as the usage of certain data sources might be counter-productive. With this in mind, this thesis has been mainly motivated by the importance of the notion of trust in an information source.

6.1 Summary of contributions

Chapter 2 has provided the required background into surrogate modelling techniques for Mf-EBB problems. Three kinds of Mf-EBB problems have been defined, namely model creation with a fixed sample, model creation with a sample budget, and function optimisation with a sample budget. By using a novel illustrative example with Australian elevation data, the dangers of not conducting any assessment of the data being used have been highlighted. This example has helped motivate the work of this thesis as low-fidelity information from the same source can vary in usefulness, depending on the data-gathering process and the objective of the surrogate modelling technique being employed.

Chapter 3 has then taken a more global approach to the question of trust in an information source. The focus of this chapter has been to highlight the need for a heterogeneous

set of benchmarks, and a more diverse set of features. This has been achieved by investigating Bf-EBB problems, and running the same model performance analysis twice. By showcasing the large variation in results when using only classical instances, and when using classical and newly proposed instances, the need for a more thorough understanding of "good" and "bad" low-fidelity sources has been highlighted. This chapter has also indicated the potential benefits of analysing sources in terms of not only their overall accuracy, but of the frequency with which these sources are locally accurate.

Chapter 4 has then used the newly proposed instances and features to conduct a thorough computational study to characterise harmful data sources. This characterisation has been conducted with the clear reminder from the previous chapter of the importance of removing bias from the study. A recently developed instance filtering technique has been employed to this end to generate multiple sets of function pairs that are objectively diverse. The implementation of these unbiased benchmarks and the associated metadata are now available on GitHub [92] and FigShare [93], respectively, for future practitioners to assess new techniques. These diverse benchmarks have then been used to analyse when a two-source model (i.e. Co-Kriging) outperforms a single-source model (i.e. Kriging). The aim has been not only to obtain simple, easy to use rules, but to do so in a way that is intuitive enough to be trusted by industrial practitioners. Instance Space Analysis has been used here due to its strong visualisation aspect, leading to intuitive insights into model performance. Furthermore, the analysis has been performed using only the limited data available to train models, a first in the literature. This means the developed guidelines can be directly applied to industrial problems.

Finally, Chapter 5 has applied the findings of previous chapters to the setting with budget constraints. Here some of the data used to train the surrogate models has been gathered by the surrogate models themselves, increasing the complexity of the techniques being employed. This chapter has not aimed to present final adaptive techniques however, but rather to showcase the potential for great performance improvements when coupling prior data assessment with surrogate modelling techniques. In particular, the method Rules-based Adaptive Co-Kriging which both assesses whether the low-fidelity source is harmful, and only chooses one source to sample at each iteration, has been shown to outperform both classical Kriging and Co-Kriging in terms of model correlation. This method has also shown promise in terms of model error, despite using guidelines that were not developed specifically for this purpose.

6.2 Future directions

Many avenues for future work exist. Perhaps the most evident ones are the extension of the results found in this thesis to other facets of classical Mf-EBB problems. Most of the work conducted in this thesis has examined the usefulness of a single low-fidelity source by focusing on Bf-EBB problems. Many Mf-EBB problems however have more than one low-fidelity source available. Making an accurate assessment of which subset of available information sources should be combined into a single model remains an unanswered challenge. It is also worth investigating the usefulness of the provided guidelines on other classical Bf-EBB models. Whilst this has not been tested in this work, the internally additive nature of Co-Kriging makes it quite likely that other common surrogate modelling approaches such as Co-RBFs or Hierarchical Kriging will benefit from similar if not identical guidelines. Another classical aspect that has not been investigated in this work is the optimisation of the high-fidelity source. Indeed, the focus has always been on the performance of the surrogate modelling methods in terms of final model accuracy. Whilst it is likely many of the ideas found in this work will also be valuable during optimisation, an assessment of trust of information sources within an optimisation framework still needs to be conducted.

In terms of application to industrial problems in general, and to testing processes at Boeing Australia in particular, there is still a gap to be bridged before the results found in this thesis can be applied. Many simplifying assumptions have been made here to isolate and investigate certain aspects of Mf-EBB problems. Perhaps most notable is the fact that in many applications, no single source of information is labelled as the truth, and the true behaviour of a particular prototype needs to instead be gleaned from a collection of information sources. Other more complex characteristics include stochastic outputs, as well as information sources with restricted or non-identical domains. It is also possible for this field to be exposed to *grey-boxes*, where practitioners do have some information available on each of the information sources. This might include a pre-existing notion of trust in each of the sources, particularly in some regions of the sample space. The existence of this information would imply the requirement of a "human-in-the-loop" to aid the identification of harmful information. Some of the requirements of the trained model might also include more than just a performance prediction, such as the need for a set of coefficients to provide engineers with an intuition of how different

forces affect the constructed prototype. Whilst all of these considerations are left for future work, the concepts discussed in this thesis have proved another step forward in the ongoing conversation of best practices in industrial processes.

Finally and perhaps most excitingly, the work presented in this thesis has shown the power of characterising harmful data sources to establish trust. To do so, an "all or nothing" approach has been taken. That is, the chosen assessment has been whether using data from a source is a good or a bad decision. By developing guidelines that aim to correctly choose a single-source or two-source model, techniques that are accurate more often can be devised. This approach however can never hope to outperform classic techniques on any given instance. A finer approach that has not yet been examined here, or in the literature, would consist of processing the data in a way that removes only the harmful components before being used in model construction. Whilst it is of course possible that all data from a low-fidelity source is beneficial, or that it is all harmful, it is far more likely that the qualities change based on the region of the sampling domain where they were obtained. By continuing on the path taken in this work and further examining how to filter out harmful information, it is possible that even more powerful techniques can be devised.

Appendix A

Theoretical derivation of Kriging and Co-Kriging

This Appendix provides the full theoretical derivation of both Kriging and Co-Kriging, as presented by Jones [3] and Forrester et al. [38], respectively, adapted notation-wise to that used in the rest of the thesis.

A.1 Kriging

As presented in Section 2.2.1.3, Kriging is a single-source model trained with data obtained from a set of locations $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and the set of objective function values $\mathbf{y} = \{y_1, \ldots, y_n\} = \{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)\}$ at those locations. The derivation of the model assumes the function samples made so far are realisations of random normal variables $Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_n)$ with mean μ and variance σ^2. The multivariate random variable $\mathbf{Y} = [Y(\mathbf{x}_1) \ldots Y(\mathbf{x}_n)]$ is assumed to have the distribution $\mathbf{Y} \sim N(\mathbf{1}\mu, \sigma^2\mathbf{R})$, with

$$\mathbf{R}_{i,j} = R(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{ -\sum_{k=1}^{d} \theta_k |\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}|^{p_k} \right\},$$

where $\mathbf{x}^{(k)}$ denotes the kth coordinate of point $\mathbf{x}$. The probability of having observed the values $\mathbf{y}$ given the process $\mathbf{Y}$ is given by

$$f_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{(2\pi)^{\frac{n}{2}}\,(\sigma^2)^{\frac{n}{2}}\,|\mathbf{R}|^{\frac{1}{2}}} \exp\left\{-\frac{(\mathbf{y}-\mathbf{1}\mu)^\top \mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2}\right\}.$$

The hyperparameter values for $\mu, \sigma^2, \theta_1, \ldots, \theta_d$ and $p_1, \ldots, p_d$ are chosen to be the MLEs. In practice, first the (monotonically increasing) log function is applied, leading to the maximisation of

$$\log f_{\mathbf{Y}}(\mathbf{y}) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2}\log|\mathbf{R}| - \frac{(\mathbf{y}-\mathbf{1}\mu)^\top \mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2}. \tag{A.1}$$

An analytical solution can be found for the MLEs $\hat{\mu}$ and $\hat{\sigma}^2$ of μ and σ^2, respectively. Setting the derivative of the log-likelihood with respect to μ and σ^2 to zero and solving gives the following

$$\begin{aligned}
0 &= -\mathbf{y}^\top \mathbf{R}^{-1}\mathbf{1} - \mathbf{1}^\top \mathbf{R}^{-1}\mathbf{y} + 2\hat{\mu}\mathbf{1}^\top \mathbf{R}^{-1}\mathbf{1}, \\
&= -2\mathbf{1}^\top \mathbf{R}^{-1}\mathbf{y} + 2\hat{\mu}\mathbf{1}^\top \mathbf{R}^{-1}\mathbf{1} \quad \text{as } \mathbf{y}^\top \mathbf{R}^{-1}\mathbf{1} = \mathbf{1}^\top \mathbf{R}^{-1}\mathbf{y}, \\
\Rightarrow \hat{\mu} &= \frac{\mathbf{1}^\top \mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^\top \mathbf{R}^{-1}\mathbf{1}}, \\
0 &= -\frac{n}{2\hat{\sigma}^2} + \frac{(\mathbf{y}-\mathbf{1}\mu)^\top \mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\,(\hat{\sigma}^2)^2}, \\
\Rightarrow \hat{\sigma}^2 &= \frac{(\mathbf{y}-\mathbf{1}\mu)^\top \mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{n}.
\end{aligned}$$

Substituting this solution into the log-likelihood and removing constant terms leads to the so-called concentrated log-likelihood

$$-\frac{n}{2}\log(\hat{\sigma}^2) - \frac{1}{2}\log|\mathbf{R}|.$$

Finding the remainder of the hyperparameters must be done numerically as no analytical solution exists. This consists of an auxiliary optimisation problem where the concentrated log-likelihood is maximised with respect to $\theta_1, \ldots, \theta_d$ and $p_1, \ldots, p_d$; Appendix B defines some of the available algorithms for this purpose. It is recommended when training a Kriging model to first preprocess the training data by linearly scaling it to lie

within the hypercube $[0,1]^d$, and to use the bounds $10^{-3} \leq \theta_i \leq 10^3$ and $0.1 \leq p_i \leq 2.0$ when finding the MLEs.

Having trained the hyperparameters, MLEs can again be used to derive the model prediction $s^{krig}(\mathbf{x})$ at a location $\mathbf{x}$. First define the vector $\mathbf{r}(\mathbf{x}) = [R(\mathbf{x}, \mathbf{x}_1) \dots R(\mathbf{x}, \mathbf{x}_n)]^\top$. Then, assuming the training dataset consists of the points $\{\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x}\}$ with objective function values $\{y_1, \dots, y_n, s^{krig}(\mathbf{x})\}$, the new correlation matrix $\mathbf{R}_\mathbf{x}$ is given by

$$\mathbf{R}_\mathbf{x} = \begin{bmatrix} \mathbf{R} & \mathbf{r}(\mathbf{x}) \\ \mathbf{r}(\mathbf{x})^\top & 1 \end{bmatrix}.$$

The MLE of $s^{krig}(\mathbf{x})$ can then be found by maximising the log-likelihood with respect to $s^{krig}(\mathbf{x})$. Substituting the new data in the log-likelihood given in Equation A.1, the only element containing the term $s^{krig}(\mathbf{x})$ is

$$\begin{aligned} &-\frac{1}{2\hat{\sigma}^2} \begin{bmatrix} \mathbf{y} - \mathbf{1}\hat{\mu} \\ s^{krig}(\mathbf{x}) - \hat{\mu} \end{bmatrix}^\top \begin{bmatrix} \mathbf{R} & \mathbf{r}(\mathbf{x}) \\ \mathbf{r}(\mathbf{x})^\top & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y} - \mathbf{1}\hat{\mu} \\ s^{krig}(\mathbf{x}) - \hat{\mu} \end{bmatrix} \\ =&-\frac{1}{2\hat{\sigma}^2} \begin{bmatrix} \mathbf{y} - \mathbf{1}\hat{\mu} \\ s^{krig}(\mathbf{x}) - \hat{\mu} \end{bmatrix}^\top \begin{bmatrix} \mathbf{R}^{-1} + \alpha \mathbf{R}^{-1}\mathbf{r}(\mathbf{x})\mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1} & -\alpha \mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) \\ \alpha \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1} & \alpha \end{bmatrix} \begin{bmatrix} \mathbf{y} - \mathbf{1}\hat{\mu} \\ s^{krig}(\mathbf{x}) - \hat{\mu} \end{bmatrix}, \end{aligned}$$

using block matrix inversion and with $\alpha = \left(1 - \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1}\mathbf{r}(\mathbf{x})\right)^{-1}$. Expanding the expression and again only taking the terms which contain $s^{krig}(\mathbf{x})$ gives

$$-\frac{\alpha}{2\hat{\sigma}^2}\left(s^{krig}(\mathbf{x}) - \hat{\mu}\right)^2 + \frac{\alpha \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})}{\hat{\sigma}^2}\left(s^{krig}(\mathbf{x}) - \hat{\mu}\right), \tag{A.2}$$

which once again taking the derivative with respect to $s^{krig}(\mathbf{x})$ and setting it to zero results in the predictor

$$s^{krig}(\mathbf{x}) = \hat{\mu} + \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}).$$

In addition to the prediction $s^{krig}(\mathbf{x})$, a Kriging model also provides a prediction error metric $v^{krig}(\mathbf{x})$, which intuitively is inversely related to the curvature of the log-likelihood

as a one-dimensional function of $s^{krig}(\mathbf{x})$. That is, if locally different values of $s^{krig}(\mathbf{x})$ lead to much smaller log-likelihoods, the MLE prediction should have a very small error and the curvature of the log-likelihood is very large. Conversely, if different values of $s^{krig}(\mathbf{x})$ lead to similar log-likelihoods, then the prediction is more prone to error and the curvature of the log-likelihood function is very small. The error metric is therefore given by absolute of the inverse of the second derivative of Equation A.2 with respect to $s^{krig}(\mathbf{x})$, namely

$$v^{krig}(\mathbf{x}) \approx \hat{\sigma}^2 \left[1 - \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1} \mathbf{r}(\mathbf{x})\right].$$

The real mean-squared error of the predictor contains an additional term $\hat{\sigma}^2 \frac{(1-\mathbf{1}^\top \mathbf{R}^{-1}\mathbf{r}(\mathbf{x}))^2}{\mathbf{1}^\top \mathbf{R}^{-1}\mathbf{1}}$, however in practice this term is so small that it can be safely neglected.

A.2 Co-Kriging

As presented in Section 2.2.2.1, Co-Kriging models are trained using data from a high- and a low-fidelity source. That is the training data used consists of a set of n_l locations $\mathbf{X}_l = \{\mathbf{x}_1^l, \mathbf{x}_2^l, \ldots, \mathbf{x}_{n_l}^l\}$ with known low-fidelity function values $\mathbf{y}_l = \{y_1^l, \ldots, y_{n_l}^l\} = \{f_l(\mathbf{x}_1), \ldots, f_l(\mathbf{x}_{n_l})\}$, and a set of n_h locations $\mathbf{X}_h = \{\mathbf{x}_1^h, \mathbf{x}_2^h, \ldots, \mathbf{x}_{n_h}^h\}$ with known high-fidelity function values $\mathbf{y}_h = \{y_1^h, \ldots, y_{n_h}^h\} = \{f_h(\mathbf{x}_1), \ldots, f_h(\mathbf{x}_{n_h})\}$. The Co-Kriging framework models the low- and high-fidelity responses as realisations of a multivariate random variable

$$\mathbf{Y} = (\mathbf{Y}_l(\mathbf{X}_l), \mathbf{Y}_h(\mathbf{X}_h)) = (Y_l(\mathbf{x}_1^l), \ldots, Y_l(\mathbf{x}_{n_l}^l), Y_h(\mathbf{x}_1^h), \ldots, Y_h(\mathbf{x}_{n_h}^h)).$$

The multivariate random variable $\mathbf{Y}_l(\mathbf{X}_l)$ is treated as a multivariate normal random variable with distribution $\sim N\left(\mu_l, \sigma_l^2 \mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)\right)$. The multivariate random variable $\mathbf{Y}_h(\mathbf{X}_h)$ is represented by a scaling of ρ of the response of the cheap expensive function $\mathbf{Y}_l(\mathbf{X}_h)$ plus a new Gaussian process $\mathbf{Y}_{diff}(\mathbf{X}_h)$ which models the difference between the cheap and expensive objective functions, that is

$$\mathbf{Y}_h(\mathbf{X}_h) = \rho \mathbf{Y}_l(\mathbf{X}_h) + \mathbf{Y}_{diff}(\mathbf{X}_h).$$

The random variable $\mathbf{Y}_{diff}$ is also treated as a multivariate normal random variable with distribution $\sim N\left(\mu_{diff}, \sigma^2_{diff}\mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)\right)$. It is assumed that $\mathbf{Y}_l$ and $\mathbf{Y}_{diff}$ are independent. Recall the following matrix definitions

$$\begin{aligned}
\mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)_{i,j} &= R_l(\mathbf{x}^l_i, \mathbf{x}^l_j) \qquad 1 \le i, j \le n_l, \\
\mathbf{R}_l(\mathbf{X}_h, \mathbf{X}_l)_{i,j} &= R_l(\mathbf{x}^h_i, \mathbf{x}^l_j) \qquad 1 \le i \le n_h \quad 1 \le j \le n_l, \\
\mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_h)_{i,j} &= R_l(\mathbf{x}^l_i, \mathbf{x}^h_j) \qquad 1 \le i \le n_l \quad 1 \le j \le n_h, \\
\mathbf{R}_l(\mathbf{X}_h, \mathbf{X}_h)_{i,j} &= R_l(\mathbf{x}^h_i, \mathbf{x}^h_j) \qquad 1 \le i, j \le n_h, \\
\mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)_{i,j} &= R_{diff}(\mathbf{x}^h_i, \mathbf{x}^h_j) \qquad 1 \le i, j \le n_h,
\end{aligned}$$

$$\text{where } R_l(\mathbf{x}_1, \mathbf{x}_2) = \exp\left\{-\sum_{k=1}^{d} \theta^l_k |\mathbf{x}^{(k)}_1 - \mathbf{x}^{(k)}_2|^{p^l_k}\right\},$$

$$R_{diff}(\mathbf{x}_1, \mathbf{x}_2) = \exp\left\{-\sum_{k=1}^{d} \theta^{diff}_k |\mathbf{x}^{(k)}_1 - \mathbf{x}^{(k)}_2|^{p^{diff}_k}\right\}.$$

Training the hyperparameters of a Co-Kriging model is a two-step process. First note the likelihood of having observed the values $\mathbf{y}_l$ given the process $\mathbf{Y}_l(\mathbf{X}_l)$ is given by

$$f_{\mathbf{Y}_l(\mathbf{X}_l)}(\mathbf{y}_l) = \frac{1}{(2\pi)^{\frac{n_l}{2}} \left(\sigma^2_l\right)^{\frac{n_l}{2}} |\mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)|^{\frac{1}{2}}} \exp\left\{-\frac{(\mathbf{y}_l - \mathbf{1}\mu_l)^\top \mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)^{-1}(\mathbf{y}_l - \mathbf{1}\mu_l)}{2\sigma^2_l}\right\}.$$

