NTNU
Det skapende universitet

Surrogatbasert optimering ved bruk av kunstige nevrale nettverk

Identifikasjon av lønnsomme O&M strategier for offshore vindfarmer gjennom stokastiske simuleringer

Marius Rise Gallala

Master i fysikk og matematikk
Innlevert: juni 2016
Hovedveileder: Jo Eidsvik, MATH
Medveileder: Iver Bakken Sperstad, SINTEF

Norges teknisk-naturvitenskapelige universitet
Institutt for matematiske fag

NTNU

Norwegian University of
Science and Technology

Surrogate-based optimisation using artificial neural networks

Identifying profitable O&M strategies for offshore wind farms through stochastic simulations

Marius Rise Gallala

Master of Science thesis: Industrial Mathematics

June, 2016

Supervisor 1: Jo Eidsvik, NTNU
Supervisor 2: Iver Bakken Sperstad, SINTEF

Preface

This master thesis constitutes the course TMA4905 Statistics for the Industrial Mathematics study program at NTNU. The focus of this thesis is to study how surrogate based optimisation can be used to identify favourable operation and maintenance strategies for offshore wind farms. The thesis builds upon the specialisation project TMA4500 Industriell Mathematics performed by the author the autumn of 2015. Parts of the project was presented as a poster at the EERA deepwind conference in Trondheim in January 2016.

The work related to the thesis was performed by the author during the spring of 2016. I would like to express my sincere gratitude to my supervisor Jo Eidsvik at NTNU, for inspiring discussions and continuous feedback. I am very grateful to my co-supervisor Iver Bakken Sperstad at Sintef Energy for introducing me to the offshore wind energy industry and for providing helpful suggestions.

Summary

Reducing the operation and maintenance (O&M) cost of offshore wind farms is essential in order to reduce the associated cost of energy. Simulation models enable us to evaluate the expected cost and the amount of electricity produced for different O&M strategies. Such strategies are characterised by decision variables such as the available vessel fleet, location of harbour, charting policies or others. The set of all possible strategies is a high dimensional space, and since the simulations are time consuming we can only explore small parts of it. We are restricted to find optimal solutions to sub problems, without knowing if these local solutions are close to the global optimal solution.

In this thesis, a surrogate model based on Artificial Neural Networks are used to more efficiently explore the input space. The overall goal is to find O&M strategies that are close to the global optimal solution. The model is fitted to the available input-output relations generated by the simulation model. If the surrogate model is an accurate representation of the simulation model, the choice of decision variables that optimizes the surrogate model should give good suggestions to O&M strategies. The next input for simulation is performed for strategies with high surrogate prediction and/or high related uncertainty. Such balance between exploitation and exploration is used to aid the search towards the global optimal solution.

The surrogate based optimisation approach is demonstrated on a relevant decision problem from the offshore wind industry. The identified strategies are most likely close to the (unknown) global optimal strategy. They can be used to gain knowledge of which strategies are favourable, and the surrogat model's predictions and corresponding uncertainties enables efficient comparison of different strategies.

Sammendrag

Kostnader tilknyttet drift- og vedlikeholdsoperasjoner utgjør en stor andel av kostnadene for offshore vindparker. Reduksjon av slike kostnader er essensielt for å gjøre energikilden mer konkurransedyktig. Strategier for hvordan drift- og vedlikeholdsoperasjoner utføres kan karakteriseres av beslutningvariabler som flåtesammensetning, lokasjon for havn, rutiner for chartering av fartøy eller andre. Simuleringsmodeller kan benyttes for å evaluere kostnad og mengde produsert energi for ulike strategier. Rommet av mulige strategier er høydimensjonalt, og da hver enkelt simulering er relativt tidkrevende, er det ofte bare mulig å utforske en liten del av det. Det er derfor vanskelig å vite om man har identifisert den strategien som gir mest gunstig simuleringsresultat, eller om det finnes andre som er vesentlig bedre

I denne oppgavene brukes en approksimasjon av simuleringsmodellen, kalt surrogatmodell, for å mer effektivt utforske løsningsrommet. Det overordnede målet er å finne strategier som maksimerer simuleringsmodellen. Surrogatmodellen benytter kunstige nevrale nettverk og tidligere simuleringer for å beskrive sammenhengen mellom beslutningvariablene og simuleringsresultat. Hvis surrogatmodellen representerer denne relasjon presist, vil strategiene som maksimerer surrogatmodellen representere gode strategier. Den neste simuleringen utføres for strategier der surrogatmodellen predikerer gunstig simuleringsresultat eller høy grad av usikkerhet. Slik balanse mellom utnyttelse og utforskning blir brukt til å styre søket etter den globalt beste strategien.

Den surrogatbaserte fremgangsmåten for optimering anvendes på et relevant beslutningsproblem fra offshore vindkraft industrien. Strategiene som identifiseres er trolig omtrentlig like gode som den (ukjente) optimale strategien. Disse kan gi kunnskap om hvilke strategier som er gunstige, og surrogatmodellens prediksjoner med tilhørende usikkerhet kan benyttes for effektiv sammenligning.

Table of Contents

Preface i

Summary iii

Sammendrag v

1 Introduction 1

2 Artificial neural networks 5
- 2.1 Underlying model 6
- 2.2 Backpropagation algorithm 9
  - 2.2.1 Gradient descent 10
  - 2.2.2 Levenberg-Marquardt 11
- 2.3 Regularisation techniques 12
  - 2.3.1 Early stopping 12
  - 2.3.2 Regularisation 14
  - 2.3.3 Bayesian regularisation 16
  - 2.3.4 Assessment of the effect of the number of vertices 17
- 2.4 Bagging for stabilization and quantification of uncertainty 19
  - 2.4.1 Bagging 21
  - 2.4.2 Uncertainty estimation 22

3 Optimisation by sequential design 27
- 3.1 Introduction 27
  - 3.1.1 Notational remarks 28
- 3.2 Statistical distribution of $f(\mathbf{x})$ 28
- 3.3 Infill functions 31
  - 3.3.1 Probability of improvement 31
  - 3.3.2 Expected Improvement 32
  - 3.3.3 Upper confidence bound 35
  - 3.3.4 Generalised Expected Improvement with constraints 35

3.3.5 Sampling paths for synthetic data 37
3.4 Several input points 40
3.4.1 Joint maximisation 40
3.4.2 Pool of infill functions 46
3.4.3 Sample paths for synthetic data 48
3.5 Constrained optimisation 53
3.5.1 Constrained expected improvement 53
3.5.2 Penalty method 54

4 Case study on NOWIcob simulation model 55
4.1 Introduction 55
4.1.1 Maximal theoretical income 60
4.1.2 Annual service constraint 61
4.2 Vessel fleet optimisation-problem 63
4.3 Optimisation algorithm for the VFO problem 64

5 Experimental results 67
5.1 The optimisation process 67
5.2 Optimal strategies 70
5.2.1 Comparison of fleet mix 77
5.3 Limitations 77

6 Conclusion 81
6.1 Possible improvements 81

Bibliography 83

Appendix 89
A Input data and model assumptions 89

Chapter 1

Introduction

Many problems and physical phenomena are difficult to analyse by physical experimentation. The experiments are either time consuming, costly, prohibited or restricted in other ways. Facing such problems there are many advantages of utilising a simulation model, also called computer code, see for example Banks (1998). In this project, the problem of finding cost-effective operation and maintenance (O&M) strategies for offshore wind farms is studied. The cost of energy for wind farms is often higher than alternative energy sources. In order to decrease the cost of energy for a wind farm, the produced amount of energy has to increase or the total life cycle cost must decrease. The O&M cost constitutes about one third (Shafiee, 2015) of the overall cost, so reducing this cost is essential in order to reduce the cost of energy.