Similarly to training a Kriging model, the MLEs $\hat{\mu}_l, \hat{\sigma}^2_l, \hat{\theta}^l_1, \ldots, \hat{\theta}^l_d, \hat{p}^l_1, \ldots, \hat{p}^l_d$ are found by numerically maximising the concentrated log-likelihood

$$\max_{\theta^l_1, \ldots, \theta^l_d, p^l_1, \ldots, p^l_d} -\frac{n_l}{2}\log(\hat{\sigma}^2_l) - \frac{1}{2}\log(|\mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)|),$$

where

$$\begin{aligned}
\hat{\mu}_l &= \frac{\mathbf{1}^\top \mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)^{-1}\mathbf{y}_l}{\mathbf{1}^\top \mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)^{-1}\mathbf{1}}, \\
\hat{\sigma}^2_l &= \frac{(\mathbf{y}_l - \mathbf{1}\hat{\mu}_l)^\top \mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)^{-1}(\mathbf{y}_l - \mathbf{1}\hat{\mu}_l)}{n_l}.
\end{aligned}$$

Next the vector $\mathbf{d} = \mathbf{y}_h - \rho\mathbf{y}_l(\mathbf{X}_h)$ is defined, where $\mathbf{y}_h = (f_h(\mathbf{x}^h_1), \ldots, f_h(\mathbf{x}^h_{n_h}))^\top$, and $\mathbf{y}_l(\mathbf{X}_h)_i$ is $f_l(\mathbf{x}^h_i)$ if this value is known, and otherwise it is $\hat{\mu}_l + \mathbf{r}_l(\mathbf{x})^\top \mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l)^{-1}(\mathbf{y}_l - \mathbf{1}\hat{\mu}_l)$, with $\mathbf{r}_l(\mathbf{x}) = (R_l(\mathbf{x}, \mathbf{x}^l_1), \ldots, R_l(\mathbf{x}, \mathbf{x}^l_{n_l}))^\top$. That is, if a point has not been evaluated

by f_l yet, the trained Kriging predictor of the low-fidelity source is used instead. The likelihood of having observed the values $\mathbf{d}$ given the process $\mathbf{Y}_{diff}(\mathbf{X}_h)$ is given by

$$f_{\mathbf{Y}_{diff}(\mathbf{X}_h)}(\mathbf{d}) = \frac{1}{(2\pi)^{\frac{n_h}{2}} \left(\sigma^2_{diff}\right)^{\frac{n_h}{2}} |\mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)|^{\frac{1}{2}}} \cdot \exp\left\{ -\frac{(\mathbf{d} - \mathbf{1}\mu_{diff})^\top \mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)^{-1}(\mathbf{d} - \mathbf{1}\mu_{diff})}{2\sigma^2_{diff}} \right\}.$$

Once again, the MLEs $\hat{\rho}, \hat{\mu}_{diff}, \hat{\sigma}^2_{diff}, \hat{\theta}_1^{diff}, \ldots, \hat{\theta}_d^{diff}, \hat{p}_1^{diff}, \ldots, \hat{p}_d^{diff}$ are found by numerically maximising the concentrated log-likelihood

$$\max_{\rho, \theta_1^{diff}, \ldots, \theta_d^{diff}, p_1^{diff}, \ldots, p_d^{diff}} -\frac{n_h}{2} \log(\hat{\sigma}^2_{diff}) - \frac{1}{2} \log(|\mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)|),$$

where

$$\hat{\mu}_{diff} = \frac{\mathbf{1}^\top \mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)^{-1}\mathbf{d}}{\mathbf{1}^\top \mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)^{-1}\mathbf{1}},$$
$$\hat{\sigma}^2_{diff} = \frac{(\mathbf{d} - \mathbf{1}\hat{\mu}_{diff})^\top \mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h)^{-1}(\mathbf{d} - \mathbf{1}\hat{\mu}_{diff})}{n_h}.$$

Note that the value of $\hat{\rho}$ impacts the vector $\mathbf{d}$ which in turn has an impact on the log-likelihood. It is again recommended when finding the MLEs of a Co-Kriging model to first preprocess the training data by linearly scaling it to lie within the hypercube $[0, 1]^d$ and the output values $\mathbf{y}_l, \mathbf{y}_h$ to lie within $[0, 1]$, and to use the bounds $10^{-3} \leq \theta_i^l, \theta_i^{diff}, \leq 10^3$, $0.1 \leq p_i^l, p_i^{diff} \leq 2.0$ and $-100 \leq \rho \leq 100$. Finally, the multivariate random variable $\mathbf{Y} = (\mathbf{Y}_l(\mathbf{X}_l), \mathbf{Y}_h(\mathbf{X}_h))$ has the distribution $\sim N(\mu, \mathbf{C})$, with

$$\mathbf{C} = \begin{bmatrix} \hat{\sigma}_l^2 \mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_l) & \hat{\rho}\hat{\sigma}_l^2 \mathbf{R}_l(\mathbf{X}_l, \mathbf{X}_h) \\ \hat{\rho}\hat{\sigma}_l^2 \mathbf{R}_l(\mathbf{X}_h, \mathbf{X}_l) & \hat{\rho}^2\hat{\sigma}_l^2 \mathbf{R}_l(\mathbf{X}_h, \mathbf{X}_h) + \hat{\sigma}^2_{diff}\mathbf{R}_{diff}(\mathbf{X}_h, \mathbf{X}_h) \end{bmatrix}.$$

The likelihood of having observed the values $\mathbf{y} = (\mathbf{y}_l, \mathbf{y}_h)$ is given by

$$f_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{(2\pi)^{\frac{n_l+n_h}{2}} |\mathbf{C}|^{\frac{1}{2}}} \exp\left\{ -\frac{(\mathbf{y} - \mathbf{1}\mu)^\top \mathbf{C}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2} \right\},$$

with the log-likelihood being

$$\log f_{\mathbf{Y}}(\mathbf{y}) = -\frac{n_l + n_h}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{C}| - \frac{(\mathbf{y} - \mathbf{1}\mu)^\top \mathbf{C}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2}. \tag{A.3}$$

The last MLE $\hat{\mu}$ of μ can once again me analytically derived by maximising this log-likelihood and is given by

$$\hat{\mu} = \frac{\mathbf{1}^\top \mathbf{C}^{-1}\mathbf{y}}{\mathbf{1}^\top \mathbf{C}^{-1}\mathbf{1}}.$$

Having found the MLEs of the hyperparameters, MLEs are once again used to derive the model prediction $s^{cokrig}(\mathbf{x})$ of the value of f_h at a location $\mathbf{x}$. First define the following vector

$$\mathbf{c}(\mathbf{x}) = \begin{bmatrix} \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}_1^l, \mathbf{x}) \\ \vdots \\ \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}_{n_l}^l, \mathbf{x}) \\ \\ \hat{\rho}^2\hat{\sigma}_l^2 R_l(\mathbf{x}_1^h, \mathbf{x}) + \hat{\sigma}_{diff}^2 R_{diff}(\mathbf{x}_1^h, \mathbf{x}) \\ \vdots \\ \hat{\rho}^2\hat{\sigma}_l^2 R_l(\mathbf{x}_{n_h}^h, \mathbf{x}) + \hat{\sigma}_{diff}^2 R_{diff}(\mathbf{x}_{n_h}^h, \mathbf{x}) \end{bmatrix}.$$

Now, assuming the training dataset consists of the low-fidelity observations $\mathbf{y}_l$ at locations $\mathbf{X}_l$, and the high-fidelity observations $\{y_1^h, \ldots, y_{n_h}^h, s^{cokrig}(\mathbf{x})\}$ at locations $\{\mathbf{x}_1^h, \ldots, \mathbf{x}_{n_h}^h, \mathbf{x}\}$, the new correlation matrix $\mathbf{C}_{\mathbf{x}}$ is given by

$$\mathbf{C}_{\mathbf{x}} = \begin{bmatrix} \mathbf{C} & \mathbf{c}(\mathbf{x}) \\ \mathbf{c}(\mathbf{x})^\top & \hat{\rho}\hat{\sigma}_l^2 + \hat{\sigma}_{diff}^2 \end{bmatrix}.$$

The MLE of $s^{cokrig}(\mathbf{x})$ can be then found by maximising the log-likelihood with respect to $s^{cokrig}(\mathbf{x})$. Substituting the new data in the log-likelihood given in Equation A.3, the

only element containing the term $s^{cokrig}(\mathbf{x})$ is

$$
\begin{aligned}
&-\frac{1}{2}\begin{bmatrix} \mathbf{y}-\mathbf{1}\hat{\mu} \\ s^{cokrig}(\mathbf{x})-\hat{\mu} \end{bmatrix}^{\top} \begin{bmatrix} \mathbf{C} & \mathbf{c}(\mathbf{x}) \\ \mathbf{c}(\mathbf{x})^{\top} & \hat{\rho}\hat{\sigma}_l^2+\hat{\sigma}_{diff}^2 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y}-\mathbf{1}\hat{\mu} \\ s^{cokrig}(\mathbf{x})-\hat{\mu} \end{bmatrix} \\
=&-\frac{1}{2}\begin{bmatrix} \mathbf{y}-\mathbf{1}\hat{\mu} \\ s^{cokrig}(\mathbf{x})-\hat{\mu} \end{bmatrix}^{\top} \begin{bmatrix} \mathbf{C}^{-1}+\alpha\mathbf{C}^{-1}\mathbf{c}(\mathbf{x})\mathbf{c}(\mathbf{x})^{\top}\mathbf{C}^{-1} & -\alpha\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}) \\ \alpha\mathbf{c}(\mathbf{x})^{\top}\mathbf{C}^{-1} & \alpha \end{bmatrix} \begin{bmatrix} \mathbf{y}-\mathbf{1}\hat{\mu} \\ s^{cokrig}(\mathbf{x})-\hat{\mu} \end{bmatrix},
\end{aligned}
$$

using block matrix inversion and with $\alpha = \left(\hat{\rho}\hat{\sigma}_l^2+\hat{\sigma}_{diff}^2-\mathbf{c}(\mathbf{x})^{\top}\mathbf{C}^{-1}\mathbf{c}(\mathbf{x})\right)^{-1}$. Expanding the expression and again only taking the terms which contain $s^{cokrig}(\mathbf{x})$ gives

$$
-\frac{\alpha}{2}\left(s^{cokrig}(\mathbf{x})-\hat{\mu}\right)^2+\alpha\mathbf{c}(\mathbf{x})^{\top}\mathbf{C}^{-1}(\mathbf{y}-\mathbf{1}\hat{\mu})\left(s^{cokrig}(\mathbf{x})-\hat{\mu}\right). \tag{A.4}
$$

Once again, taking the derivative with respect to $s^{cokrig}(\mathbf{x})$ and setting it to zero results in the predictor

$$
s^{cokrig}(\mathbf{x}) = \hat{\mu}+\mathbf{c}(\mathbf{x})^{\top}\mathbf{C}^{-1}(\mathbf{y}-\mathbf{1}\hat{\mu}).
$$

Similarly to Kriging, a Co-Kriging model also provides a prediction error metric $v^{cokrig}(\mathbf{x})$. This is again taken to be inversely related to the curvature of the log-likelihood as a one-dimensional function of $s^{cokrig}(\mathbf{x})$. This error is therefore given by the absolute of the inverse of the second derivative of Equation A.4 with respect to s^{cokrig}, namely

$$
v^{cokrig} \approx \hat{\rho}\hat{\sigma}_l^2+\hat{\sigma}_{diff}^2-\mathbf{c}(\mathbf{x})^{\top}\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}).
$$

Once again, the real mean-squared error of the predictor contains an additional term $\frac{\left(1-\mathbf{1}^{\top}\mathbf{C}^{-1}\mathbf{c}(\mathbf{x})\right)}{\mathbf{c}(\mathbf{x})^{\top}\mathbf{C}^{-1}\mathbf{c}(\mathbf{x})}$ which in practice is very small as is safely neglected.

Appendix B

Auxiliary optimisation algorithms

When employing surrogate modelling methods, auxiliary optimisation problems often arise. This can be the case when finding optimal hyperparameters when training Kriging and Co-Kriging models, or when finding further sample sites according to a chosen acquisition function. Considerable work has been dedicated to the development of algorithms to find optimal hyperparameters of Kriging models [33, 34], particularly focusing on strategies that deliver good results with a relatively small sample budget i.e. 1,000 iterations. The development of these techniques is important because evaluating the log-likelihood of Kriging and Co-Kriging models can become computationally expensive. Techniques of this type however tend to contain a large number of tuning parameters and might be hard to implement. As the focus of this thesis is not the absolute performance of Kriging and Co-Kriging models, but rather their comparative performance, simpler techniques that are easy to implement and have few tuning parameters have been used with a large sample budget of 10,000. Specifically, the results of Chapter 3 are obtained by using Accelerated Random Search (ARS) [35] for all auxiliary optimisation problems, and the results of Chapters 4 and 5 are obtained by using a newly developed method named Random Heavy Tuning for all auxiliary optimisation problems. Both of these techniques are detailed here. Note that for all given pseudocodes, it is assumed the optimisation consists of the minimisation of a function f with domain $\Omega = [0, 1]^d$. The implemented code applies appropriate linear scaling when this is not the case.

B.1 Accelerated Random Search

Accelerated Random Search[35] consists of maintaining a set of hypercubes in the feasible region, and at each iteration randomly selecting a point within each of the hypercubes to find better solutions. The hypercubes shrink and grow depending on failure/success in finding better points to conduct a more global/local search. Given a location $\mathbf{x} \in \Omega$ and a radius $r \in (0, 1]$, the space $B(\mathbf{x}, r)$ where a new random point is chosen is defined as

$$B(\mathbf{x}, r) = \{\mathbf{x}' \quad | \quad \mathbf{x}' \in \Omega, \quad \|\mathbf{x} - \mathbf{x}'\|_\infty \leq r\},$$

where $\|\|_\infty$ denotes the uniform norm. Algortihm 1 gives the full pseudocode for this algorithm assuming a minimisation of f.

Algorithm 1 Accelerated Random Search

Input: A function $f : \Omega = [0, 1]^d \to \mathbb{R}$, a total number of iterations N, a number k of hypercubes, a shrink factor c, and a shrink threshold ϵ.
Output: A point $\mathbf{x}^*$ with the best objective function value found.

```
n ← 1
Select k random points x_1, ..., x_k ∈ Ω
Set r_i ← 1 for i = 1, ..., k
while n ≤ N do
    n ← n + 1
    for i in 1, ..., k do
        Randomly choose x'_i ∈ B(x_i, r_i) ∩ Ω
        if f(x'_i) < f(x_i) then
            x_i ← x'_i
            r_i ← 1
        else
            r_i ← r_i/c
            if r_i < ε then
                r_i ← 1
            end if
        end if
    end for
end while
x* ← argmin_{i=1,...,k}{f(x_i)}
```

As mentioned above, one of the benefits of ARS is the limited amount of tuning parameters. In this work the values $c = 2$ and $\epsilon = 10^{-7}$ are used. Note however that the exponential reduction of the radius means choosing other c and ϵ values will not alter the behaviour of the search significantly. A recommended $k = 10$ hypercubes are used,

leading to $N = 1000$ iterations for a total of $10,000$ evaluations of the chosen function f. Furthermore, this algorithm does not exploit any known structure of the function being optimised. This can often be a benefit if the function being optimised does not have "nice" properties, which can often be the case both for the concentrated log-likelihood and for certain acquisition functions.

B.2 Random Heavy Tuning

The work of Toal et al. [34] compares a variety of Kriging hyperparameter tuning strategies. In particular, the so-called Heavy Tuning strategy performs quite well, in part due to the relatively high number of 10,000 function evaluations, but also due to the technique relying on three separate, more simple optimisation algorithms. Two runs of local optimisation are first performed via the Amoeba [94] algorithm, also known as the Nelder-Mead method. This ensures an initial solution of good quality is found. Next, a Genetic Algorithm (GA) is run to both further explore the space and exploit good solutions, using the best solution found in the previous step as part of the population. Finally, a greedy search is performed using Dynamic Hill Climbing (DHC) [95] starting from the best point found so far. This approach can be briefly summarised as finding a good initial solution, then exploring the space, and then exploiting the best region found. In the work, it is suggested that half of the total evaluations should be used while running the GA algorithm, with the remaining evaluations used in the Dynamic Hill Climbing.

In the implemented code used for this thesis and available on GitHub [85], two small modifications are applied to Heavy Tuning to generate Random Heavy Tuning, which is described in Algorithm 2. Instead of using a GA which is guided by the best solution found when exploring the space, ARS is used instead. Here one of the hypercubes is initially centered on the best solution found by the Amoeba algorithm, whilst the remaining 9 hypercubes are randomly initialised. This ensures that the exploration of the space is mainly independent of the best solution found so far. Furthermore, no tuning parameters such as the number of generations and population size need to be chosen for the GA. The second modification is the fact that 80% of the total function evaluations are spent doing ARS, with the remaining 20% being spent on DHC. This is simply because, in the observed experiments, DHC converges very quickly and thus spending a larger

amount of the evaluations in the search for good regions seems to yield better results. The pseudocode for the Amoeba, ARS and DHC algorithms are given in Algorithms 1, 3 and 4, respectively. The recommended parameters $\alpha = 1, \gamma = 2, \rho = 1/2, \sigma = 1/2$ are used for the Amoeba algorithm, and a tolerance of $\epsilon = 10^{-7}$ is used for the Amoeba, ARS and DHC algorithms.

Algorithm 2 Random Heavy Tuning

Input: A function $f : \Omega = [0, 1]^d \to \mathbb{R}$ of dimension d.
Output: A point $\mathbf{x}^*$ with the best objective function value found.

1: Run the Amoeba algorithm with an initial random set of $d+1$ points with a maximum number of function evaluations $50d$.
2: Get the best point $\mathbf{x}^*$ found by the Amoeba algorithm.
3: Generate a new set of $d + 1$ points randomly chosen in $B(\mathbf{x}^*, 0.1)$.
4: Run the Aomeba algorithm with the newly chosen points with a maximum number of function evaluations $50d$.
5: Get the best point $\mathbf{x}^*$ found by the Amoeba algorithm.
6: Run ARS for 800 iterations with 10 points, using $\mathbf{x}^*$ as one of the starting centres of the hypercubes.
7: Get the best point $\mathbf{x}^*$ found by the ARS algorithm.
8: Run DHC with a maximum number of 2000 function evaluations starting from $\mathbf{x}^*$.
9: If DHC converges before using all the function evaluations, restart it from the point furthest from all evaluated points so far.
10: Return the best point $\mathbf{x}^*$ found by the DHC algorithm.

Algorithm 3 Amoeba algorithm

Input: A function $f : \Omega = [0,1]^d \to \mathbb{R}$ of dimension d, a maximum number of function evaluations N, a reflection coefficient α, an expansion coefficient γ, a contraction coefficient ρ, a shrink coefficient σ, and a tolerance ϵ.

Output: A point $\mathbf{x}^*$ with the best objective function value found.

1: $n \leftarrow 1$
2: Select $d+1$ random points $\mathbf{x}_1, \ldots, \mathbf{x}_{d+1} \in \Omega$
3: Order the points so that $f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \cdots \leq f(\mathbf{x}_{d+1})$
4: **if** $f(\mathbf{x}_{d+1}) - f(\mathbf{x}_1) < \epsilon$ or $n > N$ **then**
5: $\quad \mathbf{x}^* \leftarrow \mathbf{x}_1$
6: $\quad$ Return $\mathbf{x}^*$
7: **end if**
8: Compute the centroid $\mathbf{x}_o$ of all points except $\mathbf{x}_{d+1}$
9: Compute the reflection $\mathbf{x}_r = \mathbf{x}_o + \alpha(\mathbf{x}_o - \mathbf{x}_{d+1})$
10: $n \leftarrow n + 1$
11: **if** $f(\mathbf{x}_1) \leq f(\mathbf{x}_r) < f(\mathbf{x}_n)$ **then**
12: $\quad \mathbf{x}_{d+1} \leftarrow \mathbf{x}_r$
13: $\quad$ Go to Line 3
14: **else if** $f(\mathbf{x}_r) < f(\mathbf{x}_1)$ **then**
15: $\quad$ Compute the expansion $\mathbf{x}_e = \mathbf{x}_o + \gamma(\mathbf{x}_r - \mathbf{x}_o)$
16: $\quad n \leftarrow n + 1$
17: $\quad$ **if** $f(\mathbf{x}_e) < f(\mathbf{x}_r)$ **then**
18: $\quad\quad \mathbf{x}_{d+1} \leftarrow \mathbf{x}_e$
19: $\quad$ **else**
20: $\quad\quad \mathbf{x}_{d+1} \leftarrow \mathbf{x}_r$
21: $\quad$ **end if**
22: $\quad$ Go to Line 3
23: **else if** $f(\mathbf{x}_r) < f(\mathbf{x}_{d+1})$ **then**
24: $\quad$ Compute the outside contraction $\mathbf{x}_c = \mathbf{x}_o + \rho(\mathbf{x}_r - \mathbf{x}_o)$
25: $\quad n \leftarrow n + 1$
26: $\quad$ **if** $f(\mathbf{x}_c) < f(\mathbf{x}_r)$ **then**
27: $\quad\quad \mathbf{x}_{d+1} \leftarrow \mathbf{x}_c$
28: $\quad\quad$ Go to Line 3
29: $\quad$ **end if**
30: **else**
31: $\quad$ Compute the inside contraction $\mathbf{x}_c = \mathbf{x}_o + \rho(\mathbf{x}_{d+1} - \mathbf{x}_o)$
32: $\quad n \leftarrow n + 1$
33: $\quad$ **if** $f(\mathbf{x}_c) < f(\mathbf{x}_{d+1})$ **then**
34: $\quad\quad \mathbf{x}_{d+1} \leftarrow \mathbf{x}_c$
35: $\quad\quad$ Go to Line 3
36: $\quad$ **end if**
37: **end if**
38: $\mathbf{x}_i \leftarrow \mathbf{x}_1 + \sigma(\mathbf{x}_i - \mathbf{x}_1)$ for $i = 2, \ldots, d+1$
39: $n \leftarrow n + d$
40: Go to Line 3

Algorithm 4 Dynamic Hill Climbing

Input: A function $f : \Omega = [0,1]^d \to \mathbb{R}$ of dimension d, a maximum number of function evaluations N, a starting point $\mathbf{x}_1$ and a tolerance ϵ.

Output: A point $\mathbf{x}^*$ with the best objective function value found.

1: $n \leftarrow 1$
2: $k \leftarrow 1$
3: Initialise d vectors $\mathbf{d}_1^+, \ldots, \mathbf{d}_d^+$ where the ith entry of vector $\mathbf{d}_i^+$ is 0.5, and every other entry is 0
4: Initialise d vectors $\mathbf{d}_1^-, \ldots, \mathbf{d}_d^-$ where the ith entry of vector $\mathbf{d}_i^-$ is -0.5, and every other entry is 0
5: Initialise the dynamic vectors $\mathbf{d}^+ \leftarrow \mathbf{0}$ and $\mathbf{d}^- \leftarrow \mathbf{0}$
6: Define the set of candidate directions $\mathcal{D} = \{\mathbf{d}_1^+, \ldots, \mathbf{d}_d^+, \mathbf{d}_1^-, \ldots, \mathbf{d}_d^-, \mathbf{d}^+, \mathbf{d}^-\}$
7: Choose a direction vector $\mathbf{d} = \text{argmax}_{\mathbf{d} \in \mathcal{D}}\{\|\mathbf{d}\|\}$
8: **if** $n > N$ or $\|\mathbf{d}\| < \epsilon$ **then**
9: $\quad \mathbf{x}^* \leftarrow \mathbf{x}_k$
10: $\quad$ Return $\mathbf{x}^*$
11: **end if**
12: $n \leftarrow n + 1$
13: $\mathbf{x}' \leftarrow \mathbf{x}_k + \mathbf{d}$
14: **if** $\mathbf{x}' \notin [0,1]^d$ or $f(\mathbf{x}_k) \leq f(\mathbf{x}')$ **then**
15: $\quad \mathbf{d} \leftarrow \mathbf{d}/2$
16: $\quad$ Go to line 7
17: **end if**
18: $k \leftarrow k + 1$
19: $\mathbf{x}_k \leftarrow \mathbf{x}'$
20: **if** the same direction led to an improvement twice in a row **then**
21: $\quad \mathbf{d} \leftarrow 2\mathbf{d}$
22: **end if**
23: **if** $k \geq 3$ **then**
24: $\quad \mathbf{d}^+ \leftarrow \mathbf{x}_k - \mathbf{x}_{k-2}$
25: $\quad \mathbf{d}^- \leftarrow -\mathbf{d}^+$
26: **end if**
27: Go to line 7

Appendix C

Implemented literature instances

This appendix provides the definitions of the implemented literature bi-fidelity instances provided in the GitHub repository Bi-fidelity Surrogate Modelling benchmark problems [92]. Note that depending on the chapter, an instance definition might include the number of high- and low-fidelity samples available, or a total sampling budget with a cost ratio. Here, the term instance denotes only a pair of functions (f_l, f_h) as well as their domain Ω, where f_h is labelled the high-fidelity source and f_l the low-fidelity source. As defined in Chapter 3, these instances are either fixed, parameter-based or error-based. The implemented literature instances of these types are defined in Sections C.1, C.2 and C.3, respectively. The same chapter defines a new instance generation procedure to create disturbance-based instances. These are generated by adding a disturbance to a chosen high-fidelity function. In Chapter 3 the high-fidelity instances are taken from the COCO test suite [75]; in Chapters 4 and 5 this procedure is also applied to high-fidelity functions from the implemented bi-fidelity instances. The details of the chosen high-fidelity sources for this new type of instances are given in Section C.4.

C.1 Fixed instances

The term fixed denotes instances generated by choosing a high-fidelity function, and applying a modification to generate a low-fidelity source. A total of 44 instances of this type from 10 different studies [54–56, 58, 60, 64, 70–73] have been implemented.

Five instances from the work of March and Willcox [55] have been implemented, which consist of the Rosenbrock function as the high-fidelity source, and five different options for the low fidelity source

$$
\begin{aligned}
&f_h, f_l : [-5, 10]^2 \to \mathbb{R}, \\
&f_h(\mathbf{x}) = (x_2 - x_1^2)^2 + (1 - x_2)^2, \\
&f_l^1(\mathbf{x}) = 0, \\
&f_l^2(\mathbf{x}) = x_1^2 + x_2^2, \\
&f_l^3(\mathbf{x}) = x_1^4 + x_2^2, \\
&f_l^4(\mathbf{x}) = f_h(\mathbf{x}), \\
&f_l^5(\mathbf{x}) = -x_1^2 - x_2^2.
\end{aligned}
\tag{C.1}
$$

Note that the low-fidelity sources do not constitute a single instance, but rather the pairs (f_l^1, f_h), (f_l^2, f_h), (f_l^3, f_h), (f_l^4, f_h) and (f_l^5, f_h) are five distinct bi-fidelity instances.

Three instances from the work of Rajnarayan et al. [54] have been implemented, which include Hartmann's 3-dimensional function

$$
\begin{aligned}
&f_h, f_l : [0, 1]^3 \to \mathbb{R}, \\
&f_h(\mathbf{x}) = -\sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{3} \beta_{i,j}(x_j - P_{i,j})^2\right], \\
&f_l(\mathbf{x}) = f_h(\mathbf{x}) + 0.5 \sin\left(10(x_1 + x_2 + x_3 - 12)\right), \\
&\text{where } \alpha = \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}, \beta = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}, P = \begin{bmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{bmatrix} \times 10^{-4},
\end{aligned}
\tag{C.2}
$$

Hartmann's 6-dimensional function

$$
\begin{aligned}
f_h, f_l &: [0,1]^6 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= -\sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{6} \beta_{i,j}(x_j - P_{i,j})^2\right], \\
f_l(\mathbf{x}) &= f_h(\mathbf{x}) + 0.5 \sin\left(25(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 - 24)\right),
\end{aligned}
$$

$$
\text{where } \alpha = \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}, \beta = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}, \tag{C.3}
$$

$$
P = \begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix} \times 10^{-4},
$$

and the 4-dimensional Wood's problem

$$
\begin{aligned}
f_h, f_l &: [-1,1]^4 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= 100(x_2 - x_1)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2, \\
&\quad + 10.1((1 - x_2)^2 + (1 - x_4)^2) + 19.8(1 - x_2)(1 - x_4), \\
f_l(\mathbf{x}) &= f_h(\mathbf{x}) + 10\left((x_1 - 0.5)^2 + (x_2 - 0.5)^2 + (x_3 - 0.5)^2 + (x_4 - 0.5)^2\right).
\end{aligned} \tag{C.4}
$$

Five instances from the work of Liu et al. [56] have been implemented, which include the Ellipsoid function

$$
\begin{aligned}
f_h, f_l &: [-30, 30]^{20} \to \mathbb{R}, \\
f_h(\mathbf{x}) &= \sum_{i=1}^{20} i s_i^h (x_i - s_i^s)^2, \\
f_l(\mathbf{x}) &= \sum_{i=1}^{20} i x_i^2, \\
\text{where } s^h &= [0.3, 0.4, 0.2, 0.6, 1, 0.9, 0.2, 0.8, 0.5, 0.7, \\
&\quad 0.4, 0.3, 0.7, 1, 0.9, 0.6, 0.2, 0.8, 0.2, 0.5], \\
s^s &= [1.8, 0.4, 2, 1.2, 1.4, 0.6, 1.6, 0.2, 0.8, 1, \\
&\quad 1.3, 1.1, 2, 1.4, 0.5, 0.3, 1.6, 0.7, 0.3, 1.9],
\end{aligned}
\tag{C.5}
$$

the Dixon and Price problem

$$
\begin{aligned}
f_h, f_l &: [-30, 30]^{20} \to \mathbb{R}, \\
f_h(\mathbf{x}) &= (x_1 - s_1^s - 1)^2 + \sum_{i=2}^{20} i(2(x_i - s_i^s)^2 - (x_{i-1} - s_{i-1}^s))^2, \\
f_l(\mathbf{x}) &= (x_1 - 1)^2 + \sum_{i=2}^{20} i(2x_i^2 - x_{i-1})^2, \\
\text{where } s^s &= [1.8, 0.5, 2, 1.2, 0.4, 0.2, 1.4, 0.3, 1.6, 0.6, \\
&\quad 0.8, 1, 1.3, 1.9, 0.7, 1.6, 0.3, 1.1, 2, 1.4],
\end{aligned}
\tag{C.6}
$$

the Styblinski-Tang problem

$$
\begin{aligned}
f_h, f_l &: [-5, 5]^5 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= 0.5 \sum_{i=1}^{5} (x_i - s_i^s)^4 - 16(x_i - s_i^s)^2 + 5(x_i - s_i^s), \\
f_l(\mathbf{x}) &= 0.5 \sum_{i=1}^{5} x_i^4 - 16x_i^2 + 5x_i, \\
\text{where } s^s &= [0.28, 0.59, 0.47, 0.16, 0.32],
\end{aligned}
\tag{C.7}
$$

and the Ackley problem with $d = 10, 20$

$$
\begin{aligned}
f_h, f_l &: [-30, 30]^d \to \mathbb{R}, \\
f_h(\mathbf{x}) &= -20 \exp\left\{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}(x_i - s_i^s)^2}\right\} \\
&\quad - \exp\left\{\frac{1}{d}\sum_{i=1}^{d}\cos(2.6\pi(x_i - s_i^s))\right\} + 20 + e, \\
f_l(\mathbf{x}) &= -20 \exp\left\{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right\} \\
&\quad - \exp\left\{\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right\} + 20 + e,
\end{aligned}
\tag{C.8}
$$

$$
\begin{aligned}
\text{where for } d = 10, s^s &= [1.3, 0.1, 1.4, 0.8, 1.7, 1, 1.5, 0.6, 2, 0.4], \\
\text{for } d = 20, s^s &= [1.2, 0.2, 1.4, 0.8, 1.8, 1, 1.6, 0.6, 2, 0.4, \\
&\quad 1.3, 0.3, 1.5, 0.9, 1.9, 1.1, 1.7, 0.7, 2.1, 0.5].
\end{aligned}
$$

Two instances from the work of Liu et al. [58] have been implemented, including a variation of Forrester's one dimensional example [38]

$$
\begin{aligned}
f_h, f_l &: [0, 1] \to \mathbb{R}, \\
f_h(x) &= 5x^2 \sin(12x), \\
f_l(x) &= f_h(x) + (x^3 - 0.5)\sin(3x - 0.5) + 4\cos(2x),
\end{aligned}
\tag{C.9}
$$

and the Branin function

$$
\begin{aligned}
f_h, f_l &: [-5, 10] \times [0, 15] \to \mathbb{R}, \\
f_h(\mathbf{x}) &= \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10, \\
f_l(\mathbf{x}) &= 10\sqrt{f_h((x_1 - 2, x_2 - 2))} + 2(x_1 - 2.5) - 3(3x_2 - 7) - 1.
\end{aligned}
\tag{C.10}
$$

The Himmelblau function from the work of Wu et al. [60] has been implemented

$$\begin{aligned} f_h, f_l &: [-3,3]^2 \to \mathbb{R}, \\ f_h(\mathbf{x}) &= (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2, \\ f_l(\mathbf{x}) &= ((0.5x_1)^2 + 0.8x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + x_2^3 - (x_1 + 1)^2. \end{aligned} \tag{C.11}$$

A further 20 instances from the work of Shi et al. [64] are also implemented, including the Gramacy Lee function

$$\begin{aligned} f_h, f_l &: [0,1] \to \mathbb{R}, \\ f_h(x) &= \frac{\sin(10\pi x)}{2x} + (x-1)^4, \\ f_l(x) &= \frac{\sin(10\pi x)}{x} + 2(x-1)^4, \end{aligned} \tag{C.12}$$

the Currin Sin function

$$\begin{aligned} f_h, f_l &: [0,1] \to \mathbb{R}, \\ f_h(x) &= \sin(2\pi(x-0.1)) + x^2, \\ f_l(x) &= \sin(2\pi(x-0.1)), \end{aligned} \tag{C.13}$$

the Holsclaw function

$$\begin{aligned} f_h, f_l &: [0,10] \to \mathbb{R}, \\ f_h(\mathbf{x}) &= \frac{x\sin(x)}{10}, \\ f_l(\mathbf{x}) &= \frac{x\sin(x) + x}{10}, \end{aligned} \tag{C.14}$$

the Santner function

$$\begin{aligned} f_h, f_l &: [0,1] \to \mathbb{R}, \\ f_h(x) &= e^{-1.4x}\cos(3.5\pi x), \\ f_l(x) &= e^{-1.4x}\cos(3.5\pi x) + 0.75x^2, \end{aligned} \tag{C.15}$$

the Branin function

$$\begin{aligned} f_h, f_l &: [0,5]^2 \to \mathbb{R}, \\ f_h(\mathbf{x}) &= \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10, \\ f_l(\mathbf{x}) &= (1 - 0.124\pi)\cos(x), \end{aligned} \tag{C.16}$$

two newly defined functions

$$\begin{aligned} f_h, f_l &: [-1,1]^2 \to \mathbb{R}, \\ f_h(\mathbf{x}) &= 101x_1^2 + 101(x_1^2 + x_2^2)^2, \\ f_l(\mathbf{x}) &= x_1^2 + 100(x_1^2 + x_2^2)^4, \end{aligned} \tag{C.17}$$

and

$$\begin{aligned} f_h, f_l &: [-5,10]^2 \to \mathbb{R}, \\ f_h(\mathbf{x}) &= (1 - 0.2x_2 + 0.05\sin(4\pi x_2 - x_1))^2 + (x_2 - 0.5\sin(2\pi x_1))^2, \\ f_l(\mathbf{x}) &= (1 - 0.2x_2 + 0.05\sin(4\pi x_2 - x_1))^2 + 4(x_2 - 0.5\sin(2\pi x_1))^2, \end{aligned} \tag{C.18}$$

the Beale function

$$\begin{aligned} f_h, f_l &: [-4.5,4.5]^2 \to \mathbb{R}, \\ f_h(\mathbf{x}) &= (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2, \\ f_l(\mathbf{x}) &= (1.5 - x_1 + x_1x_2)^2 + x_1 + x_2, \end{aligned} \tag{C.19}$$

the Styblinski Tang function

$$\begin{aligned} f_h, f_l &: [-4,4]^2 \to \mathbb{R}, \\ f_h(\mathbf{x}) &= x_1^4 - 16x_1^2 + 5x_1 + x_2^4 - 16x_2^2 + 5x_2, \\ f_l(\mathbf{x}) &= x_1^4 - 16x_1^2 + x_2^4 - 16x_2^2, \end{aligned} \tag{C.20}$$

the Currin exponential function

$$
\begin{aligned}
f_h, f_l &: [0, 0.5]^2 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= \frac{(1 - e^{-\frac{1}{2x_2}})(2300x_1^3 + 1900x_1^2 + 2902x_1 + 60)}{100x_1^3 + 500x_1^2 + 4x_1 + 20} \\
f_l(\mathbf{x}) &= -0.4 f_h((x_1 + 0.05, x_2 + 0.05)) \\
&\quad + 0.25 f_h((x_1 + 0.05, \max\{0, x_2 - 0.05\})) \\
&\quad + 0.25 f_h((x_1 - 0.05, x_2 + 0.05)) \\
&\quad + 0.25 f_h((x_1 - 0.05, \max\{0, x_2 - 0.05\})),
\end{aligned} \tag{C.21}
$$

the Lim function

$$
\begin{aligned}
f_h, f_l &: [0, 10]^2 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= \frac{1}{6}\left[(30 + 5x_1 \sin(5x_1))(4 + e^{-5x_2}) - 100\right], \\
f_l(\mathbf{x}) &= \frac{1}{6}\left[(30 + 5x_1 \sin(5x_1))(4 + \frac{2}{5}e^{-5x_2}) - 100\right],
\end{aligned} \tag{C.22}
$$

the Gramacy function

$$
\begin{aligned}
f_h, f_l &: [-2, 2]^2 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= x_1 e^{-x_1^2 - x_2^2}, \\
f_l(\mathbf{x}) &= x_1 e^{-x_1^2 - x_2^2} + \frac{x_1}{10},
\end{aligned} \tag{C.23}
$$

the Cheng Sandu function

$$
\begin{aligned}
f_h, f_l &: [-1, 1]^2 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= \cos(x_1 + x_2) e^{x_1 x_2}, \\
f_l(\mathbf{x}) &= \cos(x_1 + x_2) e^{x_1 x_2} + \cos(x_1^2 + x_2^2),
\end{aligned} \tag{C.24}
$$