Finding good O&M strategies is a complex problem: the strategies involve many decisions that interact and develop over time. In addition, there are inherent uncertainties for several aspects that affect the O&M strategies. Typically decision problems may be

- location of the wind farm
- fleet mix
- number of technicians stationed at a nearby harbour

From the mentioned examples it is obvious that the different decision problems are interrelated, e.g. a wind farm with rough weather may favor robust vessels, while O&M tasks on a wind farm with calm weather may favour vessels with high speed. The optimal solution to each sub decision problem could be found by introducing some criteria of optimality, e.g. choose

- a location with calm climate, preferably nearby shore
- a fleet mix that ensures high availability, without exceeding a given budget
- a work force that ensures good utilisation of the fleet

The optimality criteria may differ for each sub problem and for each stakeholder. In this project, the focus is on the long-term goals of minimising the total O&M cost and maximising the produced amount of electricity.

Failures of different types tend to occur for the turbines at a wind farm. Such failures may cause the turbines to stop, and some may require technicians assessing the turbine. The fixed and variable cost for vessels that can transport personnel, or are used otherwise to perform the maintenance tasks, represents one of the key challenges of reducing the O&M cost.

There have been developed several simulation models for offshore wind farms. An extensive review and comparison of some of them can be found in Hofmann (2011) and Dinwoodie et al. (2015) respectively. In this project, the NOWIcob (**N**orwegian **o**ffshore **wi**nd power life cycle **c**ost and benefit model) simulation model is used. Each simulation is characterised by a set of input parameters and variables that fully describe the wind farm and all aspects of the O&M strategies.

Knowledge of the relation between input variables and output can be obtained by performing simulations for different inputs. However, these simulations can be time-consuming as well as having a high number of input variables. It is often possible to explore only a subset of all possible configurations of input variables. In many cases, it is of interest to identify inputs that (approximately) optimise the output without excessive evaluations of the simulation model. This can be achieved by surrogate based optimisation. Such methodology utilise a simplified model of the simulation model, called surrogate model, in the search for the global optimiser. The surrogate model can be used to predict the output for different input. Based on the predictions, the next point(s) for simulation is selected, and the information from these simulation are used to update the surrogate model, and the process is repeated.

In this project, we study how a surrogate model based on artificial neural networks (ANNs) can be used to optimise the output of a simulation model. The rationale for using ANNs are their universal predictive power, they can be applied for different input and output types and the seamless transition when adding or removing input or output variables. The latter ability is practical when considering O&M strategies, since there are many possible decision problems that can be studied. Thus it is of interest that the surrogate model can be used across a range of sub problems, instead of being specialised to solve a specific type of problems.

Efficient surrogate model optimisation is sensitive to the balance between exploration and exploitation when selecting the next input for simulation. In this project, common criteria as probability of improvement, expected improvement and the upper confidence bound are studied. The simulation model under consideration is independent of other simulations. Thus, in order to utilise parallel processing, we propose methods for selecting several points between each update of the surrogate model. These points can be evaluated simultaneously, and therefore reduce the overall time consumption used to identify (approximate) optimisers of the simulation model.

An outline of the report is given in the following. In Chapter 2 ANNs and their usability as surrogate model is studied. Concepts related to fitting ANNs are discussed in sections 2.2 and 2.3. In Chapter 3 sequential methods for optimisation of the simulation model is studied. These methods utilise the surrogate model and an infill function to select the next point(s) for simulation. Common infill functions requires a measure of the uncertainty of the surrogate model, which is quantified by using aggregated bootstraping (Bagging) introduced in Section 2.4. In section 3.4 methods for selecting several input points simultaneously is proposed and demonstrated on a synthetic example. In Section 4.1 the NOWIcob simulation model is introduced, and a decision problem related to O&M strategies is introduced in Section 4.2. The surrogate based optimisation method is tested on three instances of this problem. The performance of the method and the identified optimal strategies is assessed in Section 5. In Chapter 6 discussion of the surrogate based optimisation method's performance and suggestions for further work is presented.

Chapter 2

Artificial neural networks

This chapter will introduce surrogate models based on artificial neural networks (ANNs) and related tools for fitting and validating these models. ANNs is a large class of models that are inspired by biological neural networks. ANNs have been used within many application for classification, forecasting or approximation of an unknown function. ANNs used for image recognition, speech recognition, or other complex problems, may have hundreds of millions of adjustable model parameters (LeCun et al., 2015). In this project, the class of ANNs considered is limited to two layer feed forward neural networks (FFNN). This architecture is among the simplest and most widely used, and such networks may fit any continuous function arbitrarily well. See for example Cybenko (1989) for a proof.

ANNs have been utilised within a variety of applications, for example to forecast water resource variables (flow, rainfall, pH, and more) for a variety of locations (Maier and Dandy, 2000) and to predict the future behaviour (position, speed and course) of naval vessels (Zissis et al., 2015). They have also been applied for recognition of handwritten digits (Knerr et al., 1992) and steering autonomous vehicles (Pomerleau, 1989). Other examples can be found in Hadsell et al. (2009); Mirowski et al. (2009); Bekirev et al. (2015)

In section 2.1 a graphical representation of ANNs is presented, and the mathematical relation between input $\mathbf{x}$ and output $\mathbf{f}(\mathbf{x};\boldsymbol{\theta})$ for a two layer FFNN is explained. The back-propagation (BP) algorithm for updating the model parameters $\boldsymbol{\theta}$ is derived in section 2.2. The process of determining the model parameters is referred to as training or fitting an ANN.

Models based on the Gaussian Process (GP) are commonly used in surrogate based optimisation, see for example Storlie et al. (2013); Gramacy and Lee (2009). The GP is analytical tractable and gives a measure of prediction uncertainty, which may explain its widespread use. The rationale for using ANNs is the seamless transition when adding or removing input and output variables of different types. GP models with several output variables may require cumbersome implementation and specification of covariance structures, see for example Álvarez and Lawrence (2011); Osborne et al. (2008), whereas ANNs can be fitted in a more automatic manner with well-studied procedures as the BP algorithm. For a thorough survey of other alternatives for surrogate models we refer to Shan and Wang (2010).

The complexity and flexibility of an ANN depends on the network architecture. Since the underlying functional relation $f^{true}(\mathbf{x})$ is unknown, the architecture should be flexible enough to model a broad class of functions. However, a very flexible network may be prone to overfitting. In Section 2.3, regularisation techniques to address this problem are studied. The effect of such techniques and the choice of network architecture are studied on synthetic data in section 2.3.4.

A surrogate model $\mathbf{f}(\mathbf{x})$ based on ANNs can be used to approximate $\mathbf{f}^{true}(\mathbf{x})$. It is in many cases interesting to assess the uncertainty of the predictions $\mathbf{f}(\mathbf{x})$. A method called *bagging*, discussed in section 2.4, can be utilized to form more stable predictions $\mathbf{f}(\mathbf{x})$ and quantify the model uncertainty. The uncertainty measure is useful when selecting the next point(s) for simulation, and is therefore an essential aspect of the adaptive optimisation process proposed in Chapter 3.

2.1 Underlying model

In this section, basic graph theory is used to formalise the graphical representation of ANNs. The corresponding model equations are derived for the two layer FFNN. Notation that are used frequently is defined in the following. Let

$$\mathbf{D} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n = \{(\mathbf{x}^i, \mathbf{f}^{true}(\mathbf{x}^i))\}_{i=1}^n \tag{2.1}$$

denote a set of n input-output realisations of the underlying function $\mathbf{f}^{true}(\mathbf{x})$. It is assumed that the input and output have P and K components, i.e. $\mathbf{x}^i = (x_1, \dots, x_P)^T$ and $\mathbf{y}^i = (y_1, \dots, y_K)^T$. Let $\mathbf{f}(\mathbf{x}; \boldsymbol{\theta}) = (f_1(\mathbf{x}; \boldsymbol{\theta}), \dots, f_K(\mathbf{x}; \boldsymbol{\theta}))$ denote the surrogate model prediction for the F model parameters $\boldsymbol{\theta} = (\theta_1, \dots, \theta_F)$. The shorthand notation $\mathbf{f}(\mathbf{x})$ is used as well.

An ANN can be represented as a weighted directed graph with vertices and edges. The vertices are connected by edges with weights that represent the strength of the connections. The vertices are grouped into three types: input, computing or output vertices. The input vertices represent the input $\mathbf{x}$, and the information is passed and processed by the edges to the computing vertices, until it reaches the output vertices $\mathbf{f}(\mathbf{x}; \boldsymbol{\theta})$.

Definition 2.1.1. (Architecture) Networks with the same architecture have the same directed graph and vertex functions but possibly with different weights.

The direction of the edges describes the way the information is passed. The input vertices has no incoming edges. The output layer represents the output $\mathbf{f}(\mathbf{x}; \boldsymbol{\theta})$ obtained by presenting the network with an input $\mathbf{x}$.

Definition 2.1.2. (Feed forward neural network) The architecture of a feed forward neural network is defined by a directed acyclic graph and a choice of vertex functions.

For a directed acyclic graph, the computing vertices V may be grouped in different layers $L_0, \dots, L_{H+1}$. Layer L_0 and L_{H+1} contains the input and output vertices respectively. The layers are arranged such that the vertices contained in layer L_i only have incoming

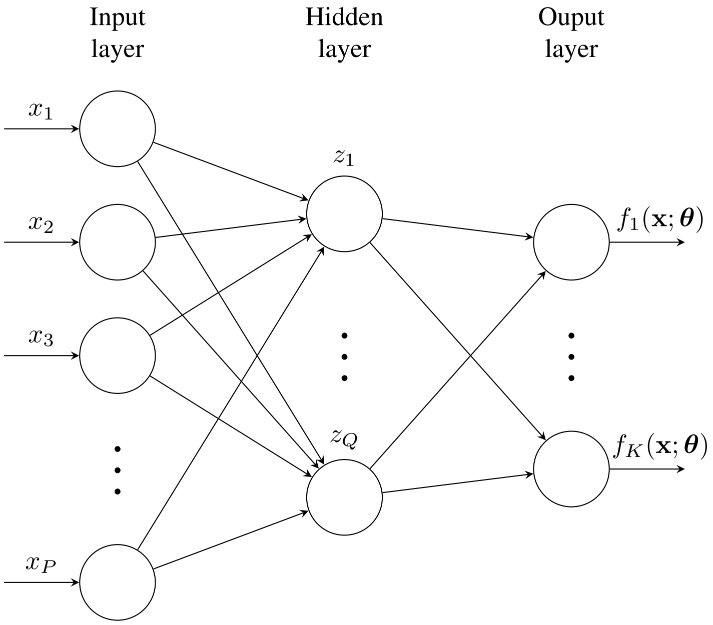

Input layer
Hidden layer
Ouput layer
x_1
x_2
x_3
x_P
z_1
z_Q
$f_1(\mathbf{x};\boldsymbol{\theta})$
$f_K(\mathbf{x};\boldsymbol{\theta})$

Figure 2.1: Illustration of a two layer feedforward network.

edges from layers L_j where $j < i$. The layers $\{L_i\}; i = 1, \ldots, H$ are commonly referred to as *hidden layers* since the values of the vertices in these layers are of no direct interest. They are only used in the process of computing the output $\mathbf{f}\,(\mathbf{x};\boldsymbol{\theta})$.

A special case of the FFNN is the multilayer perceptron (MLP). For such networks, the edges to the vertices in L_i come only from vertices in layer L_{i-1}. The proceeding results and discussion is related to MLP with one hidden layer. Some of the result may be extended to more complex networks. See for example Fine et al. (1999) for a more detailed discussion of ANNs and different architectures. Figure 2.1 shows a graphical representation of the vertices and edges for the architecture under consideration.

The vertices $z_1, \ldots, z_Q$ in the hidden layer are commonly referred to as derived features of the input $\mathbf{x}$. A linear combination of the input is passed to each of the Q derived features. At these computing vertices, the information from the inward edges is transformed by function $\sigma(\cdot)$ called *activation* function. Thus, the q'th derived feature may be written as

$$z_q = \sigma(\alpha_{0,q} + \boldsymbol{\alpha}_q^T \mathbf{x}), \qquad q = 1, \ldots, Q, \tag{2.2}$$

where the scalars $\alpha_{0,1}, \ldots, \alpha_{0,Q}$ and the vectors $\boldsymbol{\alpha}_q^T = (\alpha_{q,1}, \ldots, \alpha_{q,P})$ represent bias and weight terms for the q'th linear transformation. These weights and biases are contained in the set of model parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_F)$. The function $\sigma(\cdot)$ represents a possible non-linear transformation of the linear combinations of the input.

Definition 2.1.3. (Sigmoid functions) A sigmoid function is a bounded differentiable real

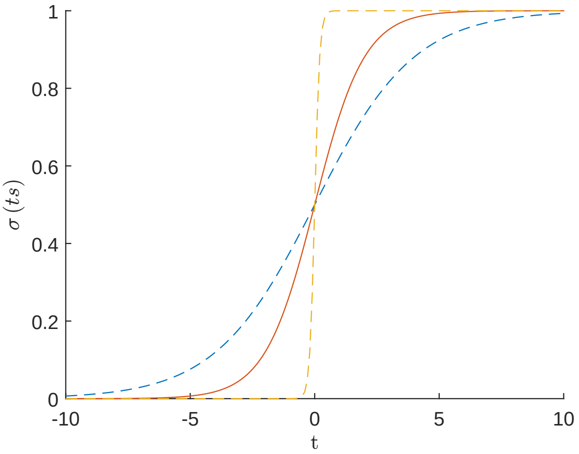


Figure 2.2: The sigmoid function $\sigma(st)$ as a function of t for $s = 1$ (red), $s = 0.5$ (dotted blue) and $s = 10$ (dotted yellow). For the latter value of s the sigmoid function is approximately the hard limit function. The above figure is adopted from a similar illustration in Hastie et al. (2009).

function that is defined for all real input values and that has a positive derivative everywhere (Han and Moraga, 1995).

Sigmoid functions are commonly used as activation function in neural networks. These functions may be recognised by their ”S-shaped” form. Examples of sigmoidal functions and their properties may be found in Menon et al. (1996). A particularly popular sigmoidal function is defined as

$$\sigma\,(t) = \frac{1}{1+e^{-t}}, \tag{2.3}$$

and is displayed graphically in Figure 2.3. For very small arguments, the function $\sigma()$ is approximately linear, and for large input it approaches the hard limit function.

The derived features $\mathbf{z} = (z_1, \ldots, z_Q)^T$, represented by the vertices in the hidden layer, are connected to the output layer with edges. The k'th output can be expressed as

$$f_k(\mathbf{x};\boldsymbol{\theta}) = g_k(\beta_{0,k} + \boldsymbol{\beta}_k^T\mathbf{z}), \qquad\qquad k = 1, \ldots, K, \tag{2.4}$$

where the scalars $\beta_{0,1}, \ldots, \beta_{0,K}$ and the vectors $\boldsymbol{\beta}_k = (\beta_{k,1}, \ldots, \beta_{k,Q})^T$ represents bias terms and linear transformations of the derived features $\mathbf{z}$. These parameters are part of the set of model parameters $\boldsymbol{\theta}$. The k'th output of the network can be expressed explicitly in terms of the input $\mathbf{x}$, the functions $\sigma()$, $g_1(), \ldots, g_K()$ and the model parameters $\boldsymbol{\theta}$.