Hartmann's 3-dimensional function

$$
\begin{aligned}
f_h, f_l &: [0,1]^3 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= -\sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{3} \beta_{i,j}(x_j - P_{i,j})^2\right], \\
f_l(\mathbf{x}) &= -\sum_{i=1}^{4} \exp\left[-\sum_{j=1}^{3} \beta_{i,j}(x_j - P_{i,j})^2\right], \\
\text{where } \alpha &= \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}, \beta = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}, P = \begin{bmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{bmatrix} \times 10^{-4},
\end{aligned}
\tag{C.25}
$$

Dette and Pepelyshev's exponential function

$$
\begin{aligned}
f_h, f_l &: [0,1]^3 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= 100e^{\frac{-2}{x_1^{1.75}}} + e^{\frac{-2}{x_2^{1.5}}} + e^{\frac{-2}{x_3^{1.25}}}, \\
f_l(\mathbf{x}) &= 100e^{\frac{-2}{x_1^{1.75}}} + e^{\frac{-2}{x_2^{1.5}}} + 0.2e^{\frac{-2}{x_3^{1.25}}},
\end{aligned}
\tag{C.26}
$$

Hartmann's 4-dimensional function

$$
\begin{aligned}
f_h, f_l &: [0,1]^4 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= -\sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{4} \beta_{i,j}(x_j - P_{i,j})^2\right], \\
f_l(\mathbf{x}) &= -\sum_{i=1}^{4} \exp\left[-\sum_{j=1}^{4} \beta_{i,j}(x_j - P_{i,j})^2\right],
\end{aligned}
$$

$$
\text{where } \alpha = \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}, \beta = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}, \tag{C.27}
$$

$$
P = \begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix} \times 10^{-4},
$$

the Park function

$$
\begin{aligned}
f_h, f_l &: [0,1]^4 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= \frac{x_1}{2}\left(\sqrt{1 + \frac{(x_1 + x_3^3)x_4}{x_1^2}} - 1\right) + (x_1 + 3x_4)e^{1+\sin(x_3)}, \\
f_l(\mathbf{x}) &= 0.79(1 + \frac{\sin(x_1)}{10})f_h(\mathbf{x}) - 2x_1 + x_2^2 + x_3^2 + 0.5,
\end{aligned} \tag{C.28}
$$

Hartmann's 6-dimensional function

$$
\begin{aligned}
f_h, f_l &: [0,1]^6 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= -\sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{6} \beta_{i,j}(x_j - P_{i,j})^2\right], \\
f_l(\mathbf{x}) &= -\sum_{i=1}^{4} \exp\left[-\sum_{j=1}^{6} \beta_{i,j}(x_j - P_{i,j})^2\right],
\end{aligned}
$$

$$
\text{where } \alpha = \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}, \beta = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}, \tag{C.29}
$$

$$
P = \begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix} \times 10^{-4},
$$

the Rosenbrock function

$$
\begin{aligned}
f_h, f_l &: [-5,10]^6 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= \sum_{i=1}^{5} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2, \\
f_l(\mathbf{x}) &= \sum_{i=1}^{5} 100(x_{i+1} - x_i)^2 + 4(x_i - 1)^2,
\end{aligned} \tag{C.30}
$$

and finally Dette and Pepelyshev's function

$$
\begin{aligned}
f_h, f_l &: [0,1]^8 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= 4(x_1 - 2 + 8x_2 - 8x_2^2)^2 + (3 - 4x_2)^2 + 16\sqrt{x_3 + 1}(2x_3 - 1)^2 \\
&\quad + \sum_{i=4}^{8} i \log\left(1 + \sum_{j=3}^{i} x_j\right), \\
f_l(\mathbf{x}) &= 4(x_1 - 2 + 8x_2 - 8x_2^2)^2 + (3 - 4x_2)^2 + 16\sqrt{x_3 + 1}(2x_3 - 1)^2 \\
&\quad + \sum_{i=4}^{8} \log\left(1 + \sum_{j=3}^{i} x_j\right).
\end{aligned} \tag{C.31}
$$

Five instances from the work of Dong et al. [70] have been implemented, including the Bohachevsky function

$$
\begin{aligned}
f_h, f_l &: [-100, 100]^2 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7, \\
f_l(\mathbf{x}) &= f_h((0.7x_1, x_2)) + x_1 x_2 - 12,
\end{aligned} \tag{C.32}
$$

the Booth function

$$
\begin{aligned}
f_h, f_l &: [-10, 10]^2 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2, \\
f_l(\mathbf{x}) &= f_h((0.4x_1, x_2)) + 1.7x_1 x_2 - x_1 + 2x_1,
\end{aligned} \tag{C.33}
$$

the Branin function

$$
\begin{aligned}
f_h, f_l &: [-5, 10] \times [0, 15] \to \mathbb{R}, \\
f_h(\mathbf{x}) &= \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10 - 22.5x_2, \\
f_l(\mathbf{x}) &= f_h((0.7x_1, 0.7x_2)) + 20(0.9 + x_1)^2 - 50,
\end{aligned} \tag{C.34}
$$

the Himmelblau function

$$\begin{aligned} f_h, f_l &: [-3,3]^2 \to \mathbb{R}, \\ f_h(\mathbf{x}) &= (x_1^2 + x_2 - 11)^2 + (x_2^2 + x_1 - 7)^2, \\ f_l(\mathbf{x}) &= f_h((0.5x_1, 0.8x_2)) + x_2^3 - (x_1+1)^2, \end{aligned} \tag{C.35}$$

and the six hump camelback function

$$\begin{aligned} f_h, f_l &: [-2,2]^2 \to \mathbb{R}, \\ f_h(\mathbf{x}) &= 4x_1^2 - 2.1x_1^4 + \frac{x_1^6}{3} + x_1x_2 - 4x_2^2 + 4x_2^4, \\ f_l(\mathbf{x}) &= f_h((0.7x_1, 0.7x_2)) + x_1x_2 - 15. \end{aligned} \tag{C.36}$$

Two instances from the work of Xiong et al. [71] have been implemented, namely the first Park function

$$\begin{aligned} f_h, f_l &: [0,1]^4 \to \mathbb{R}, \\ f_h(\mathbf{x}) &= \frac{x_1}{2}(\sqrt{1 + (x_1 + x_3^3)\frac{x_4}{x_1^2} - 1} + (x_1 + 3x_4)e^{1+\sin(x_3)}, \\ f_l(\mathbf{x}) &= \frac{1+\sin(x_1)}{10} f_h(\mathbf{x}) - 2x_1 + x_2^2 + x_3^2 + 0.5, \end{aligned} \tag{C.37}$$

and the second Park function

$$\begin{aligned} f_h, f_l &: [0,1]^4 \to \mathbb{R}, \\ f_h(\mathbf{x}) &= \frac{2}{3}e^{x_1+x_2} - x_4\sin(x_3) + x_3, \\ f_l(\mathbf{x}) &= 1.2 f_h(\mathbf{x}) - 1. \end{aligned} \tag{C.38}$$

Finally, one instance from the work of Park et al. [73] is implemented, namely Hartmann's 6-dimensional function

$$
\begin{aligned}
&f_h, f_l : [0,1]^6 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= -\frac{1}{1.94}\left(2.58 + \sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{6} \beta_{i,j}(x_j - P_{i,j})^2\right]\right), \\
f_l(\mathbf{x}) &= -\frac{1}{1.94}\left(2.58 + \sum_{i=1}^{4} \alpha_i' \left[e^{-\frac{4}{9}} + \frac{1}{9}e^{-\frac{4}{9}}\left\{4 - \sum_{j=1}^{6} \beta_{i,j}(x_j - P_{i,j})^2\right\}\right]^9\right), \\
\text{where } \alpha &= \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}, \alpha' = \begin{bmatrix} 0.5 \\ 0.5 \\ 2 \\ 4 \end{bmatrix}, \beta = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}, \\
P &= \begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix} \times 10^{-4}.
\end{aligned}
\tag{C.39}
$$

C.2 Parameter-based instances

The term parameter-based denotes instances of the type presented by Toal [18], where the high-fidelity function is fixed, and the low-fidelity function contains a single parameter $A \in [0,1]$. This parameter can be changed to create function pairs with low-fidelity functions of varying quality. A total of 7 such "families" of instances are implemented from two studies.

Four instance families from the work of Toal [18] have been implemented, which include the Branin function

$$
\begin{aligned}
f_h, f_l &: [-5, 10] \times [0, 15] \to \mathbb{R}, \\
f_h(\mathbf{x}) &= \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10, \\
f_l(\mathbf{x}) &= f_h(\mathbf{x}) - (A + 0.5)\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2,
\end{aligned}
\tag{C.40}
$$

the Paciorek function

$$
\begin{aligned}
f_h, f_l &: [0.3, 1]^2 \to \mathbb{R}, \\
f_h(\mathbf{x}) &= \sin\left(\frac{1}{x_1 x_2}\right), \\
f_l(\mathbf{x}) &= f_h(\mathbf{x}) - 9A^2\cos\left(\frac{1}{x_1 x_2}\right),
\end{aligned}
\tag{C.41}
$$

Hartmann's 3-dimensional function

$$
\begin{aligned}
f_h, f_l &: [0, 1] \to \mathbb{R}, \\
f_h(\mathbf{x}) &= -\sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{3} \beta_{i,j}(x_j - P_{i,j})^2\right], \\
f_l(\mathbf{x}) &= -\sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{3} \beta_{i,j}(x_j - \frac{3}{4}P_{i,j}(A+1))^2\right], \\
\text{where } \alpha &= \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}, \beta = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}, P = \begin{bmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{bmatrix} \times 10^{-4},
\end{aligned}
\tag{C.42}
$$

and the 10-dimensional Trid function

$$
\begin{aligned}
f_h, f_l &: [-100, 100]^{10} \to \mathbb{R}, \\
f_h(\mathbf{x}) &= \sum_{i=1}^{10}(x_i - 1)^2 - \sum_{i=2}^{10} x_i x_{i-1}, \\
f_l(\mathbf{x}) &= \sum_{i=1}^{10}(x_i - A)^2 - (A - 0.65)\sum_{i=2}^{10} i x_i x_{i-1}.
\end{aligned}
\tag{C.43}
$$

Three instance families from the work of Song et al. [40] have also been implemented, including a variation of Forrester's one dimensional example [38]

$$\begin{aligned} f_h, f_l &: [0,1] \to \mathbb{R}, \\ f_h(x) &= (6x-2)^2 \sin(12x-4), \\ f_l(x) &= (1 - A^2 - 2A) \cdot f_h(x) + 10(x - 0.5) - 5, \end{aligned} \tag{C.44}$$

the Branin function

$$\begin{aligned} f_h, f_l &: [-5,10] \times [0,15] \to \mathbb{R}, \\ f_h(\mathbf{x}) &= \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10, \\ f_l(\mathbf{x}) &= f_h(\mathbf{x}) - (0.5A^2 + A + 0.2)\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2, \end{aligned} \tag{C.45}$$

and the Colville function

$$\begin{aligned} f_h, f_l &: [-10,10]^4 \to \mathbb{R}, \\ f_h(\mathbf{x}) &= 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2, \\ &\quad + 10.1\left((x_2 - 1)^2 + (x_4 - 1)^2\right) + 19.8(x_2 - 1)(x_4 - 1), \\ f_l(\mathbf{x}) &= f_h(\mathbf{x}) - (A + 0.5)(5x_1^2 + 4x_2^2 + 3x_3^2 + x_4^2). \end{aligned} \tag{C.46}$$

C.3 Error-based instances

The term error-based denotes instances which are created by adding an error to a chosen high-fidelity function as proposed by Wang et al. [20]. As discussed in Chapter 3, only the so-called "resolution error" instances are implemented here, using the Rastrigin function with $d = 5, 10$ as f_h. Namely, these instances are defined as

$$\begin{aligned} f_h, f_l &: [-1,1]^d \to \mathbb{R}, \\ f_h(\mathbf{x}) &= \sum_{i=1}^{d} x_i^2 + 1 - \cos(10\pi x_i), \\ f_l(\mathbf{x}) &= f_h(\mathbf{x}) + e_r(\mathbf{x}, \phi), \\ \text{where } e_r(\mathbf{x}, \phi) &= \sum_{i=1}^{d} a(x_i, \phi)\cos(w(\phi)x_i + b(\phi) + \pi). \end{aligned} \tag{C.47}$$

Here $e_r(\mathbf{x}, \phi)$ contains is a composition of the functions a, b and w which are defined for four errors e_r^1, e_r^2, e_r^3 and e_r^4

$$\begin{aligned} e_r^1(\mathbf{x}, \phi) &= \sum_{i=1}^{d} a^1(\phi) \cos(w^1(\phi) x_i + b^1(\phi) + \pi), \\ \text{where } a^1(\phi) &= \theta^1(\phi), \quad w^1(\theta) = 10\pi\theta^1(\phi), \\ b^1(\phi) &= 0.5\pi\theta^1(\phi), \quad \theta^1(\phi) = 1 - 0.0001\phi, \end{aligned} \tag{C.48}$$

$$\begin{aligned} e_r^2(\mathbf{x}, \phi) &= \sum_{i=1}^{d} a^2(\phi) \cos(w^2(\phi) x_i + b^2(\phi) + \pi), \\ \text{where } a^2(\phi) &= \theta^2(\phi), \quad w^2(\theta) = 10\pi\theta^2(\phi), \\ b^2(\phi) &= 0.5\pi\theta^2(\phi), \quad \theta^2(\phi) = e^{-0.00025\phi}, \end{aligned} \tag{C.49}$$

$$\begin{aligned} e_r^3(\mathbf{x}, \phi) &= \sum_{i=1}^{d} a^3(\phi) \cos(w^3(\phi) x_i + b^3(\phi) + \pi), \\ \text{where } a^3(\phi) &= \theta^3(\phi), \quad w^3(\theta) = 10\pi\theta^3(\phi), \quad b^3(\phi) = 0.5\pi\theta^3(\phi), \\ \theta^3(\phi) &= \begin{cases} 1 - 0.0002\phi & 0 \leq \phi < 1000, \\ 0.8 & 1000 \leq \phi < 2000, \\ 1.2 - 0.0002\phi & 2000 \leq \phi < 3000, \\ 0.6 & 3000 \leq \phi < 4000, \\ 1.4 - 0.0002\phi & 4000 \leq \phi < 5000, \\ 0.4 & 5000 \leq \phi < 6000, \\ 1.6 - 0.0002\phi & 6000 \leq \phi < 7000, \\ 0.2 & 7000 \leq \phi < 8000, \\ 1.8 - 0.0002\phi & 8000 \leq \phi < 9000, \\ 0 & 9000 \leq \phi < 10000, \end{cases} \end{aligned} \tag{C.50}$$

$$
\begin{aligned}
e_r^4(\mathbf{x}, \phi) &= \sum_{i=1}^{d} a^4(x_i, \phi) \cos(w^1(\phi) x_i + b^4(\phi) + \pi), \\
\text{where } a^4(x_i, \phi) &= \theta^4(\phi)\psi(x_i), \quad w^4(\theta) = 10\pi\theta^4(\phi), \\
\theta^4(\phi) &= 1 - 0.0001\phi, \quad \psi(x_i) = 1 - |x_i|.
\end{aligned}
\tag{C.51}
$$

C.4 Disturbance-based instances

The generating procedure for disturbance-based instances is given in Chapter 3. In the same chapter the chosen high-fidelity functions are taken from the COCO test suite [75], which contains 24 generating functions for which the dimension can be specified. The domain is taken to be $[-5, 5]^d$. First define the following terms

$\mathbf{x}^*$: The location of the optimum. Unless otherwise specified, this is randomly chosen inside the domain.

f^* : The optimum value. If not specified, this is randomly chosen between -100 and 100.

$\mathbf{1}_-^+$: A d-dimensional vector with entries -1 or 1 randomly chosen.

Λ^α : Diagonal matrix where the ith entry is $\alpha^{\frac{1}{2}\frac{i-1}{d-1}}$.

$\mathbf{Q}, \mathbf{R}$: Random orthogonal rotation matrices.

T_{asy}^β : Maps each entry x_i of a point $\mathbf{x}$ to $x_i^{1+\beta\frac{i-1}{d-1}\sqrt{x_i}}$ if $x_i > 0$, and x_i otherwise.

T_{osz} : Maps each entry x_i of a point $\mathbf{x}$ to $\text{sign}(x_i)\exp(\hat{x}_i + 0.049(\sin(c_1\hat{x}_i) + sin(c_2\hat{x}_i))$,

$$
\text{where } \hat{x}_i = \begin{cases} \log(|x_i|) & \text{if } x_i \neq 0 \\ 0 & \text{otherwise} \end{cases}, \quad \text{sign}(x_i) = \begin{cases} -1 & \text{if } x_i < 0 \\ 0 & \text{if } x_i = 0 \\ 1 & \text{if } x_i > 0 \end{cases},
$$

$$
c_1 = \begin{cases} 10 & \text{if } x_i > 0 \\ 5.5 & \text{otherwise} \end{cases}, \quad c_2 = \begin{cases} 7.9 & \text{if } x_i > 0 \\ 3.1 & \text{otherwise} \end{cases}.
$$

The COCO functions are then defined as follows

$$f_1(\mathbf{x}) = \|\mathbf{z}\|^2 + f^*,$$

where $\mathbf{z} = \mathbf{x} - \mathbf{x}^*$,

$$f_2(\mathbf{x}) = f^* + \sum_{i=1}^{d} 10^{6\frac{i-1}{d-1}} z_i^2,$$

where $\mathbf{z} = T_{osz}(\mathbf{x} - \mathbf{x}^*)$,

$$f_3(\mathbf{x}) = f^* + \|\mathbf{z}\|^2 + 10\left(d - \sum_{i=1}^{d} \cos(2\pi z_i)\right),$$

where $\mathbf{z} = \Lambda^{10} T_{asy}^{0.2}(T_{osz}(\mathbf{x} - \mathbf{x}^*))$,

$$f_4(\mathbf{x}) = 10\left(d - \sum_{i=1}^{d}\cos(2\pi z_i)\right) + \sum_{i=1}^{d} z_i^2 + f^*,$$

where $z_i = s_i T_{osz}(x_i - x_i^*)$,

$$s_i = \begin{cases} 10 \times 10^{\frac{1}{2}\frac{i-1}{d-1}} \text{ if } z_i > 0 \text{ and i = 1,3,5,...} \\ 10^{\frac{1}{2}\frac{i-1}{d-1}} \qquad \text{otherwise} \end{cases},$$

$$f_5(\mathbf{x}) = \sum_{i=1}^{d} 5|s_i| - s_i z_i + f^*,$$

where $z_i = x_i$ if $x_i^* x_i < 5^2$ and $z_i = x_i^*$ otherwise,

$$s_i = \text{sign}(x_i^*)10^{\frac{i-1}{d-1}},$$

$$\mathbf{x}^* = 5 \times \mathbf{1}_-^+,$$

$$f_6(\mathbf{x}) = T_{osz}\left(\sum_{i=1}^{d}(s_i z_i)^2\right)^{0.9} + f^*,$$

where $\mathbf{z} = \mathbf{Q}\Lambda^{10}\mathbf{R}(\mathbf{x} - \mathbf{x}^*)$,

$$s_i = \begin{cases} 10^2 \text{ if } z_i x_i^* > 0 \\ 1 \text{ otherwise} \end{cases},$$

$$f_7(\mathbf{x}) = 0.1\max\left(\frac{|\hat{z}_1|}{10^4}, \sum_{i=1}^{d} 10^{2\frac{i-1}{d-1}} z_i^2\right) + f^*,$$

where $\hat{\mathbf{z}} = \Lambda^{10}\mathbf{R}(\mathbf{x} - \mathbf{x}^*)$,

$$\bar{z}_i = \begin{cases} \lfloor 0.5 + \hat{z}_i \rfloor & \text{if } |\hat{z}_i| > 0.5 \\ \lfloor 0.5 + 10\hat{z}_i \rfloor / 10 & \text{otherwise} \end{cases},$$

$$\mathbf{z} = \mathbf{Q}\bar{\mathbf{z}},$$

$$f_8(\mathbf{x}) = \sum_{i=1}^{d-1}\left(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2\right) + f^*,$$

where $\mathbf{z} = \max\left(1, \frac{\sqrt{d}}{8}\right)(\mathbf{x} - \mathbf{x}^*) + \mathbf{1}$,

Here $\mathbf{x}^* \in [-3, 3]^d$,

$$f_9(\mathbf{x}) = \sum_{i=1}^{d-1} \left(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2\right) + f^*,$$

where $\mathbf{z} = \max\left(1, \frac{\sqrt{d}}{8}\right)\mathbf{R}\mathbf{x} + \mathbf{1}/2,$

$$f_{10}(\mathbf{x}) = \sum_{i=1}^{d} 10^{6\frac{i-1}{d-1}} z_i^2 + f^*,$$

where $\mathbf{z} = T_{osz}(\mathbf{R}(\mathbf{x} - \mathbf{x}^*)),$

$$f_{11}(\mathbf{x}) = 10^6 z_1^2 + \sum_{i=2}^{d} z_i^2 + f^*,$$

where $\mathbf{z} = T_{osz}(\mathbf{R}(\mathbf{x} - \mathbf{x}^*)),$

$$f_{12}(\mathbf{x}) = z_1^2 + 10^6 \sum_{i=2}^{d} z_i^2 + f^*,$$

where $\mathbf{z} = \mathbf{R}T_{asy}^{0.5}(\mathbf{R}(\mathbf{x} - \mathbf{x}^*)),$

$$f_{13}(\mathbf{x}) = z_1^2 + 100\sqrt{\sum_{i=2}^{d} z_i^2} + f^*,$$

where $\mathbf{z} = \mathbf{Q}\Lambda^{10}\mathbf{R}(\mathbf{x} - \mathbf{x}^*),$

$$f_{14}(\mathbf{x}) = \sqrt{\sum_{i=1}^{d} |z_i|^{2+4\frac{i-1}{d-1}}} + f^*,$$

where $\mathbf{z} = \mathbf{R}(\mathbf{x} - \mathbf{x}^*),$

$$f_{15}(\mathbf{x}) = 10\left(d - \sum_{i=1}^{d} \cos(2\pi z_i)\right) + \|\mathbf{z}\|^2 + f^*,$$

where $\mathbf{z} = \mathbf{R}\Lambda^{10}\mathbf{Q}T_{asy}^{0.2}(T_{osz}(\mathbf{R}(\mathbf{x} - \mathbf{x}^*))),$

$$f_{16}(\mathbf{x}) = 10\left(\frac{1}{d}\sum_{i=1}^{d}\sum_{k=0}^{11} \frac{1}{2^k}\cos(2\pi 3^k(z_i + \frac{1}{2})) - f_0\right)^3 + f^*,$$

where $\mathbf{z} = \mathbf{R}\Lambda^{0.01}\mathbf{Q}T_{osz}(\mathbf{R}(\mathbf{x} - \mathbf{x}^*)),$

$$f_0 = \sum_{k=0}^{11} \frac{1}{2^k}\cos(2\pi 3^k(\frac{1}{2})),$$

$$f_{17}(\mathbf{x}) = \left(\frac{1}{d-1}\sum_{i=1}^{d-1}\sqrt{s_i} + \sqrt{s_i}\sin^2\left(50 s_i^{\frac{1}{5}}\right)\right)^2 + f^*,$$

where $\mathbf{z} = \Lambda^{10}\mathbf{Q}T_{asy}^{0.5}(\mathbf{R}(\mathbf{x}-\mathbf{x}^*))$,

$$s_i = \sqrt{z_i^2 + z_{i+1}^2},$$

$$f_{18}(\mathbf{x}) = \left(\frac{1}{d-1}\sum_{i=1}^{d-1}\sqrt{s_i} + \sqrt{s_i}\sin^2\left(50 s_i^{\frac{1}{5}}\right)\right)^2 + f^*,$$

where $\mathbf{z} = \Lambda^{1000}\mathbf{Q}T_{asy}^{0.5}(\mathbf{R}(\mathbf{x}-\mathbf{x}^*))$,

$$s_i = \sqrt{z_i^2 + z_{i+1}^2},$$

$$f_{19}(\mathbf{x}) = \frac{10}{d-1}\sum_{i=1}^{d-1}\left(\frac{s_i}{4000} - \cos(s_i)\right) + 10 + f^*,$$

where $\mathbf{z} = \max\left(1, \frac{\sqrt{d}}{8}\right)\mathbf{R}\mathbf{x} + 0.5$,

$$s_i = 100(z_i^2 - z_{i+1}^2)^2 + (z_i - 1)^2,$$

$$f_{20}(\mathbf{x}) = -\frac{1}{100d}\sum_{i=1}^{d} z_i \sin(\sqrt{|z_i|}) + 4.189828872724339 + f^*,$$

where $\hat{\mathbf{x}} = 2 \times \mathbf{1}_-^+ \otimes \mathbf{x}$,

$$\hat{z}_1 = \hat{x}_1, \quad \hat{z}_{i+1} = \hat{x}_{i+1} + 0.25(\hat{x}_i - 2|x_i^*|),$$

$$\mathbf{z} = 100(\Lambda^{10}(\hat{\mathbf{z}} - 2|\mathbf{x}^*|) + 2|\mathbf{x}^*|),$$

$\mathbf{x}^* = \frac{4.209687463}{2}\mathbf{1}_-^+$, using the same $\mathbf{1}_-^+$ vector as above,

$$f_{21}(\mathbf{x}) = T_{osz}\left(10 - \max_{i=1,\ldots,101}\left(w_i \exp\left\{-\frac{1}{2d}(\mathbf{x}-\mathbf{y}_i)^\top \mathbf{R}^\top \mathbf{C}_i \mathbf{R}(\mathbf{x}-\mathbf{y}_i)\right\}\right)\right)^2 + f^*,$$

where $w_1 = 10, \quad w_i = 1.1 + 8 \times \frac{i-2}{99}$ for $2 \le i \le 101$,

$\mathbf{C}_i = \frac{1}{\alpha_i^{0.25}}\Lambda^{\alpha_i}$ where Λ^{α_i} is defined as usual but with randomly permuted diagonal elements. For $2 \le i \le 101$ α_i is randomly drawn from $\{1000^{2\frac{i}{99}} | j = 0, \ldots, 99\}$ without replacement, and $\alpha_1 = 1000$,

$\mathbf{y}_1 = \mathbf{x}^* \in [-4, 4], \quad \mathbf{y}_i \in [-5, 5]$ for $2 \le i \le 101$ are randomly chosen,

$$f_{22}(\mathbf{x}) = T_{osz}\left(10 - \max_{i=1,\ldots,21}\left(w_i \exp\left\{-\frac{1}{2d}(\mathbf{x}-\mathbf{y}_i)^\top \mathbf{R}^\top \mathbf{C}_i \mathbf{R}(\mathbf{x}-\mathbf{y}_i)\right\}\right)\right)^2 + f^*,$$

where $w_1 = 10, \quad w_i = 1.1 + 8 \times \frac{i-2}{99}$ for $2 \leq i \leq 21$,

$\mathbf{C}_i = \frac{1}{\alpha_i^{0.25}}\Lambda^{\alpha_i}$ where Λ^{α_i} is defined as usual but with randomly permuted diagonal elements. For $2 \leq i \leq 21$ α_i is randomly drawn from $\{1000^{2\frac{i}{19}} | j = 0, \ldots, 19\}$ without replacement, and $\alpha_1 = 1000^2$,

$\mathbf{y}_1 = \mathbf{x}^* \in [-3.92, 3.92], \quad \mathbf{y}_i \in [-4.9, 4.9]$ for $2 \leq i \leq 21$ are randomly chosen,

$$f_{23}(\mathbf{x}) = \frac{10}{d^2}\prod_{i=1}^{d}\left(1 + i\sum_{j=1}^{32}\frac{|2^j z_i - [2^j z_i]|}{2^j}\right)^{\frac{10}{d^{1/2}}} - \frac{10}{d^2} + f^*,$$

where $\mathbf{z} = \mathbf{Q}\Lambda^{100}\mathbf{R}(\mathbf{x} - \mathbf{x}^*)$,

$$f_{24}(\mathbf{x}) = \min\left(\sum_{i=1}^{d}(\hat{x}_i - \mu_0)^2, d + s\sum_{i=1}^{d}(\hat{x}_i - \mu_1)^2\right) + 10\left(d - \sum_{i=1}^{d}\cos(2\pi z_i)\right) + f^*,$$

where $\hat{\mathbf{x}} = 2\text{sign}(\mathbf{x}^*) \otimes \mathbf{x}, \quad \mathbf{x}^* = \frac{\mu_0}{2}\mathbf{1}_-^+$,

$\mathbf{z} = \mathbf{Q}\Lambda^{100}\mathbf{R}(\hat{\mathbf{x}} - \mu_0\mathbf{1})$,

$\mu_0 = 2.5, \quad \mu_1 = -\sqrt{\frac{\mu_0^2 - 1}{s}}, \quad s = 1 - \frac{1}{2\sqrt{d+20} - 8.2}$.

As outlined in Chapter 3, new bi-fidelity instances are generated by taking these functions with a variety of dimensions as the high-fidelity source f_h and adding a disturbance to generate a low-fidelity source. In Chapter 4, the newly created set of instances is supplemented by also using 37 high-fidelity functions implemented from the literature, namely those defined in Equations C.1, C.2, C.4, C.5, C.6, C.7, C.8 with $d = 10, 20$, C.9, C.10, C.12, C.13, C.14, C.15, C.17, C.18, C.19, C.20, C.21, C.22, C.23, C.26, C.27, C.28, C.30, C.31, C.32, C.33, C.35, C.36, C.38, C.39 C.41, C.43, C.46, and C.47 with $d = 5, 10$.

C.5 SOLAR

The SOLAR package [91] provides ten black-box instances for benchmarking black-box methods. Each of the available instances present different design characteristics of a molten salt solar power plant simulated by black-box numerical models. Different design aspects of such a power plant can be investigated, such as placement and dimensions of the heliostats (i.e. the mirrors which reflect sunlight into a receiver tower), the characteristics of the central power which uses light to melt salt, the heat storage characteristics of the plant, the heat transfer system between the molten salt and water to produce steam, and the type of turbine used to generate power. The variety of these aspects produces a collection includes instances which might include integer and categorical input variables, stochasticity and / or variable fidelity.

As stated in Chapter 4, the tenth instance is relevant to the work of this thesis as it is a deterministic function with continuous inputs bounded within a hypercube. This instance has 5 inputs which affect the power storage capability of the power plant, including the heat storage dimension and insulation thickness. When running the code available in GitHub [96], a fidelity $fid \in (0, 1]$ can be specified which impacts the accuracy of the output. In practice this parameter impacts the convergence of the inner simulations of the software; this leads to a situation where lowering the fidelity leads to much faster evaluations of a set of inputs which in turn may decrease the accuracy of the output. Using same notation as the rest of this appendix, the tenth SOLAR instance is defined as

$$f_{fid}^{SOLAR} : [793, 995] \times [2, 50] \times [2, 30] \times [0.01, 5] \times [0.01, 5] \to \mathbb{R},$$
$$f_{fid}^{SOLAR}(\mathbf{x}) = \text{Run SOLAR with input } \mathbf{x} \text{ and fidelity } fid.$$

Note that this constitutes a true black-box function as no analytical expression for the output exists. As described in Chapter 4, setting $f_1^{SOLAR} = f_h$ and $f_{fid}^{SOLAR} = f_l$ for $fid \in \{0.1, 0.2, \ldots, 0.9\}$ gives rise to a set of 9 bi-fidelity fuction pairs.

Appendix D

Flacco features

This Appendix provides the definitions of the features from the R package flacco [86] used in Chapter 4 to characterise harmful data sources. These features characterise the landscape of a black-box function, and are used in the chapter to characterise the high-fidelity source f_h, the low-fidelity source f_l, and the difference $f_h - f_l$, using both large and small datasets. These calculated feature values are provided in Figshare [93]. The details of how these features are used however have no impact on their definition, which is assumed to be calculated for a function $f : \Omega \to \mathbb{R}$ of dimension d using a sample $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \Omega$, $\mathbf{Y} = \{y_1, \ldots, y_n\} = \{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)\}$. The used features are grouped by flacco into feature sets, each of which is defined in the section of the same name.

D.1 ELA distribution features

The Exploratory Landscape Analysis (ELA) distribution features [87] characterise the distribution of the objective function values given in $\mathbf{Y}$. They are defined as

$$\text{Skewness}(\mathbf{Y}) = \left(\frac{1}{sd(\mathbf{Y})}\right)^3 \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^3,$$

$$\text{Kurtosis}(\mathbf{Y}) = \left(\frac{1}{sd(\mathbf{Y})}\right)^4 \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^4 - 3,$$

where $sd(\mathbf{Y})$ and $\bar{y}$ are the standard deviation and the mean of the values in $\mathbf{Y}$, respectively. The third feature used from this set, namely Peaks($\mathbf{Y}$), first estimates the probability density function of the distribution of the values in the set $\mathbf{Y}$, and then finds the number of peaks in the distribution with a height of at least 0.01.

D.2 ELA levelset prediction

The ELA levelset prediction features [87] denoted $MMCE^q_{lda}$ calculate the classification error of a predictive linear model. That is, given a quantile $q \in \{0.1, 0.25, 0.5\}$, the data is split into two groups containing the locations at which the objective function value is above the qth quantile, and below the qth quantile. A linear model is then trained to predict whether a location has an objective function value which lies above or below the qth quantile. The resulting $MMCE^q_{lda}$ feature value indicates the probability of error for this prediction, with a value of 0 indicating a linear model can correctly predict which group a location lies in 100% of the time, and a value of 1 indicating a linear model incorrectly predicts this 100% of the time.

D.3 ELA meta modelling

The ELA meta modelling features [87] analyse how easily four simple models can model the observed data. Namely, given a problem dimension d, the models m_L, m_{LI}, m_Q and m_{QI} given by

$$
\begin{aligned}
m_L(\mathbf{x}) &= \alpha_0^L + \sum_{i=1}^{d} \alpha_i^L x_i, \\
m_{LI}(\mathbf{x}) &= \alpha_0^{LI} + \sum_{i=1}^{d} \alpha_i^{LI} x_i + \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \alpha_{i,j}^{LI} x_i x_j, \\
m_Q(\mathbf{x}) &= \alpha_0^Q + \sum_{i=1}^{d} \alpha_i^Q x_i^2, \\
m_{QI}(\mathbf{x}) &= \alpha_0^{QI} + \sum_{i=1}^{d} \alpha_i^{QI} x_i + \sum_{i=1}^{d} \alpha_{i,i}^{QI} x_i^2 + \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \alpha_{i,j}^{QI} x_i x_j,
\end{aligned}
$$

where the α coefficients are trained by least squares regression. The accuracy with which these models capture the behaviour of the data is measured using the adjusted R^2, that is the following features are calculated

$$\bar{R}_M = 1 - \frac{n-1}{n-k-1}\left(\frac{\sum_{i=1}^{d}(y_i - m_M(\mathbf{x}_i))^2}{\sum_{i=1}^{d}(y_i - \bar{y})^2}\right), \tag{D.1}$$

for every model m_M where n is the number of samples and k the number of independent variables of model m_M e.g. the feature $\bar{R}_L$ calculates the adjusted R^2 of model m_L. Two further features based on the models m_L and m_Q which have the coefficients $\alpha_0^L, \ldots, \alpha_d^L$ and $\alpha_0^Q, \ldots, \alpha_d^Q$, namely

$$CN_L = \frac{\max\{|\alpha_0^L|, \ldots, |\alpha_d^L|\}}{\min\{|\alpha_0^L|, \ldots, |\alpha_d^L|\}},$$
$$CN_Q = \frac{\max\{|\alpha_0^Q|, \ldots, |\alpha_d^Q|\}}{\min\{|\alpha_0^Q|, \ldots, |\alpha_d^Q|\}},$$

that is the features CN_L and CN_Q calculate the ratio of the minimum and maximum absolute coefficients of a linear and quadratic model without interactions, respectively.