In this section, it was assumed that the model parameters $\boldsymbol{\theta}$ were known. However, these parameters are seldom known a priori. In the proceeding section, a popular and efficient procedure for updating the weights $\boldsymbol{\theta}$ is derived for the particular network considered in this section.

Algorithm 1 Backpropagation algorithm

```
Obtain $\mathbf{D}^n = \{(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^n, \mathbf{y}^n)\}$
Initialise $\boldsymbol{\theta}^0$
Iteration $r = 0$
repeat
    Compute output $\mathbf{f}(\mathbf{x^i}; \boldsymbol{\theta}^r)$ for $i = 1, \ldots, N$
    Compute loss (error) $L(\boldsymbol{\theta}^r)$
    Compute gain $\Delta_{\boldsymbol{\theta}}$
    Update parameters $\boldsymbol{\theta}^{r+1} = \boldsymbol{\theta}^r + \Delta_{\boldsymbol{\theta}}$
    $r = r + 1$
until Stopping criterion met
return $\boldsymbol{\theta}^{r+1}$                              ▷ Model parameters
```

2.2 Backpropagation algorithm

The backpropagation algorithm (BP) is widely used to fit ANNs. In each iteration, the algorithm uses a set of input and output relations $\mathbf{D} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$ to update the model parameters $\boldsymbol{\theta}$. The updates depend on the choice of an error function and which numerical optimisation method is used to minimise this function. In the following, a description of central steps of the algorithm is provided. In section 2.2.1 and 2.2.2, two alternative numerical optimisation methods are described.

Let $L(\mathbf{D}; \boldsymbol{\theta})$ denote an loss (error) function that measures the correspondence between the predictions and the observations in $\mathbf{D}$ for a set of model parameters $\boldsymbol{\theta}$. The predictions $f_1(\mathbf{x}^i; \boldsymbol{\theta}), \ldots, f_K(\mathbf{x}^i; \boldsymbol{\theta})$ are obtained by propagating the input vector $\mathbf{x}^i$ through the network. The error is calculated by comparing the predictions with the observed output $y_{i,1}, \ldots, y_{i,K}$. A common choice is to use the squared distance between the predictions and the observations as a measure of model fit. For a specific data sample pair $(\mathbf{x}^i, \mathbf{y}^i)$ the i'th squared loss is defined as

$$L_i(\mathbf{D}; \boldsymbol{\theta}) = \sum_{k=1}^{K} \left(f_k(\mathbf{x}^i; \boldsymbol{\theta}) - y_{i,k}\right)^2 = l_i^2. \tag{2.5}$$

Define the total error as the sum of the error for each data sample in $\mathbf{D}$, i.e.

$$L(\mathbf{D}; \boldsymbol{\theta}) = \sum_{i=1}^{n} L_i(\mathbf{D}; \boldsymbol{\theta}). \tag{2.6}$$

For convenience, introduce the notation $\mathbf{l} = (l_1, \ldots, l_n)^T$. With this notation $L(\mathbf{D}; \boldsymbol{\theta})$ can be written compactly as

$$L(\mathbf{D}; \boldsymbol{\theta}) = \mathbf{l}^T \mathbf{l}. \tag{2.7}$$

The BP procedure is summarised in Algorithm 1 . In each iteration, the model parameters are updated based on the loss function $L(\mathbf{D}; \boldsymbol{\theta}^r)$ and the gain

$$\Delta_{\boldsymbol{\theta}} = \boldsymbol{\theta}^{r+1} - \boldsymbol{\theta}^r. \tag{2.8}$$

The gain is determined by using a numerical optimisation technique for minimisation of the error function. The process of presenting the network with data samples, computing the errors and gain, and adding the gain to the current model parameters are repeated iteratively. For later use, we introduce the gradient $\nabla L(\mathbf{D};\boldsymbol{\theta})$, where the j'th element is defined as

$$\nabla L(\mathbf{D};\boldsymbol{\theta})_j = \frac{\partial L(\mathbf{D};\boldsymbol{\theta})}{\partial \theta_j}. \tag{2.9}$$

Using the fact that $L(\mathbf{D};\boldsymbol{\theta})$ is a sum of squares, the j'th element of the gradient may be written as

$$\frac{\partial L(\mathbf{D};\boldsymbol{\theta})}{\partial \theta_j} = \sum_{i=1}^{n} \frac{\partial l_i^2}{\partial \theta_j} = 2\sum_{i=1}^{n} l_i \frac{\partial l_i}{\partial \theta_j}, \tag{2.10}$$

and the gradient can be expressed on matrix form as

$$\nabla L(\mathbf{D};\boldsymbol{\theta}) = 2J^T\mathbf{l}, \tag{2.11}$$

where J is the Jacobian matrix with elements (i,j) defined as

$$J_{i,j}(\boldsymbol{\theta}) = \frac{\partial l_i}{\partial \theta_j}. \tag{2.12}$$

Different numerical numerical optimisation techniques can be used to determine the gain $\Delta_{\boldsymbol{\theta}} = \boldsymbol{\theta}^{r+1} - \boldsymbol{\theta}^r$ in order to minimise the loss function $L(\mathbf{D};\boldsymbol{\theta})$. Two of the most widely studied methods are introduced in the proceeding sections 2.2.1 and 2.2.2.

2.2.1 Gradient descent

By using a gradient descent method, the $(r+1)$ update may be written on the form

$$\boldsymbol{\theta}^{r+1} = \boldsymbol{\theta}^r - \gamma^r \nabla L(\mathbf{D};\boldsymbol{\theta}^r), \tag{2.13}$$

where γ^r is the step size in the negative direction of the gradient $\nabla L(\mathbf{D};\boldsymbol{\theta})$. The rightmost term in the above equation represent the gain $\Delta_{\boldsymbol{\theta}}$. By inserting for the gradient, from Equation (2.11), the $(r+1)$ iteration is given by

$$\boldsymbol{\theta}^{r+1} = \boldsymbol{\theta}^r - \gamma^r 2J(\boldsymbol{\theta}^r)^T \mathbf{l}. \tag{2.14}$$

The $(r+1)$ iteration can be written on elementwise form, by using the definition of $\nabla l(\boldsymbol{\theta}^r)$ in Equation (2.9), i.e.

$$\theta_j^{r+1} = \theta_j^r - 2\gamma^r \sum_{i=1}^{n} l_i \frac{\partial l_i}{\partial \theta_j}. \tag{2.15}$$

From Equation (2.15), it is clear that the j'th update contains the partial derivatives of $l_1, \ldots, l_n$ with respect to θ_j. Recall from Equation (2.5) that

$$l_i = \sum_{k=1}^{K} f_k\left(\mathbf{x}^i;\boldsymbol{\theta}\right) - y_{i,k}, \tag{2.16}$$

where the network output $f_k\left(\mathbf{x}^i;\boldsymbol{\theta}\right)$ for a MLP network with one hidden layer is given by Equation (2.4). The partial derivatives of l_i with respect to the model parameters

$$\begin{aligned}\boldsymbol{\theta} &= \{\theta_1,\dots,\theta_F\} && (2.17)\\ &= \{\alpha_{0,1},\dots,\alpha_{0,Q},\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_Q,\beta_{0,1},\dots,\beta_{0,K},\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_K\}, && (2.18)\end{aligned}$$

are derived in Hastie et al. (2009), and can can be inserted into Equation (2.15) in order to obtain an explicit expression for the $(r+1)$ update of the j' model parameter θ_j. The focus of this thesis is more on the utilisation of ANNs. Different fitting procedures are implemented in many software tools, hence the details of such procedures are omitted in order to reduce the scope of the project.