D.4 Information content

The information content features are taken from the work of Muñoz et al. [88] and are based on the Information Content of Fitness Sequences (ICoFiS). These features measure the degree of smoothness and ruggedness of the landscape. From the provided sample $\mathbf{X}$, a "walk" through the space i.e. a sequence of locations $\mathbf{x}'_1, \ldots, \mathbf{x}'_n$ with objective function values $y'_1, \ldots, y'_N$ is generated by choosing a random location as a starting point and then iteratively choosing the nearest non-visited neighbour as the next "step". Given a parameter $\epsilon \in [10^{-5}, 10^{15}]$, a symbol sequence $\mathbf{\Phi}(\epsilon) = \{\phi_1, \ldots, \phi_{n-1}\}$ is defined with

$$\phi_i(\epsilon) = \begin{cases} \bar{1} \text{ if } \frac{y_{i+1}-y_i}{\|\mathbf{x}_{i-1}-\mathbf{x}_i\|} & < -\epsilon, \\ 0 \text{ if } \left|\frac{y_{i+1}-y_i}{\|\mathbf{x}_{i-1}-\mathbf{x}_i\|}\right| & \leq \epsilon, \\ 1 \text{ if } \frac{y_{i+1}-y_i}{\|\mathbf{x}_{i-1}-\mathbf{x}_i\|} & > \epsilon. \end{cases}$$

Given this sequence of symbols, the information content is given by

$$H(\epsilon) = -\sum_{a \neq b} p_{ab} \log_6(p_{ab}),$$

where $a, b \in \{\bar{1}, 0, 1\}$ and p_{ab} is the probability of finding the pair of symbols ab in the sequence. The value $H(\epsilon)$ lies in the range $[0, 1]$ and $0 \log_6(0) \equiv 0$. The sequence $\phi'(\epsilon)$ is next generated by removing all the zeros from the sequence $\mathbf{\Phi}'(\epsilon)$, and then removing any repeated symbols leading to a sequence $\ldots \bar{1}1\bar{1}1 \ldots$. This represents the changes in concavity of the walk and is used to measure the smoothness

$$M(\epsilon) = \frac{|\mathbf{\Phi}'(\epsilon)|}{n-1}.$$

Having defined these two measures of a walk, the features can now be defined

$$\begin{aligned}
H_{max} &= \max_{\epsilon}\{H(\epsilon)\}, \\
\epsilon_S &= \log_{10}(\min_{\epsilon}\{\epsilon : H(\epsilon) < 0.05\}), \\
M_0 &= M(0), \\
\epsilon_{max} &= \max_{\epsilon}\{\epsilon : H(\epsilon) = H_{max}\}, \\
\epsilon_{ratio} &= \log_{10}(\max_{\epsilon}\{\epsilon : M(\epsilon) > 0.5M_0\}).
\end{aligned}$$

Here H_{max} denotes the maximum information content of the sequence. The feature ϵ_S denotes the epsilon value for which the sequence contains 0 almost exclusively. The feature M_0 is known as the initial partial information and gives the highest number of inflection points (obtained when $\epsilon = 0$) in the $\mathbf{\Phi}'$ sequence, ϵ_{ratio} measures at which point the same sequence contains half the number of inflection points. Finally, ϵ_{max} denotes the largest ϵ value which leads to the highest information content value.

D.5 Nearest better clustering

Nearest better clustering features [89] are based on the comparison of a point's nearest neighbour and a point's nearest better neighbour i.e. a point with a better objective function value. First define the nearest neighbour d_{nn} and nearest better neighbour d_{nb}

functions

$$d_{nn}(\mathbf{x}_i, \mathbf{X}) = \min\left\{\{\|\mathbf{x}_i - \mathbf{x}_j\| \quad | \quad \mathbf{x}_j \in \mathbf{X} \setminus \{\mathbf{x}_i\}\}\right\},$$
$$d_{nb}(\mathbf{x}_i, \mathbf{X}) = \min\left\{\{\|\mathbf{x}_i - \mathbf{x}_j\| \quad | \quad y_j < y_i \text{ and } \mathbf{x}_j \in \mathbf{X} \setminus \{\mathbf{x}_i\}\}\right\}.$$

Note that for the point(s) with the lowest objective function value, the nearest better neighbour distance is defined to be the nearest neighbour distance. The sets $\mathcal{D}_{nn}$ and $\mathcal{D}_{nb}$ containing all of the nearest neighbour and nearest better neighbour distances, respectively can be defined along with the set $\mathcal{Q}_{nn/nb}$ containing the ratios of these distances

$$\mathcal{D}_{nn} = \{d_{nn}(\mathbf{x}_i, \mathbf{X}) \quad | \quad \mathbf{x}_i \in \mathbf{X}\},$$
$$\mathcal{D}_{nb} = \{d_{nb}(\mathbf{x}_i, \mathbf{X}) \quad | \quad \mathbf{x}_i \in \mathbf{X}\},$$
$$\mathcal{Q}_{nn/nb} = \left\{\frac{d_{nn}(\mathbf{x}_i, \mathbf{X})}{d_{nb}(\mathbf{x}_i, \mathbf{X})} \quad | \quad \mathbf{x}_i \in \mathbf{X}\right\}.$$

The first three features can now be defined which measure the distribution of the values in these sets

$$NBC_{mean} = \frac{\text{mean}(\mathcal{D}_{nn})}{\text{mean}(\mathcal{D}_{nb})},$$
$$NBC_{sd} = \frac{\text{sd}(\mathcal{D}_{nn})}{\text{sd}(\mathcal{D}_{nb})},$$
$$NBC_{coeff} = \frac{\text{sd}(\mathcal{Q}_{nn/nb})}{\text{mean}(\mathcal{Q}_{nn/nb})}.$$

Recall the correlation between two sets $\mathbf{A}$ and $\mathbf{B}$ with n entries and means $\bar{A}$ and $\bar{B}$ is given by

$$\text{corr}(\mathbf{A}, \mathbf{B}) = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{A})^2(b_i - \bar{B})^2}}.$$

Let $\deg^-(\mathbf{x}, \mathbf{X})$ be the number of points in $\mathbf{X} \setminus \{\mathbf{x}\}$ for which $\mathbf{x}$ is the nearest better neighbour, and

$$\mathcal{D}_{\text{deg}} = \{\deg^-(\mathbf{x}, \mathbf{X}) \quad | \quad \mathbf{x}_i \in \mathbf{X}\}.$$

The last two features are then defined as

$$NBC_{corr} = \text{corr}(\mathcal{D}_{nn}, \mathcal{D}_{nn}),$$
$$NBC_{fitcorr} = \text{corr}(Y, \mathcal{D}_{\text{deg}}).$$

D.6 Principal component analysis

The two features used from this feature set are the features PCA_{corr} and PCA_{cov}. To obtain the feature value for PCA_{corr}, the $(d+1)\times(d+1)$ correlation matrix is calculated (d inputs and one output). Next the amount of principal components required to explain 90% of the variability is calculated and divided by $d+1$ to obtain the relative amount. A similar process is followed to calculate PCA_{cov} by first constructing the covariance matrix.

D.7 Dispersion

Dispersion features [90] are used to compare distributions of the distances between sample points and the distances between the points with the best objective function values. Given a chosen ϵ% to separate the "best" points, let $\mathbf{X}_\epsilon$ be the set of points in $\mathbf{X}$ for which the objective function value lies in the bottom ϵ^{th} quantile. Then define the set of pairwise distances

$$\mathcal{D} = \{\|\mathbf{x}_i - \mathbf{x}_j\| \quad | \quad \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}\},$$
$$\mathcal{D}_\epsilon = \{\|\mathbf{x}_i - \mathbf{x}_j\| \quad | \quad \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_\epsilon\}.$$

Then the following two features are calculated for $\epsilon \in \{2, 5, 10\}$

$$\overline{DISP}_\epsilon = \frac{\text{mean}(\mathcal{D}_\epsilon)}{\text{mean}(\mathcal{D})},$$
$$\widetilde{DISP}_\epsilon = \frac{\text{median}(\mathcal{D}_\epsilon)}{\text{median}(\mathcal{D})}.$$

Appendix E

IMSPE derivation

This appendix provides the full derivation of the expressions used in Chapter 5. Specifically, details are given for the derivation of the functions $W(\mathbf{x}_1, \mathbf{x}_2), W^1(\mathbf{x}_1, \mathbf{x}_2), W^2(\mathbf{x}_1, \mathbf{x}_2)$ and $W^3(\mathbf{x}_1, \mathbf{x}_2)$, as well as the derivations of the closed form expressions for the change in $IMSPE$.

E.1 W functions

Recall the definition of the entries of the matrix $\mathbf{W}^{krig}$ given in Equation 5.1, namely

$$\begin{aligned}
\mathbf{W}_{i,j} &= W(\mathbf{x}_i, \mathbf{x}_j), \\
&= \int_{\mathbf{x}\in\Omega} \mathbf{r}(\mathbf{x})_i \mathbf{r}(\mathbf{x})_j d\mathbf{x}, \\
&= \int_{\mathbf{x}\in\Omega} R(\mathbf{x}, \mathbf{x}_i) R(\mathbf{x}, \mathbf{x}_j) d\mathbf{x}, \\
&= \int_{\mathbf{x}\in\Omega} \exp\left\{-\sum_{k=1}^{d} \theta_k |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k}\right\} \exp\left\{-\sum_{k=1}^{d} \theta_k |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k}\right\} d\mathbf{x}, \\
&= \int_{\mathbf{x}\in\Omega} \prod_{k=1}^{d} \exp\left\{\theta_k |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k} - \theta_k |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k}\right\} d\mathbf{x}, \\
&= \prod_{k=1}^{d} \int_{x_k^{\perp}}^{x_k^{\top}} \exp\left\{\theta_k |x - \mathbf{x}_i^{(k)}|^{p_k} - \theta_k |x - \mathbf{x}_j^{(k)}|^{p_k}\right\} dx.
\end{aligned}$$

Here $x_k^{\top}$ and $x_k^{\perp}$ denote the upper and lower bounds of the kth dimension of the hypercube domain Ω. As stated in Chapter 5, assuming $p_1 = \cdots = p_d = 2$ and $\Omega = [0, 1]^d$ leads to

the simplification

$$
\begin{aligned}
W(\mathbf{x}_i, \mathbf{x}_j) &= \prod_{k=1}^{d} \int_0^1 \exp\left\{\theta_k (x - \mathbf{x}_i^{(k)})^2 - \theta_k (x - \mathbf{x}_j^{(k)})^2\right\} dx, \\
&= \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k, \theta_k),
\end{aligned}
$$

where $w(\cdot,\cdot,\cdot,\cdot)$ is defined in Equation 5.2. Next recall the definition of the entries of the matrix $\mathbf{W}^{cokrig}$ given in Equation 5.5, namely

$$
\begin{aligned}
\mathbf{W}_{i,j}^{cokrig} &= \int_{\mathbf{x}\in\Omega} \mathbf{c}(\mathbf{x})_i \mathbf{c}(\mathbf{x})_j d\mathbf{x}, \\
&= \begin{cases}
W^1(\mathbf{x}_i^l, \mathbf{x}_j^l) & \text{for } 1 \le i, j \le n_l, \\
W^2(\mathbf{x}_i^l, \mathbf{x}_{j-n_l}^h) & \text{for } 1 \le i \le n_l < j \le n_l + n_h, \\
W^2(\mathbf{x}_j^l, \mathbf{x}_{i-n_l}^h) & \text{for } 1 \le j \le n_l < i \le n_l + n_h, \\
W^3(\mathbf{x}_{i-n_l}^h, \mathbf{x}_{j-n_l}^h) & \text{for } n_l < i, j \le n_l + n_h,
\end{cases}
\end{aligned}
$$

with the functions W^1, W^2 and W^3 defined as

$$
\begin{aligned}
W^1(\mathbf{x}_i, \mathbf{x}_j) &= \int_{\mathbf{x}\in\Omega} \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}, \mathbf{x}_i) \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}, \mathbf{x}_j) d\mathbf{x}, \\
W^2(\mathbf{x}_i, \mathbf{x}_j) &= \int_{\mathbf{x}\in\Omega} \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}, \mathbf{x}_i) \left(\hat{\rho}^2\hat{\sigma}_l^2 R_l(\mathbf{x}, \mathbf{x}_j) + \hat{\sigma}_{diff}^2 R_{diff}(\mathbf{x}, \mathbf{x}_j)\right) d\mathbf{x}, \\
W^3(\mathbf{x}_i, \mathbf{x}_j) &= \int_{\mathbf{x}\in\Omega} \left(\hat{\rho}^2\hat{\sigma}_l^2 R_l(\mathbf{x}, \mathbf{x}_i) + \hat{\sigma}_{diff}^2 R_{diff}(\mathbf{x}, \mathbf{x}_i)\right) \cdot \\
&\qquad \left(\hat{\rho}^2\hat{\sigma}_l^2 R_l(\mathbf{x}, \mathbf{x}_j) + \hat{\sigma}_{diff}^2 R_{diff}(\mathbf{x}, \mathbf{x}_j)\right) d\mathbf{x}.
\end{aligned}
$$

Some manipulation leads to

$$
\begin{aligned}
W^1(\mathbf{x}_i, \mathbf{x}_j) &= \int_{\mathbf{x}\in\Omega} \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}, \mathbf{x}_i) \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}, \mathbf{x}_j) d\mathbf{x}, \\
&= \left(\hat{\rho}\hat{\sigma}_l^2\right)^2 \int_{\mathbf{x}\in\Omega} \prod_{k=1}^{d} \exp\left\{-\theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k^l} - \theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k^l}\right\} d\mathbf{x}, \\
&= \left(\hat{\rho}\hat{\sigma}_l^2\right)^2 \prod_{k=1}^{d} \int_{x_k^\perp}^{x_k^\top} \exp\left\{-\theta_k^l |x - \mathbf{x}_i^{(k)}|^{p_k^l} - \theta_k^l |x - \mathbf{x}_j^{(k)}|^{p_k^l}\right\} dx,
\end{aligned}
$$

$$
\begin{aligned}
W^2(\mathbf{x}_i, \mathbf{x}_j) &= \int_{\mathbf{x}\in\Omega} \hat{\rho}\hat{\sigma}_l^2 R_l(\mathbf{x}, \mathbf{x}_i) \left(\hat{\rho}^2 \hat{\sigma}_l^2 R_l(\mathbf{x}, \mathbf{x}_j) + \hat{\sigma}_{diff}^2 R_{diff}(\mathbf{x}, \mathbf{x}_j)\right) d\mathbf{x}, \\
&= \hat{\rho}^3 \left(\hat{\sigma}_l^2\right)^2 \int_{\mathbf{x}\in\Omega} R_l(\mathbf{x}, \mathbf{x}_i) R_l(\mathbf{x}, \mathbf{x}_j) d\mathbf{x} \\
&\quad + \hat{\rho}\hat{\sigma}_l^2 \hat{\sigma}_{diff}^2 \int_{\mathbf{x}\in\Omega} R_l(\mathbf{x}, \mathbf{x}_i) R_{diff}(\mathbf{x}, \mathbf{x}_j) d\mathbf{x}, \\
&= \hat{\rho}^3 \left(\hat{\sigma}_l^2\right)^2 \int_{\mathbf{x}\in\Omega} \prod_{k=1}^{d} \exp\left\{-\theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k^l} - \theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k^l}\right\} d\mathbf{x} \\
&\quad + \hat{\rho}\hat{\sigma}_l^2 \hat{\sigma}_{diff}^2 \int_{\mathbf{x}\in\Omega} \prod_{k=1}^{d} \exp\left\{-\theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k^l} - \theta_k^{diff} |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k^{diff}}\right\} d\mathbf{x}, \\
&= \hat{\rho}^3 \left(\hat{\sigma}_l^2\right)^2 \prod_{k=1}^{d} \int_{x_k^{\perp}}^{x_k^{\top}} \exp\left\{-\theta_k^l |x - \mathbf{x}_i^{(k)}|^{p_k^l} - \theta_k^l |x - \mathbf{x}_j^{(k)}|^{p_k^l}\right\} dx \\
&\quad + \hat{\rho}\hat{\sigma}_l^2 \hat{\sigma}_{diff}^2 \prod_{k=1}^{d} \int_{x_k^{\perp}}^{x_k^{\top}} \exp\left\{-\theta_k^l |x - \mathbf{x}_i^{(k)}|^{p_k^l} - \theta_k^{diff} |x - \mathbf{x}_j^{(k)}|^{p_k^{diff}}\right\} dx,
\end{aligned}
$$

$$
\begin{aligned}
W^3(\mathbf{x}_i,\mathbf{x}_j) &= \int_{\mathbf{x}\in\Omega} \left(\hat{\rho}^2\hat{\sigma}_l^2 R_l(\mathbf{x},\mathbf{x}_i) + \hat{\sigma}_{diff}^2 R_{diff}(\mathbf{x},\mathbf{x}_i)\right)\left(\hat{\rho}^2\hat{\sigma}_l^2 R_l(\mathbf{x},\mathbf{x}_j) + \hat{\sigma}_{diff}^2 R_{diff}(\mathbf{x},\mathbf{x}_j)\right) d\mathbf{x}, \\
&= \left(\hat{\rho}^2\hat{\sigma}_l^2\right)^2 \int_{\mathbf{x}\in\Omega} R_l(\mathbf{x},\mathbf{x}_i) R_l(\mathbf{x},\mathbf{x}_j) d\mathbf{x} \\
&\quad + \hat{\rho}^2\hat{\sigma}_l^2\hat{\sigma}_{diff}^2 \int_{\mathbf{x}\in\Omega} R_l(\mathbf{x},\mathbf{x}_i) R_{diff}(\mathbf{x},\mathbf{x}_j) d\mathbf{x} \\
&\quad + \hat{\rho}^2\hat{\sigma}_l^2\hat{\sigma}_{diff}^2 \int_{\mathbf{x}\in\Omega} R_{diff}(\mathbf{x},\mathbf{x}_i) R_l(\mathbf{x},\mathbf{x}_j) d\mathbf{x} \\
&\quad + \left(\hat{\sigma}_{diff}^2\right)^2 \int_{\mathbf{x}\in\Omega} R_{diff}(\mathbf{x},\mathbf{x}_i) R_{diff}(\mathbf{x},\mathbf{x}_j) d\mathbf{x}, \\
&= \left(\hat{\rho}^2\hat{\sigma}_l^2\right)^2 \int_{\mathbf{x}\in\Omega} \prod_{k=1}^{d} \exp\left\{-\theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k^l} - \theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k^l}\right\} d\mathbf{x} \\
&\quad + \hat{\rho}^2\hat{\sigma}_l^2\hat{\sigma}_{diff}^2 \int_{\mathbf{x}\in\Omega} \prod_{k=1}^{d} \exp\left\{-\theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k^l} - \theta_k^{diff} |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k^{diff}}\right\} d\mathbf{x} \\
&\quad + \hat{\rho}^2\hat{\sigma}_l^2\hat{\sigma}_{diff}^2 \int_{\mathbf{x}\in\Omega} \prod_{k=1}^{d} \exp\left\{-\theta_k^{diff} |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k^{diff}} - \theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k^l}\right\} d\mathbf{x} \\
&\quad + \left(\hat{\sigma}_{diff}^2\right)^2 \int_{\mathbf{x}\in\Omega} \prod_{k=1}^{d} \exp\left\{-\theta_k^{diff} |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k^{diff}} - \theta_k^{diff} |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k^{diff}}\right\} d\mathbf{x}, \\
&= \left(\hat{\rho}^2\hat{\sigma}_l^2\right)^2 \prod_{k=1}^{d} \int_{x_k^{\perp}}^{x_k^{\top}} \exp\left\{-\theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k^l} - \theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k^l}\right\} dx \\
&\quad + \hat{\rho}^2\hat{\sigma}_l^2\hat{\sigma}_{diff}^2 \prod_{k=1}^{d} \int_{x_k^{\perp}}^{x_k^{\top}} \exp\left\{-\theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k^l} - \theta_k^{diff} |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k^{diff}}\right\} dx \\
&\quad + \hat{\rho}^2\hat{\sigma}_l^2\hat{\sigma}_{diff}^2 \prod_{k=1}^{d} \int_{x_k^{\perp}}^{x_k^{\top}} \exp\left\{-\theta_k^{diff} |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k^{diff}} - \theta_k^l |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k^l}\right\} dx \\
&\quad + \left(\hat{\sigma}_{diff}^2\right)^2 \prod_{k=1}^{d} \int_{x_k^{\perp}}^{x_k^{\top}} \exp\left\{-\theta_k^{diff} |\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}|^{p_k^{diff}} - \theta_k^{diff} |\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}|^{p_k^{diff}}\right\} dx.
\end{aligned}
$$

Once again, assuming $\Omega = [0,1]^d$, and $p_1^l = \cdots = p_d^l = p_1^{diff} = \cdots = p_d^{diff} = 2$ we obtain

$$
\begin{aligned}
W^1(\mathbf{x}_i, \mathbf{x}_j) &= \left(\hat{\rho}\hat{\sigma}_l^2\right)^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^l, \theta_k^l), \\
W^2(\mathbf{x}_i, \mathbf{x}_j) &= \hat{\rho}^3 \left(\hat{\sigma}_l^2\right)^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^l, \theta_k^l) \\
&\quad + \hat{\rho}\hat{\sigma}_l^2\hat{\sigma}_{diff}^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^l, \theta_k^{diff}), \\
W^3(\mathbf{x}_i, \mathbf{x}_j) &= \left(\hat{\rho}^2\hat{\sigma}_l^2\right)^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^l, \theta_k^l) \\
&\quad + \hat{\rho}^2\hat{\sigma}_l^2\hat{\sigma}_{diff}^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^l, \theta_k^{diff}) \\
&\quad + \hat{\rho}^2\hat{\sigma}_l^2\hat{\sigma}_{diff}^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^{diff}, \theta_k^l) \\
&\quad + \left(\hat{\sigma}_{diff}^2\right)^2 \prod_{k=1}^{d} w(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \theta_k^{diff}, \theta_k^{diff}).
\end{aligned}
$$

E.2 Change in Kriging IMSPE

In order to derive the closed form expression of $\Delta_{\mathbf{x}_{next}} IMSPE$ given in Equation 5.3, first expressions are needed for $\mathbf{R}_{\mathbf{x}_{next}}^{-1}$ and $\mathbf{W}_{\mathbf{x}_{next}}^{krig}$, namely the updated $\mathbf{R}^{-1}$ and $\mathbf{W}^{krig}$ after sampling f_h at $\mathbf{x}$. Recall that $\mathbf{r}(\cdot)$ is the Kriging vector defined in Equation 2.1. Then

$$
\mathbf{R}_{\mathbf{x}_{next}} = \begin{bmatrix} \mathbf{R} & \mathbf{r}(\mathbf{x}_{next}) \\ \mathbf{r}(\mathbf{x}_{next})^\top & 1 \end{bmatrix}.
$$

Using matrix block inversion and defining $\alpha = (1 - \mathbf{r}(\mathbf{x}_{next})^\top \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}_{next}))^{-1}$, the inverse of this matrix is given by

$$
\mathbf{R}_{\mathbf{x}_{next}}^{-1} = \begin{bmatrix} \mathbf{R}^{-1} + \alpha \mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_{next})\mathbf{r}(\mathbf{x}_{next})^\top \mathbf{R}^{-1} & -\alpha \mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_{next}) \\ -\alpha \mathbf{r}(\mathbf{x}_{next})^\top \mathbf{R}^{-1} & \alpha \end{bmatrix}.
$$

The matrix $\mathbf{W}^{krig}_{\mathbf{x}_{next}}$ is defined as

$$\mathbf{W}^{krig}_{\mathbf{x}_{next}} = \begin{bmatrix} \mathbf{W}^{krig} & \mathbf{w}^{krig}(\mathbf{x}_{next}) \\ \mathbf{w}^{krig}(\mathbf{x}_{next})^\top & W(\mathbf{x}_{next}, \mathbf{x}_{next}) \end{bmatrix},$$

with $\mathbf{w}^{krig}(\mathbf{x})$ defined in Equation 5.4. Now note that $tr(\mathbf{R}^{-1}\mathbf{W}^{krig}) = \mathbf{1}^\top(\mathbf{R}^{-1} \circ \mathbf{W}^{krig})\mathbf{1}$ where $\circ$ stands for the element-wise product and $\mathbf{1}$ is a vector containing 1 in all its entries. Then

$$\begin{aligned}
\Delta_{\mathbf{x}_{next}} IMSPE &= \int_{\mathbf{x}\in\Omega} v^{krig}(\mathbf{x})d\mathbf{x} - \int_{\mathbf{x}\in\Omega} v^{krig}_{\mathbf{x}_{next}}(\mathbf{x})d\mathbf{x}, \\
&= \hat{\sigma}^2 \left[tr(\mathbf{R}^{-1}\mathbf{W}^{krig}) - tr(\mathbf{R}^{-1}_{\mathbf{x}_{next}}\mathbf{W}^{krig}_{\mathbf{x}_{next}}) \right], \\
&= -\hat{\sigma}^2 \left[\mathbf{1}^\top(\mathbf{R}^{-1} \circ \mathbf{W}^{krig})\mathbf{1} - \mathbf{1}^\top(\mathbf{R}^{-1}_{\mathbf{x}^h_{next}} \circ \mathbf{W}^{krig}_{\mathbf{x}^h_{next}})\mathbf{1} \right], \\
&= -\hat{\sigma}^2 \Big(\mathbf{1}^\top(\mathbf{R}^{-1} \circ \mathbf{W}^{krig})\mathbf{1} \\
&\qquad - \mathbf{1}^\top(\mathbf{R} \circ \mathbf{W}^{krig})\mathbf{1} \\
&\qquad - \alpha \mathbf{r}(\mathbf{x}_{next})^\top \mathbf{R}^{-1}\mathbf{W}^{krig}\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_{next}) \\
&\qquad + 2\alpha \mathbf{w}^{krig}(\mathbf{x}_{next})^\top \mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_{next}) \\
&\qquad - \alpha W(\mathbf{x}_{next}, \mathbf{x}_{next}) \Big), \\
&= \frac{\hat{\sigma}^2}{1 - \mathbf{r}(\mathbf{x}_{next})^\top \mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_{next})} \Big[\mathbf{r}(\mathbf{x}_{next})^\top \mathbf{R}^{-1}\mathbf{W}^{krig}\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_{next}) \\
&\qquad - 2\mathbf{w}^{krig}(\mathbf{x}_{next})^\top \mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_{next}) + W(\mathbf{x}_{next}, \mathbf{x}_{next}) \Big],
\end{aligned}$$

as required.

E.3 Change in Co-Kriging IMSPE

Next the derivation of closed form expression for $\Delta^l_{\mathbf{x}_{next}} IMSPE$, $\Delta^h_{\mathbf{x}_{next}} IMSPE$ and $\Delta^b_{\mathbf{x}_{next}} IMSPE$ given in Equations 5.6, 5.9 and 5.11 is provided. First recall that the matrices $\mathbf{C}_{(l,\mathbf{x}_{next})}$, $\mathbf{C}_{(h,\mathbf{x}_{next})}$ and $\mathbf{C}_{(b,\mathbf{x}_{next})}$ denote the updated $\mathbf{C}$ matrix of the Co-Kriging model after only sampling f_l, only sampling f_h, or sampling both f_l and f_h at

$\mathbf{x}_{next}$, respectively. These are defined as

$$\mathbf{C}_{(l,\mathbf{x}_{next})} = \begin{bmatrix} \hat{\sigma}_l^2 & \mathbf{c}_l(\mathbf{x}_{next}) \\ \mathbf{c}_l(\mathbf{x}_{next})^\top & \mathbf{C} \end{bmatrix},$$

$$\mathbf{C}_{(h,\mathbf{x}_{next})} = \begin{bmatrix} \mathbf{C} & \mathbf{c}(\mathbf{x}_{next}) \\ \mathbf{c}(\mathbf{x}_{next})^\top & \hat{\rho}^2\hat{\sigma}_l^2 + \hat{\sigma}_{diff}^2 \end{bmatrix},$$

$$\mathbf{C}_{(b,\mathbf{x}_{next})} = \begin{bmatrix} \mathbf{C} & \mathbf{c}_b(\mathbf{x}_{next}) \\ \mathbf{c}_b(\mathbf{x}_{next})^\top & \Sigma \end{bmatrix},$$

with $\mathbf{c}_l(\cdot)$, $\mathbf{c}(\cdot)$, $\mathbf{c}_b(\cdot)$ and Σ defined in Equations 5.8, 2.6, 5.14 and 5.16. Using block matrix inversion be obtain

$$\mathbf{C}_{(l,\mathbf{x}_{next})}^{-1} = \begin{bmatrix} \alpha_l & -\alpha_l\mathbf{c}_l(\mathbf{x}_{next})^\top\mathbf{C}^{-1} \\ -\alpha_l\mathbf{C}^{-1}\mathbf{c}_l(\mathbf{x}_{next}) & \mathbf{C}^{-1} + \alpha_l\mathbf{C}^{-1}\mathbf{c}_l(\mathbf{x}_{next})\mathbf{c}_l(\mathbf{x}_{next})^\top\mathbf{C}^{-1} \end{bmatrix},$$

$$\mathbf{C}_{(h,\mathbf{x}_{next})}^{-1} = \begin{bmatrix} \mathbf{C}^{-1} + \alpha_h\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_{next})\mathbf{c}(\mathbf{x}_{next})^\top\mathbf{C}^{-1} & -\alpha_h\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_{next}) \\ -\alpha_h\mathbf{c}(\mathbf{x}_{next})^\top\mathbf{C}^{-1} & \alpha_h \end{bmatrix},$$

$$\mathbf{C}_{(b,\mathbf{x}_{next})}^{-1} = \begin{bmatrix} \mathbf{C}^{-1} + \mathbf{C}^{-1}\mathbf{c}_b(\mathbf{x}_{next})\mathbf{A}_b\mathbf{c}_b(\mathbf{x}_{next})^\top\mathbf{C}^{-1} & -\mathbf{C}^{-1}\mathbf{c}_b(\mathbf{x}_{next})\mathbf{A}_b \\ -\mathbf{A}_b\mathbf{c}_b(\mathbf{x}_{next})^\top\mathbf{C}^{-1} & \mathbf{A}_b \end{bmatrix},$$

$$\begin{aligned} \alpha_l &= (\hat{\sigma}_l^2 - \mathbf{c}_l(\mathbf{x}_{next})^\top\mathbf{C}^{-1}\mathbf{c}_l(\mathbf{x}_{next}))^{-1}, \\ \alpha_h &= (\hat{\rho}^2\hat{\sigma}_l^2 + \hat{\sigma}_{diff}^2 - \mathbf{c}(\mathbf{x}_{next})^\top\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_{next}))^{-1}, \\ \mathbf{A}_b &= (\Sigma - \mathbf{c}_b(\mathbf{x}_{next})^\top\mathbf{C}^{-1}\mathbf{c}_b(\mathbf{x}_{next}))^{-1}. \end{aligned}$$

Similarly, define $\mathbf{W}_{(l,\mathbf{x}_{next})}^{cokrig}$, $\mathbf{W}_{(h,\mathbf{x}_{next})}^{cokrig}$ and $\mathbf{W}_{(b,\mathbf{x}_{next})}^{cokrig}$ as the updated $\mathbf{W}^{cokrig}$ matrix after only sampling f_l, only sampling f_h, or sampling both f_l and f_h at $\mathbf{x}_{next}$, respectively.

These are defined as

$$\mathbf{W}^{cokrig}_{(l,\mathbf{x}_{next})} = \begin{bmatrix} W^1(\mathbf{x}_{next},\mathbf{x}_{next}) & \mathbf{w}_l^{cokrig}(\mathbf{x}_{next})^\top \\ \mathbf{w}_l^{cokrig}(\mathbf{x}_{next}) & \mathbf{W}^{cokrig} \end{bmatrix},$$

$$\mathbf{W}^{cokrig}_{(h,\mathbf{x}_{next})} = \begin{bmatrix} \mathbf{W}^{cokrig} & \mathbf{w}_h^{cokrig}(\mathbf{x}_{next}) \\ \mathbf{w}_h^{cokrig}(\mathbf{x}_{next})^\top & W^3(\mathbf{x}_{next},\mathbf{x}_{next}) \end{bmatrix},$$

$$\mathbf{W}^{cokrig}_{(b,\mathbf{x}_{next})} = \begin{bmatrix} \mathbf{W}^{cokrig} & \mathbf{w}_b^{cokrig}(\mathbf{x}_{next}) \\ \mathbf{w}_b^{cokrig}(\mathbf{x}_{next})^\top & \mathbf{W}_b^{cokrig} \end{bmatrix},$$

with $\mathbf{w}_l^{cokrig}(\cdot)$, $\mathbf{w}_h^{cokrig}(\cdot)$, $\mathbf{w}_b^{cokrig}(\cdot)$ and $\mathbf{W}_b^{cokrig}$ defined in Equations 5.7, 5.10, 5.12 and 5.13. Once again, noting that $tr(\mathbf{C}^{-1}\mathbf{W}^{cokrig}) = \mathbf{1}^\top(\mathbf{C}^{-1}\circ\mathbf{W}^{cokrig})\mathbf{1}$, the following can be derived

$$\begin{aligned}
\Delta^l_{\mathbf{x}_{next}} IMSPE &= \int_{\mathbf{x}\in\Omega} v^{cokrig}(\mathbf{x})d\mathbf{x} - \int_{\mathbf{x}\in\Omega} v^{cokrig}_{(l,\mathbf{x}_{next})}(\mathbf{x})d\mathbf{x}, \\
&= -tr(\mathbf{C}^{-1}\mathbf{W}^{cokrig}) + tr\left(\mathbf{C}^{-1}_{(l,\mathbf{x}_{next})}\mathbf{W}^{cokrig}_{(l,\mathbf{x}_{next})}\right), \\
&= -\mathbf{1}^\top(\mathbf{C}^{-1}\circ\mathbf{W}^{cokrig})\mathbf{1} \\
&\quad + \alpha_l W^1(\mathbf{x}_{next},\mathbf{x}_{next}) \\
&\quad - 2\alpha_l \mathbf{w}_l^{cokrig}(\mathbf{x}_{next})^\top\mathbf{C}^{-1}\mathbf{c}_l(\mathbf{x}_{next}) \\
&\quad + \alpha_l \mathbf{c}_l(\mathbf{x}_{next})^\top\mathbf{C}^{-1}\mathbf{W}^{cokrig}\mathbf{C}^{-1}\mathbf{c}_l(\mathbf{x}_{next}) \\
&\quad + \mathbf{1}^\top(\mathbf{C}^{-1}\circ\mathbf{W}^{cokrig})\mathbf{1},
\end{aligned}$$

$$\begin{aligned}
= \frac{1}{\hat{\sigma}_l^2 - \mathbf{c}_l(\mathbf{x}_{next})^\top\mathbf{C}^{-1}\mathbf{c}_l(\mathbf{x}_{next})}\Big[&\mathbf{c}_l^\top(\mathbf{x}_{next})\mathbf{C}^{-1}\mathbf{W}^{cokrig}\mathbf{C}^{-1}\mathbf{c}_l(\mathbf{x}_{next}), \\
&- 2\mathbf{w}_l^{cokrig}(\mathbf{x}_{next})^\top\mathbf{C}^{-1}\mathbf{c}_l(\mathbf{x}_{next}) + W^1(\mathbf{x}_{next},\mathbf{x}_{next})\Big],
\end{aligned}$$

$$
\begin{aligned}
\Delta^{h}_{\mathbf{x}_{next}} IMSPE &= \int_{\mathbf{x}\in\Omega} v^{cokrig}(\mathbf{x})d\mathbf{x} - \int_{\mathbf{x}\in\Omega} v^{cokrig}_{(h,\mathbf{x}_{next})}(\mathbf{x})d\mathbf{x}, \\
&= -tr(\mathbf{C}^{-1}\mathbf{W}^{cokrig}) + tr\left(\mathbf{C}^{-1}_{(h,\mathbf{x}_{next})}\mathbf{W}^{cokrig}_{(h,\mathbf{x}_{next})}\right), \\
&= -\mathbf{1}^{\top}(\mathbf{C}^{-1}\circ\mathbf{W}^{cokrig})\mathbf{1} \\
&\quad + \alpha_h \mathbf{c}(\mathbf{x}_{next})^{\top}\mathbf{C}^{-1}\mathbf{W}^{cokrig}\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_{next}) \\
&\quad + \mathbf{1}^{\top}(\mathbf{C}^{-1}\circ\mathbf{W}^{cokrig})\mathbf{1} \\
&\quad - 2\alpha_h \mathbf{w}^{cokrig}_{h}(\mathbf{x}_{next})^{\top}\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_{next}) \\
&\quad + \alpha_h W^3(\mathbf{x}_{next},\mathbf{x}_{next}), \\
&= \frac{1}{\hat{\rho}^2\hat{\sigma}_l^2 + \hat{\sigma}^2_{diff} - \mathbf{c}(\mathbf{x}_{next})^{\top}\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_{next})}\Big[\mathbf{c}(\mathbf{x}_{next})^{\top}\mathbf{C}^{-1}\mathbf{W}^{cokrig}\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_{next}) \\
&\qquad - 2\mathbf{w}^{cokrig}_{h}(\mathbf{x}_{next})^{\top}\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_{next}) + W^3(\mathbf{x}_{next},\mathbf{x}_{next})\Big],
\end{aligned}
$$

$$
\begin{aligned}
\Delta^{b}_{\mathbf{x}_{next}} IMSPE &= \int_{\mathbf{x}\in\Omega} v^{cokrig}(\mathbf{x})d\mathbf{x} - \int_{\mathbf{x}\in\Omega} v^{cokrig}_{(b,\mathbf{x}_{next})}(\mathbf{x})d\mathbf{x}, \\
&= -tr(\mathbf{C}^{-1}\mathbf{W}^{cokrig}) + tr\left(\mathbf{C}^{-1}_{(b,\mathbf{x}_{next})}\mathbf{W}^{cokrig}_{(b,\mathbf{x}_{next})}\right), \\
&= -\mathbf{1}^{\top}(\mathbf{C}^{-1}\circ\mathbf{W}^{cokrig})\mathbf{1} \\
&\quad + \mathbf{1}^{\top}(\mathbf{C}^{-1}\circ\mathbf{W}^{cokrig})\mathbf{1} \\
&\quad + \mathbf{1}^{\top}\Big(\mathbf{C}^{-1}\mathbf{c}_b(\mathbf{x}_{next})\mathbf{A}_b\mathbf{c}_b(\mathbf{x}_{next})^{\top}\mathbf{C}^{-1}\circ\mathbf{W}^{cokrig}\Big)\mathbf{1} \\
&\quad - \mathbf{1}^{\top}\Big(\mathbf{C}^{-1}\mathbf{c}_b(\mathbf{x}_{next})\mathbf{A}_b\circ\mathbf{w}^{cokrig}_{b}(\mathbf{x}_{next})\Big)\mathbf{1} \\
&\quad - \mathbf{1}^{\top}\Big(\mathbf{A}_b\mathbf{c}_b(\mathbf{x}_{next})^{\top}\mathbf{C}^{-1}\circ\mathbf{w}^{cokrig}_{b}(\mathbf{x}_{next})^{\top}\Big)\mathbf{1} \\
&\quad + \mathbf{1}^{\top}\Big(\mathbf{A}_b\circ\mathbf{W}^{cokrig}_{b}\Big)\mathbf{1}, \\
&= \mathbf{1}^{\top}\Big(\mathbf{C}^{-1}\mathbf{c}_b(\mathbf{x}_{next})\mathbf{A}_b\mathbf{c}_b(\mathbf{x}_{next})^{\top}\mathbf{C}^{-1}\circ\mathbf{W}^{cokrig}\Big)\mathbf{1} \\
&\quad - \mathbf{1}^{\top}\Big(\mathbf{C}^{-1}\mathbf{c}_b(\mathbf{x}_{next})\mathbf{A}_b\circ\mathbf{w}^{cokrig}_{b}(\mathbf{x}_{next})\Big)\mathbf{1} \\
&\quad - \mathbf{1}^{\top}\Big(\mathbf{A}_b\mathbf{c}_b(\mathbf{x}_{next})^{\top}\mathbf{C}^{-1}\circ\mathbf{w}^{cokrig}_{b}(\mathbf{x}_{next})^{\top}\Big)\mathbf{1} \\
&\quad + \mathbf{1}^{\top}\Big(\mathbf{A}_b\circ\mathbf{W}^{cokrig}_{b}\Big)\mathbf{1}.
\end{aligned}
$$

Bibliography

[1] KR Quinlan, J Movva, EV Stein, and A Kupresanin. Multi-fidelity aerodynamic databases for efficient representation of hypersonic design spaces. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2021.