Recall that the update equations were derived by calculating the derivatives of the sum of the errors for each data sample in the data set $\mathbf{D}$. This is often referred to *batch* learning since a batch of data samples is considered between each update. An alternative is to use only one sample pair $(\mathbf{x}^i,\mathbf{y}^i)$ to estimate the sum of errors in Equation (2.6). Thus, the computational requirements of computing the gradient descent update for $\boldsymbol{\theta}$ in Equation (2.13) may be reduced significantly if n is large. The resulting update formula is often called *stochastic gradient descent* and may result in better estimates of $\boldsymbol{\theta}$. We refer to LeCun et al. (2012) for a more in depth study of the advantages of *batch* and *online* learning.

The derived update equations are based on a gradient (or steepest) descent method for a non-linear least squares function $L\left(\mathbf{D};\boldsymbol{\theta}\right)$. In the proceeding, a method that often outperforms the steepest descent method, in terms of convergence speed, is discussed.

2.2.2 Levenberg-Marquardt

The Levenberg-Marquardt (LM) method combines the Gauss-Newton method and a damping factor. The Gauss-Newton method uses an approximation of the Hessian to approximate the Newton method. The two methods are described in the following, the derivations are based on more detailed work that can be found in (Hagan et al., 2003, ch. 12). The $r+1$'th iteration of Newtons method of minimising the loss (error) $L\left(\mathbf{D};\boldsymbol{\theta}\right)$, with respect to $\boldsymbol{\theta}$, has the form

$$\boldsymbol{\theta}^{r+1} = \boldsymbol{\theta}^{r} + \left[\nabla^2 L\left(\mathbf{D};\boldsymbol{\theta}\right)\right]^{-1}\nabla L\left(\mathbf{D};\boldsymbol{\theta}\right), \tag{2.19}$$

where $\nabla^2 L\left(\mathbf{D};\boldsymbol{\theta}\right)$ and $\nabla L\left(\mathbf{D};\boldsymbol{\theta}\right)$ denotes the Hessian matrix and the gradient of $L\left(\mathbf{D};\boldsymbol{\theta}\right)$. The (i,j)'th element of the Hessian matrix is defined as

$$\nabla^2 L\left(\mathbf{D};\boldsymbol{\theta}\right)_{i,j} = \frac{\partial^2 L\left(\mathbf{D};\boldsymbol{\theta}\right)}{\partial\theta_i\partial\theta_j}. \tag{2.20}$$

It can be shown that the Hessian matrix of the sums of squared errors function $L\left(\mathbf{D};\boldsymbol{\theta}\right)$ is

$$\nabla^2 L\left(\mathbf{D};\boldsymbol{\theta}\right) = 2J\left(\boldsymbol{\theta}\right)^T J\left(\boldsymbol{\theta}\right) + 2\sum_{i=1}^{n} L_i\left(\mathbf{D};\boldsymbol{\theta}\right)\nabla^2 L_i\left(\mathbf{D};\boldsymbol{\theta}\right). \tag{2.21}$$

The Hessian matrix in Equation (2.21) involves second derivatives. The GN method avoids the computation of these derivatives by assuming that the term $\sum_{i=1}^{n} L_i\left(\mathbf{D};\boldsymbol{\theta}\right)\nabla^2 L_i\left(\mathbf{D};\boldsymbol{\theta}\right)$

is small, i.e.

$$\nabla^2 L(\mathbf{D};\boldsymbol{\theta}) \approx 2J(\boldsymbol{\theta})^T J(\boldsymbol{\theta}). \tag{2.22}$$

By inserting the gradient, Equation (2.11), and the approximated Hessian, Equation (2.22), into the Newton iteration defined in Equation (2.19), the $(r+1)$'th iteration of the Gauss-Newton method is

$$\begin{aligned}\boldsymbol{\theta}^{r+1} &= \boldsymbol{\theta}^r + \left[\nabla^2 L(\boldsymbol{\theta}^r)\right]^{-1} \nabla L(\boldsymbol{\theta}^r) && (2.23)\\ &= \boldsymbol{\theta}^r + \left[J(\boldsymbol{\theta}^r)^T J(\boldsymbol{\theta}^r)\right]^{-1} J(\boldsymbol{\theta}^r)^T \mathbf{l}(\boldsymbol{\theta}^r). && (2.24)\end{aligned}$$

The matrix $J(\boldsymbol{\theta})^T J(\boldsymbol{\theta})$ may not be invertible. The LM method avoids this problem by adding a term $\lambda^r \mathbf{I}$ to the approximated Hessian matrix. The parameter λ^r is commonly referred to as a damping factor. Augmenting the approximated Hessian matrix in Equation (2.23), the $(r+1)$'th iteration of the LM method takes the form

$$\boldsymbol{\theta}^{r+1} = \boldsymbol{\theta}^r + \left[J(\boldsymbol{\theta}^r)^T J(\boldsymbol{\theta}^r) + \lambda^r\right]^{-1} J(\boldsymbol{\theta}^r)^T \mathbf{l}(\boldsymbol{\theta}^r). \tag{2.25}$$

In this section, equations for computing the gain for the model parameters $\boldsymbol{\theta}^{r+1}$ were derived for two different numerical optimisation methods. In the proceeding section, techniques that may improve the generalisation ability of the network are presented.

2.3 Regularisation techniques

The number of model parameters $\boldsymbol{\theta}$ is often large compared with the number of data samples. Therefore, fitting an ANN by minimization of the loss $L(\mathbf{D};\boldsymbol{\theta})$ may lead to a complex network which describes the noise in the dataset instead of the underlying function $f^{true}(\mathbf{x})$. Two techniques commonly referred to as *early stopping* and *regularisation* to address this problem is described in the following sections 2.3.1 and 2.3.2. These techniques are illustrated by fitting ANNs, with and without early stopping and different choices of regularisation parameters, to synthetic data. In Section 2.3.3 the *Bayesian regularisation* approach which enables objective determination of regularisation parameters is introduced.

In order to ease the understanding, the different concepts are illustrated on a running example. Let $f^{true}(x)$ be defined as

$$f^{true}(x) = \begin{cases} \sin\left(\frac{\pi x}{5}\right) + \frac{1}{5}\cos\left(\frac{4\pi x}{5}\right), & \text{if } 0 \le x \le 10 \\ \frac{x}{10} - 0.8, & \text{if } 10 \le x \le 15, \end{cases} \tag{2.26}$$

and let $\mathbf{D}^n = \{(x^i, y^i)\}_{i=1}^n$ denote a dataset of n samples where y_i is a noisy observation of $f^{true}(x_i)$, as defined in Table 2.1. The function $f^{true}(x_i)$ is shown in Figure 2.3 .

2.3.1 Early stopping

Let $\mathbf{D}^{tr}$ and $\mathbf{D}^{va}$ denote a partitioning of the original data samples $\mathbf{D}$, with n^{tr} and n^{va} unique data sample pairs respectively. The sets $\mathbf{D}^{tr}$ and $\mathbf{D}^{va}$, often called training and

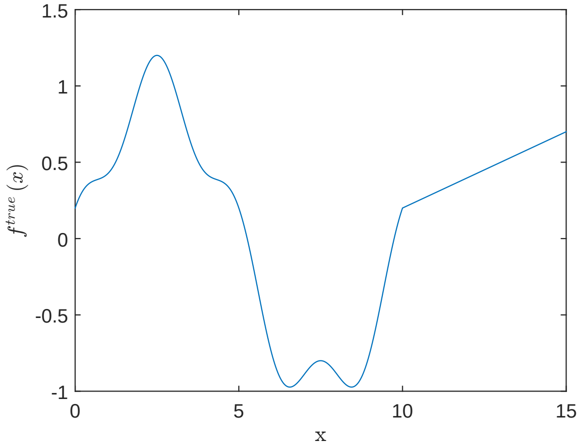


Figure 2.3: The function $f^{true}(x)$ for the running example. A similar function was used by Higdon (2002) and augmented in Gramacy and Lee (2009) with the linear region $[10, 15]$. This is used as a running example.

Table 2.1: Definition of $\mathbf{D}^n$, $\mathbf{D}_0^n$ and $\mathbf{D}^{grid,n}$ for denoting date samples $\{(x^1, y^1), \ldots, (x^b, y^n)\}$ with different procedure of generating design points $x^1, \ldots, x^n$ and observations and $y^1, \ldots, y^n$. The data samples $\mathbf{D}^n$ and $\mathbf{D}_0^n$ represent n observations of $f^{true}(x^i)$ with and without noise respectively, where x^i is drawn uniformly from $\Omega_x = [0, 15]$. The data samples $\mathbf{D}^{grid,n}$ denotes noise-free observations of $f^{true}(x^i)$ where $x^1, \ldots, x^n$ are defined as a regular grid on Ω_x. The set $\mathbf{D}^{grid,n}$ is throughout this chapter used as a test set to measure the prediction accuracy of different ANNs.

Data samples	Design points	Observations
$\mathbf{D}^n$	$x^i \sim \texttt{Unif}(\Omega_x)$	$y^i = f^{true}\left(x^i\right) + Z^i \sim \texttt{Norm}\left(0, 0.1^2\right)$
$\mathbf{D}_0^n$	$x^i \sim \texttt{Unif}(\Omega_x)$	$y^i = f^{true}\left(x^i\right)$
$\mathbf{D}^{grid,n}$	$x^i = 0.1(i-1)$	$y^i = f^{true}\left(x^i\right)$

Table 2.2: Explanation of training, validation and test set. The pairs of data samples $\mathbf{D} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$ are divided randomly into $\mathbf{D}^{tr}$, $\mathbf{D}^{va}$ and $\mathbf{D}^{te}$.

Dataset	Number of data sample pairs	Purpose
$\mathbf{D}^{tr}$	n^{tr}	The training set $\mathbf{D}^{tr}$ is repeatedly used to update the model parameters $\boldsymbol{\theta}$
$\mathbf{D}^{va}$	n^{va}	The validation set $\mathbf{D}^{va}$ is used to determine at which iteration r the model $\mathbf{f}(\mathbf{x}; \boldsymbol{\theta}^r)$ is adequate
$\mathbf{D}^{te}$	n^{te}	The test set $\mathbf{D}^{te}$ is used to assess the performance of the fitted model on unseen data.

validation sets, can be utilised to fit and determine at which iteration the fitting process should be terminated. In some cases, the data samples $\mathbf{D}$ are partitioned in an additional test set $\mathbf{D}^{te}$ that can be used to evaluate the performance of the fitted network. This partitioning of $\mathbf{D}$ and the role of each set is summarised in Table 2.2. The training set is used to update the model parameters $\boldsymbol{\theta}$. The error on the validation set is computed for each set of model parameters $\boldsymbol{\theta}^r$. Typically, both the training and validation error decreases in the first few updates of the model parameters. However, at some point the validation error may increase. This indicates that the updated parameters $\boldsymbol{\theta}^{r+1}$ begins to mimic the training set rather than the underlying function. Thus, by recording the validation error and weights in all iterations, a network with better generalisation abilities can be obtained by selecting the set of model parameters $\boldsymbol{\theta}^r$ with corresponding low validation errors rather than low training error.

The effect of early stopping is shown in Figure 2.4. for an ANN fitted to the 100 noisy observations $\mathbf{D}^{100}$. The error $L(\boldsymbol{\theta}^r)$ for the training set $\mathbf{D}^{tr}$ decreases each epoch, while the error calculated on the validation set increases for $r > 3$. The parameters $\boldsymbol{\theta}^r$ with low training error, $r > 5$, has high error on an independent test set. Thus, a network with better generalisation abilities can be achieved by selecting the weights $\boldsymbol{\theta}^r$ which minimises the validation error.

2.3.2 Regularisation

A less complex model may be achieved by augmenting the error function

$$L^{reg}(\mathbf{D}; \boldsymbol{\theta}) = aL(\mathbf{D}; \boldsymbol{\theta}) + bL(\boldsymbol{\theta}), \tag{2.27}$$

where $L(\boldsymbol{\theta})$ is a penalty term for the model complexity. The two new parameters a and b represents a balance between the two error terms. For $a >> b$ ($a << b$) the penalty term for model complexity has little (large) influence on the total loss $L^{reg}(\mathbf{D}; \boldsymbol{\theta})$ respectively. A common choice for $L(\boldsymbol{\theta})$ is the ridge penalty defined as the sum of squared parameters, i.e.

$$L(\boldsymbol{\theta}) = \sum_{j=1}^{F} \theta_j^2, \tag{2.28}$$

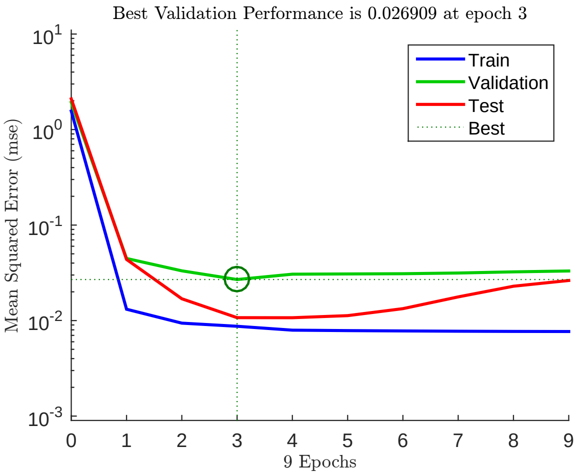


Figure 2.4: The mean squared error of the training, validation and test set as a function of $\boldsymbol{\theta}^r$. The data $\mathbf{D}^{100}$ is splitted into training, validation and test set with ratio $[.7, .15, .15]$. The number of epochs r is the number of times all training samples $\mathbf{D}^{tr}$ has been used to update the parameters $\boldsymbol{\theta}$. The mse for the training set decreases for increasing r, while the mse for the validation set is lowest for $r = 3$. This indicates that the weights $\boldsymbol{\theta}^3$ may have best generalisation abilities. The mse for of the independent increases for $r > 3$, which indicates overfitting.

where $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_F\}$. Recall that the updating formulas used in the backpropagation procedure involved derivation of the loss function $L(\mathbf{D}; \boldsymbol{\theta})$. The augmented loss function $L^{reg}(\cdot)$ has one additional term for the model complexity, hence the derivative of $L^{reg}(\cdot)$ can be written as

$$\frac{\partial}{\partial \theta_j} L^{reg}(\mathbf{D}; \boldsymbol{\theta}) = \frac{\partial}{\partial \theta_j} \left(aL(\mathbf{D}; \boldsymbol{\theta}) + bL(\boldsymbol{\theta}) \right). \tag{2.29}$$

For the case with $L(\boldsymbol{\theta})$ defined as the sum of squared weights, Equation (2.28), it is easily seen that

$$\frac{\partial}{\partial \theta_j} L^{reg}(\mathbf{D}; \boldsymbol{\theta}) = \frac{\partial}{\partial \theta_j} \Big(aL(\boldsymbol{\theta}) + bL(\boldsymbol{\theta}) \Big) = a\frac{\partial L(\mathbf{D}; \boldsymbol{\theta})}{\partial \theta_j} + 2b\theta_j. \tag{2.30}$$

The sum of squared parameters is just one example of many penalty functions. A characteristic of this penalty function is that the parameters $\theta_1, \ldots, \theta_F$ are shrunk, but not to zero unless $b = \infty$. The penalty contribution of the parameter θ_i is θ_i^2 which is very small for small θ_i. The sum of squared parameters is used throughout this project. An alternative loss function is the sum of absolute values of the parameters, called Lasso penalty (Hastie et al., 2009), which may result in more sparse models since some parameters are shrunk to zero. In the proceeding, the sum of squared parameters is used as the measure of the model complexity.