[2] Xinwang Liu, Weiwen Zhao, and Decheng Wan. Multi-fidelity co-kriging surrogate model for ship hull form optimization. *Ocean Engineering*, 243:110239, 2022.

[3] Donald R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

[4] M Giselle Fernández-Godino, Chanyoung Park, Nam H Kim, and Raphael T Haftka. Issues in deciding whether to use multifidelity surrogates. *AIAA Journal*, 57(5): 2039–2054, 2019.

[5] Hayriye Pehlivan Solak, Jeroen Wackers, Riccardo Pellegrini, Andrea Serani, Matteo Diez, Paolo Perali, Matthieu Sacher, Jean-Baptiste Leroux, Benoit Augier, Frédéric Hauville, et al. Hydrofoil optimization via automated multi-fidelity surrogate models. In *10th International Conference on Computational Methods in Marine Engineering (MARINE 2023)*, 2023.

[6] Thomas Scholcz and Joy Klinkenberg. Hull-shape optimisation using adaptive multi-fidelity kriging. In *NATO-AVT-354 workshop on multi-fidelity methods for military vehicle design*. NATO, 2022.

[7] Tom Savage, Nausheen Basha, Jonathan McDonough, Omar K Matar, and Ehecatl Antonio del Rio Chanona. Multi-fidelity data-driven design and analysis of reactor and tube simulations. *Computers & Chemical Engineering*, page 108410, 2023.

[8] Basem Elshafei, Alfredo Peña, Dong Xu, Jie Ren, Jake Badger, Felipe M Pimenta, Donald Giddings, and Xuerui Mao. A hybrid solution for offshore wind resource assessment from limited onshore measurements. *Applied Energy*, 298:117245, 2021.

[9] Negin Alemazkoor and Hadi Meidani. A data-driven multi-fidelity approach for traffic state estimation using data from multiple sources. *IEEE Access*, 9:78128–78137, 2021.

[10] Si Cheng, Bledar A Konomi, Jessica L Matthews, Georgios Karagiannis, and Emily L Kang. Hierarchical Bayesian nearest neighbor co-kriging Gaussian process models; an application to intersatellite calibration. *Spatial Statistics*, 44:100516, 2021.

[11] Xuhao Peng, Jiaqing Kou, and Weiwei Zhang. Multi-fidelity nonlinear unsteady aerodynamic modeling and uncertainty estimation based on hierarchical kriging. *Applied Mathematical Modelling*, 2023.

[12] Ruochen Wang, Yanping Yang, Xiangyu Wang, Bo Wang, and Guoxin Zhang. Co-kriging based multi-fidelity aerodynamic optimization for flying wing uav with multi-shape wingtip design. In *2021 IEEE International Conference on Unmanned Systems (ICUS)*, pages 93–98. IEEE, 2021.

[13] Sunwoong Yang and Kwanjung Yee. Design rule extraction using multi-fidelity surrogate model for unmanned combat aerial vehicles. *Journal of Aircraft*, 59(4): 977–991, 2022.

[14] Rubén Conde Arenzana, Andrés F López-Lopera, Sylvain Mouton, Nathalie Bartoli, and Thierry Lefebvre. Multi-fidelity gaussian process model for CFD and wind tunnel data fusion. In *AeroBest 2021*, 2021.

[15] Huseyin B Kurt, Murat Millidere, Fazil S Gomec, and Omur Ugur. Multi-fidelity aerodynamic dataset generation of a fighter aircraft. In *AIAA Scitech 2021 Forum*, page 0544, 2021.

[16] Nikolaos Mourousias, Ahmed Malim, Benoît G Marinus, and Mark Runacres. Assessment of multi-fidelity surrogate models for high-altitude propeller optimization. In *AIAA AVIATION 2022 Forum*, page 3752, 2022.

[17] Brian L Stevens, Frank L Lewis, and Eric N Johnson. *Aircraft control and simulation: dynamics, controls design, and autonomous systems.* John Wiley & Sons, 2015.

[18] David JJ Toal. Some considerations regarding the use of multi-fidelity kriging in the construction of surrogate models. *Structural and Multidisciplinary Optimization*, 51 (6):1223–1245, 2015.

[19] Bruce Ankenman, Barry L Nelson, and Jeremy Staum. Stochastic kriging for simulation metamodeling. In *2008 Winter simulation conference*, pages 362–370. IEEE, 2008.

[20] Handing Wang, Yaochu Jin, and John Doherty. A generic test suite for evolutionary multifidelity optimization. *IEEE Transactions on Evolutionary Computation*, 22(6): 836–850, 2017.

[21] Sander van Rijn, Sebastian Schmitt, Matthijs van Leeuwen, and Thomas Bäck. Finding efficient trade-offs in multi-fidelity response surface modelling. *Engineering Optimization*, 55(6):946–963, 2023.

[22] Juliane Müller. An algorithmic framework for the optimization of computationally expensive bi-fidelity black-box problems. *INFOR: Information Systems and Operational Research*, 58(2):264–289, 2020.

[23] Daniel G Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.

[24] Georges Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

[25] Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.

[26] H-M Gutmann. A radial basis function method for global optimization. *Journal of global optimization*, 19(3):201–227, 2001.

[27] Rommel G Regis and Christine A Shoemaker. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing*, 19(4):497–509, 2007.

[28] Stefan M Wild, Rommel G Regis, and Christine A Shoemaker. Orbit: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, 30(6):3197–3219, 2008.

[29] Juliane Müller and Christine A Shoemaker. Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *Journal of Global Optimization*, 60(2):123–144, 2014.

[30] Juliane Müller and Robert Piché. Mixture surrogate models based on dempster-shafer theory for global optimization problems. *Journal of Global Optimization*, 51: 79–104, 2011.

[31] Georges Matheron. The intrinsic random functions and their applications. *Advances in applied probability*, 5(3):439–468, 1973.

[32] Pramudita Satria Palar, Lucia Parussini, Luigi Bregant, Koji Shimoyama, and Lavi Rizki Zuhal. On kernel functions for bi-fidelity gaussian process regressions. *Structural and Multidisciplinary Optimization*, 66(2):37, 2023.

[33] David JJ Toal, NW Bressloff, AJ Keane, and CME Holden. The development of a hybridized particle swarm for kriging hyperparameter tuning. *Engineering optimization*, 43(6):675–699, 2011.

[34] David JJ Toal, Neil W Bressloff, and Andy J Keane. Kriging hyperparameter tuning strategies. *AIAA journal*, 46(5):1240–1252, 2008.

[35] MJ Appel, R Labarre, and D Radulovic. On accelerated random search. *SIAM Journal on Optimization*, 14(3):708–731, 2004.

[36] Zhong-Hua Han and Stefan Görtz. Hierarchical kriging model for variable-fidelity surrogate modeling. *AIAA Journal*, 50(9):1885–1896, 2012.

[37] Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

[38] Alexander IJ Forrester, András Sóbester, and Andy J Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the royal society a: mathematical, physical and engineering sciences*, 463(2088):3251–3269, 2007.

[39] Cédric Durantin, Justin Rouxel, Jean-Antoine Désidéri, and Alain Glière. Multifidelity surrogate modeling based on radial basis functions. *Structural and Multidisciplinary Optimization*, 56:1061–1075, 2017.

[40] Xueguan Song, Liye Lv, Wei Sun, and Jie Zhang. A radial basis function-based multi-fidelity surrogate model: exploring correlation between high-fidelity and low-fidelity models. *Structural and Multidisciplinary Optimization*, 60(3):965–981, 2019.

[41] Jonathan Tammer Eweis-Labolle, Nicholas Oune, and Ramin Bostanabad. Data fusion with latent map gaussian processes. *Journal of Mechanical Design*, 144(9): 091703, 2022.

[42] David JJ Toal. Applications of multi-fidelity multi-output kriging to engineering design optimization. *Structural and Multidisciplinary Optimization*, 66(6):125, 2023.

[43] Rodolphe Le Riche and Victor Picheny. Revisiting bayesian optimization in the light of the coco benchmark. *Structural and Multidisciplinary Optimization*, 64(5): 3063–3087, 2021.

[44] Jeong-Soo Park. Optimal Latin-hypercube designs for computer experiments. *Journal of statistical planning and inference*, 39(1):95–111, 1994.

[45] Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. *Statistical science*, 8 (1):10–15, 1993.

[46] Loic Le Gratiet. *Multi-fidelity Gaussian process regression for computer experiments.* PhD thesis, Université Paris-Diderot-Paris VII, 2013.

[47] Sambu Seo, Marko Wallat, Thore Graepel, and Klaus Obermayer. Gaussian process regression: Active data selection and test point rejection. In *Mustererkennung 2000: 22. DAGM-Symposium. Kiel, 13.–15. September 2000*, pages 27–34. Springer, 2000.

[48] Mickaël Binois, Jiangeng Huang, Robert B Gramacy, and Mike Ludkovski. Replication or exploration? Sequential design for stochastic simulation experiments. *Technometrics*, 61(1):7–23, 2019.

[49] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13: 455–492, 1998.

[50] Alexander Forrester and Donald Jones. Global optimization of deceptive functions with sparse sampling. In *12th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 5996, 2008.

[51] Deng Huang, Theodore T Allen, William I Notz, and R Allen Miller. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32:369–382, 2006.

[52] Xu He, Rui Tuo, and CF Jeff Wu. Optimization of multi-fidelity computer experiments via the eqie criterion. *Technometrics*, 59(1):58–68, 2017.

[53] Haitao Liu, Yew-Soon Ong, and Jianfei Cai. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural and Multidisciplinary Optimization*, 57(1):393–416, 2018.

[54] Dev Rajnarayan, Alex Haas, and Ilan Kroo. A multifidelity gradient-free optimization method and application to aerodynamic design. In *12th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 6020, 2008.

[55] Andrew March and Karen Willcox. Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives. *AIAA journal*, 50(5):1079–1089, 2012.

[56] Bo Liu, Slawomir Koziel, and Qingfu Zhang. A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems. *Journal of computational science*, 12:28–37, 2016.

[57] Yixin Liu, Shishi Chen, Fenggang Wang, and Fenfen Xiong. Sequential optimization using multi-level cokriging and extended expected improvement criterion. *Structural and Multidisciplinary Optimization*, 58(3):1155–1173, 2018.

[58] Haitao Liu, Yew-Soon Ong, Jianfei Cai, and Yi Wang. Cope with diverse data structures in multi-fidelity modeling: a gaussian process method. *Engineering Applications of Artificial Intelligence*, 67:211–225, 2018.

[59] Qi Zhou, Yuda Wu, Zhendong Guo, Jiexiang Hu, and Peng Jin. A generalized hierarchical co-kriging model for multi-fidelity data fusion. *Structural and Multidisciplinary Optimization*, 62:1885–1904, 2020.

[60] Yuda Wu, Jiexiang Hu, Qi Zhou, Shengyi Wang, and Peng Jin. An active learning multi-fidelity metamodeling method based on the bootstrap estimator. *Aerospace Science and Technology*, 106:106116, 2020.

[61] Xiongfeng Ruan, Ping Jiang, Qi Zhou, Jiexiang Hu, and Leshi Shu. Variable-fidelity probability of improvement method for efficient global optimization of expensive black-box problems. *Structural and Multidisciplinary Optimization*, 62(6):3021–3052, 2020.

[62] Huan Zhao, Zhenghong Gao, Fang Xu, and Lu Xia. Adaptive multi-fidelity sparse polynomial chaos-kriging metamodeling for global approximation of aerodynamic data. *Structural and Multidisciplinary Optimization*, pages 1–30, 2021.

[63] Shahrokh Shahpar, Chris Brooks, Alexander Forrester, and Andy Keane. Multi-fidelity design optimisation of a transonic compressor rotor. *9th European Conference on Turbomachinery: Fluid Dynamics and Thermodynamics, ETC 2011 - Conference Proceedings*, 2, 03 2011.

[64] Maolin Shi, Liye Lv, Wei Sun, and Xueguan Song. A multi-fidelity surrogate model based on support vector regression. *Structural and Multidisciplinary Optimization*, pages 1–13, 2020.

[65] Liye Lv, Chaoyong Zong, Chao Zhang, Xueguan Song, and Wei Sun. Multi-fidelity surrogate model based on canonical correlation analysis and least squares. *Journal of Mechanical Design*, 143(2):021705, 2021.

[66] Sander van Rijn, Sebastian Schmitt, Matthijs van Leeuwen, and Thomas Bäck. Finding efficient trade-offs in multi-fidelity response surface modelling. *Engineering Optimization*, pages 1–18, 2022.

[67] Yue Zhao, Jie Liu, and Zhelong He. A general multi-fidelity metamodeling framework for models with various output correlation. *Structural and Multidisciplinary Optimization*, 66(5):101, 2023.

[68] A Thenon, V Gervais, and M Le Ravalec. Sequential design strategy for kriging and cokriging-based machine learning in the context of reservoir history-matching. *Computational Geosciences*, 26(5):1101–1118, 2022.

[69] N. Andrés-Thió, M.A. Muñoz, and K. Smith-Miles. Bi-fidelity surrogate modelling: Showcasing the need for new test instances, 2021. available for download at https://github.com/INFORMSJoC/2021.0299.

[70] Huachao Dong, Baowei Song, Peng Wang, and Shuai Huang. Multi-fidelity information fusion based on prediction of kriging. *Structural and Multidisciplinary Optimization*, 51(6):1267–1280, 2015.

[71] Shifeng Xiong, Peter ZG Qian, and CF Jeff Wu. Sequential design and analysis of high-accuracy and low-accuracy computer codes. *Technometrics*, 55(1):37–46, 2013.

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

[73] Chanyoung Park, Raphael T Haftka, and Nam H Kim. Remarks on multi-fidelity surrogates. *Structural and Multidisciplinary Optimization*, 55(3):1029–1050, 2017.

[74] Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer, 1992.

[75] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 2020. doi: https://doi.org/10.1080/10556788.2020.1808977.

[76] John R Rice. The algorithm selection problem. In *Advances in computers*, volume 15, pages 65–118. Elsevier, 1976.

[77] Bernd Bischl, Pascal Kerschke, Lars Kotthoff, Marius Lindauer, Yuri Malitsky, Alexandre Fréchette, Holger Hoos, Frank Hutter, Kevin Leyton-Brown, Kevin Tierney, et al. Aslib: A benchmark library for algorithm selection. *Artificial Intelligence*, 237:41–58, 2016.

[78] Marius Lindauer, Jan N van Rijn, and Lars Kotthoff. The algorithm selection competitions 2015 and 2017. *Artificial Intelligence*, 272:86–100, 2019.

[79] Mustafa Mısır and Michèle Sebag. Alors: An algorithm recommender system. *Artificial Intelligence*, 244:291–314, 2017.

[80] Kate Smith-Miles and Mario Andrés Muñoz. Instance space analysis for algorithm testing: Methodology and software tools. *ACM Computing Surveys*, 55(12):1–31, 2023.

[81] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[82] Mario Andres Muñoz and Kate Smith-Miles. Instance space analysis: A toolkit for the assessment of algorithmic power, 2023. Source code is available at https://github.com/andremun/InstanceSpace.

[83] Kate Smith-Miles, Mario Andres Muñoz, and Neelofar Neelofar. Melbourne algorithm test instance library with data analytics (matilda), 2023. Source code is available at https://matilda.unimelb.edu.au/matilda/.

[84] Hossein Alipour, Mario Andrés Muñoz, and Kate Smith-Miles. Enhanced instance space analysis for the maximum flow problem. *European Journal of Operational Research*, 304(2):411–428, 2023.

[85] Nicolau Andrés-Thió. Bifidelity surrogate modelling methods, 2023. available for download at https://github.com/nandresthio/bifiEBBmethods.

[86] Pascal Kerschke and Heike Trautmann. Comprehensive feature-based landscape analysis of continuous and constrained optimization problems using the r-package flacco. In Nadja Bauer, Katja Ickstadt, Karsten Lübke, Gero Szepannek, Heike Trautmann, and Maurizio Vichi, editors, *Applications in Statistical Computing – From Music Data Analysis to Industrial Quality Improvement*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 93 – 123. Springer, 2019. doi: 10.1007/978-3-030-25147-5_7.

[87] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. Exploratory landscape analysis. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 829–836, 2011.

[88] Mario A Muñoz, Michael Kirley, and Saman K Halgamuge. Exploratory landscape analysis of continuous space optimization problems using information content. *IEEE transactions on evolutionary computation*, 19(1):74–87, 2014.

[89] Mike 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*, pages 386–395. Springer, 2012.

[90] Monte Lunacek and Darrell Whitley. The dispersion metric and the cma evolution strategy. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 477–484, 2006.

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

[92] Nicolau Andrés-Thió. Bifidelity surrogate modelling benchmark problems, 2023. available for download at https://github.com/nandresthio/bifiEBBbenchmarks.

[93] Nicolau Andrés-Thió. bifiebbbenchmarks data folder, Sep 2023. URL https://figshare.com/articles/dataset/bifiEBBbenchmarks_data_folder/24150747/1.

[94] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[95] Michael De La Maza and Deniz Yuret. Dynamic hill climbing. *AI expert*, 9(26):26, 1994.

[96] Sébastien Le Digabel. Solar, 2023. available for download at https://github.com/bbopt/solar.