Recall that the error function (2.27) is used for minimisation. Therefore, it could be parametrised by using only one parameter as for example

$$L^{reg}(\mathbf{D}; \boldsymbol{\theta}) = (1 - b)L(\mathbf{D}; \boldsymbol{\theta}) + bL(\boldsymbol{\theta}). \tag{2.31}$$

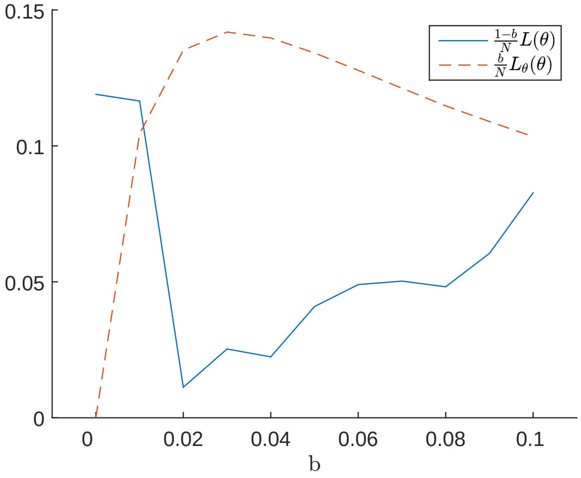


Figure 2.5: The two terms in Equation (2.31), both scaled by the reciprocal of n, computed for ANNs fitted to the dataset $\mathbf{D}^{20}$ with different values of b. The ANNs are fitted with the LM algorithm with loss function given by Equation (2.31). The sum of squared errors $L\left(\mathbf{D};\boldsymbol{\theta}\right)$ is computed over an independent test set $\mathbf{D}^{grid,151}$. Note that for small values of b, $(1-b)L\left(\mathbf{D};\boldsymbol{\theta}\right) \approx L\left(\mathbf{D};\boldsymbol{\theta}\right)$. It seems that networks with balance parameters $b \in [0.02, 0.04]$ gives lower to $L\left(\mathbf{D};\boldsymbol{\theta}\right)$ than $b > 0.4$ for this particular dataset $\mathbf{D}^{20}$, network architecture and fitting process.

The two parameters a and b are used later for the *bayesian regularisation* method described in Section 2.3.3. With the parameterisation of the loss function given in Equation (2.31), one parameter b has to be specified to balance prediction accuracy $L\left(\mathbf{D};\boldsymbol{\theta}\right)$ and model complexity $L(\boldsymbol{\theta})$. It is not straightforward to specify a value for this parameter that ensures the desired balance. The effect of b can be illustrated by fitting ANNs to the same dataset $\mathbf{D}^{20}$ and comparing the two terms in Equation (2.31), as shown in Figure 2.5. The figure indicates that networks fitted with $b \in [0.02, 0.04]$ perform better, lower error $L\left(\mathbf{D};\boldsymbol{\theta}\right)$, on the test data $\mathbf{D}^{grid,151}$. Several aspects of the fitting process have stochastic components, for example the random initialisation of the parameters $\boldsymbol{\theta}$ and the partitioning of $\mathbf{D}^{20}$ into $\mathbf{D}^{tr}$ and $\mathbf{D}^{va}$. Thus, repeating the same process results in different estimates for the optimal value of b. In the proceeding section, a method that automatically adjust the balance parameters a and b is introduced.

2.3.3 Bayesian regularisation

Determination of the parameters a and b for balancing the two error terms in the augmented error function, Equation (2.27), is not straightforward. A way to determine the optimal regularisation parameters is the Bayesian Regularisation (BR) method, first proposed in the doctoral thesis of MacKay (1992). The work extended the Bayesian interpretation of regularisation in a way that enables objective determination of a and b. For a more in depth description, we refer to MacKay (1992).

The Bayesian Regularisation (BR) approach is commonly used with the LM algorithm

Algorithm 2 Bayesian regularisation algorithm

```
Obtain $\mathbf{D}^n = \{(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^n, \mathbf{y}^n)\}$
Initialise $a$ and $b$.
Initialise $\boldsymbol{\theta}^0 = (\theta_1^0, \ldots, \theta_F)$, set $\tilde{F} = F$
repeat
    Update $\boldsymbol{\theta}^{r+1}$ by taking one LM step towards the minimum of $L^{reg}(\mathbf{D}; \boldsymbol{\theta})$
    Compute GN approx. of the Hessian $H_{GN}$
    Compute $\tilde{F} = F - 2a\texttt{Trace}\left(H_{GN}^{-1}\right)$
    Update $a = \frac{\tilde{F}}{2L(\mathbf{x})}$ and $b = \frac{n - \tilde{F}}{2L(\mathbf{x})}$
until Convergence
return $\boldsymbol{\theta}^{r+1}$                                        ▷ Model parameters
```

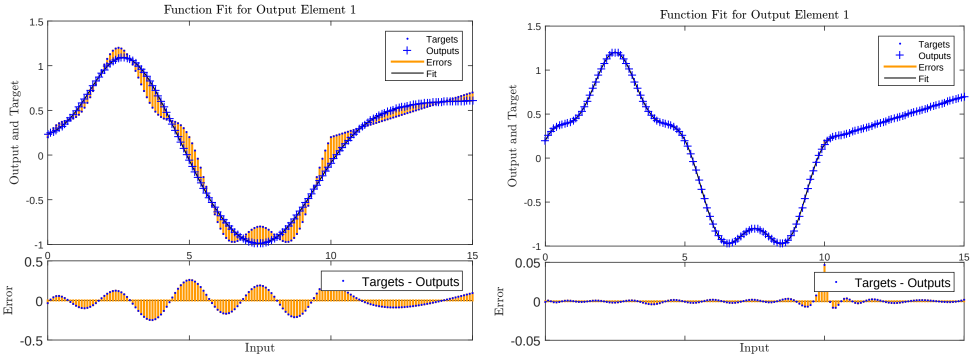
described in Section 2.2.2. The abbreviation LM-BR is used to emphasise that the BR method is used as an extension of the backpropagation algorithm with LM updates for fitting ANNs. The process of fitting networks the LM-BR is summarised in Algorithm 2. Derivation of the equations for updating a and b can be found in Hagan et al. (2003). Note that the BR approach does not use early stopping techniques. Therefore, all available data samples in $\mathbf{D}$ can be used to determine the model parameters $\boldsymbol{\theta}$, which may be beneficial when the sample size is small. A comparison of the accuracy of ANNs fitted using the back propagation with LM and LM-BR is given in the proceeding section, where the hidden layer size is altered as well.

2.3.4 Assessment of the effect of the number of vertices

A model's flexibility may be interpreted as its ability to represent a large class of functions. Increasing the number of vertices Q enables the network to represent more complex functions, but may at the same time increase the risk of overfitting. However, this problem is reduced by applying the regularisation techniques discussed in Section 2.3 for improved generalisation. The effect of the number of vertices can be illustrated by fitting ANNs with different value of Q to synthetic data. Figure 2.6 shows two ANNs with $Q = 3$ and $Q = 10$ vertices fitted to the same dataset $\mathbf{D}_0^{1000}$ without noise. The experimental setup is summarised in Table 2.3. The network with $Q = 3$ is not flexible enough to model $f^{true}(x)$. This can be seen by looking at the error plot, which is relatively high and clearly follows some trend. For the network with $Q = 10$, the errors are much smaller. This illustrates the importance of having enough vertices.

Setting Q very high is not critical if regularisation techniques as those presented in Section 2.3 are utilised. Figure 2.7 shows the root mean squared error of four ANNs fitted with the LM (early stopping) and LM-BR (regularisation) methods. The ANNs with few vertices $Q < 3$ have higher root-mean-squared errors. The error decreases by increasing the number of vertices. Comparing the the root-mean-squared errors for the ANNs fitted to $\mathbf{D}^{20}$ and $\mathbf{D}^{50}$, it seems that the latter benefits more of having a high number of vertices.

When fitting an ANN, there a are lot of choices that may effect the performance of the network. For example, a one layer MLP network architecture was assumed and the back propagation procedure was used in the derivation of the update equation. For these



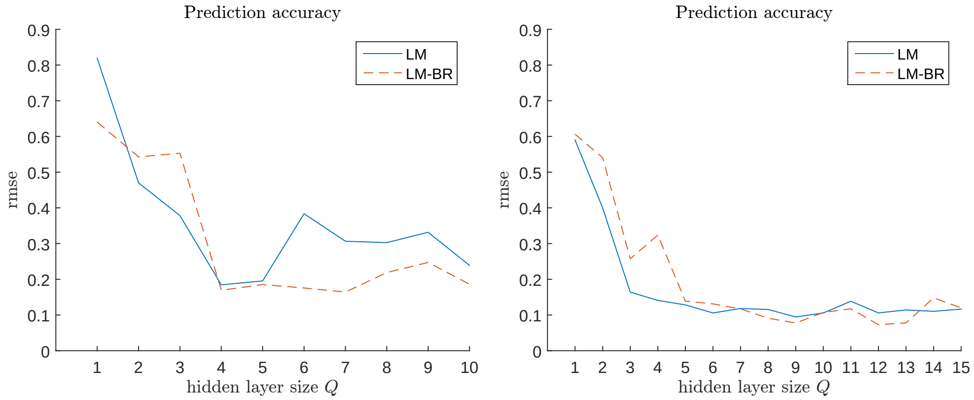
(a) 3 vertices. rmse=0.1243

(b) 10 vertices, rmse=0054.

Figure 2.6: ANNs with $Q = 3$ and $Q = 10$ vertices fitted to $\mathbf{D}_0^{1000}$ with the LM-BR algorithm. The root mean squared error is calculated over an independent test set $\mathbf{D}^{grid,151}$. The ANN with $Q = 10$ vertices is much more accurate than the simpler ANN with $Q = 3$ vertices. The former captures the non-linearities in the function $f^{true}(x)$, whereas the latter is not flexible enough to approximate $f^{true}(x)$ adequately.

Table 2.3: Experimental setup. For the LM method, the data set $\mathbf{D}$ is divided randomly into training set $\mathbf{D}^{tr}$ and validation set $\mathbf{D}^{v}$ with ratios 0.8 and 0.2. A test set $\mathbf{D}^{grid,151}$ defined as noise-free observations on a regular grid is used to calculate the root mean squared error (rmse) for different ANNs trained with different number of vertices Q and different training procedures.

Dataset $\mathbf{D}$	Design points	Observations	Q	Figure
$\mathbf{D}_0^{1000}$	$x_i \sim \texttt{Unif}(\Omega_x)$	$y_i = f^{true}(x_i)$	3 and 10	2.6
$\mathbf{D}^{20}$	$x_i \sim \texttt{Unif}(\Omega_x)$	$y_i = f^{true}(x_i) + Z_i \sim \texttt{N}\left(0, 0.1^2\right)$	$1, \ldots, 10$	2.7a
$\mathbf{D}^{50}$	$x_i \sim \texttt{Unif}(\Omega_x)$	$y_i = f^{true}(x_i) + Z_i \sim \texttt{N}\left(0, 0.1^2\right)$	$1, \ldots, 15$	2.7b



(a) Average rmse for 5 ANNs fitted to $\mathbf{D}^{20}$.

(b) Average rmse for 5 ANNs fitted to $\mathbf{D}^{50}$.

Figure 2.7: Average root mean squared as a function of the hidden layer size Q and fitting procedure. The reported results are obtained by fitting 5 ANNs with the LM procedure (solid line) and the LM-BR procedure (dotted line) with different number of vertices Q to the data samples $\mathbf{D}^{20}$ (left figure) and $\mathbf{D}^{50}$ (right figure). The root mean squared errors are computed over an independent test set $\mathbf{D}^{grid,151}$ and averaged across the 5 ANNs with the same combination of fitting procedure, Q and data samples. The rmse is high when Q is small, and decreases as Q increases.

derivations, it was implicitly assumed that the activation function σ, the transfer functions $g_1, \ldots, g_K$ and the hidden layer size Q were specified. Similarly, we chose to use the squared error function $E(\boldsymbol{\theta})$ in Equation (2.6) as a measure of model fit, and the sum of squared parameters $E_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ to describe model complexity.

The network architecture, training procedure and measure of model fit and complexity could be chosen differently. It is outside the scope of this project to go into detail of the effect of these choices. In order to increase reproducibility and credibility, we adopt the choices found in the neural network toolbox of Matlab. The toolbox enables efficient fitting of ANNs with different architectures, activation functions, training procedures etc. Throughout the project, the default parameters for two layer FFNNs with Bayesian Regularisation as training procedure are used to train ANNs. Table 2.4 gives an overview of the most important parameters used in the two layer FFNN of Matlab. For those familiar with the toolbox, it may be informal to recognise the architecture shown in Figure 2.8 . In the proceeding section, a method that can be used to form a more accurate surrogate model by using several ANNs is described.

2.4 Bagging for stabilization and quantification of uncertainty

Predictions by ANNs are often unstable. Fitting several ANNs to the same dataset will in general give different fitted networks depending on the initial model parameters $\boldsymbol{\theta}^0$. In this section, a method called *bagging* is described and utilised to achieve more stable

Table 2.4: An overview of the network architecture and other parameters used in the fitting process. The choices are similar to the default values used by the Neural network toolbox for two layer FFNN. A more in depth discussion of other choices of function, training procedures and parameters, we refer to (Hagan et al., 2003, chapters, 11-13). A thorough description of Matlabs neural network toolbox can be found in Beale et al. (2016).

Quantity	Parameter value	Explanation
Activation function	$\sigma(\mathbf{x}) = \frac{2}{1+\exp(-2*x)} - 1$	Assumed to be the tan-sigmoid function
Transfer functions	$g_k(\mathbf{x}) = \mathbf{x}$ for $k = 1, \ldots, K$	Assumed to be the identity function.
Hidden layer size	Q	Chosen by a rule of thumb for the problem at hand.
Gain	$\Delta_{\boldsymbol{\theta}}$	The LM algorithm is used to determine the gain $\Delta_{\boldsymbol{\theta}} = \boldsymbol{\theta}^{r+1} - \boldsymbol{\theta}^{r}$.
Fitness function	$L^{reg}(\boldsymbol{\theta})$	Unless stated otherwise, the Bayesian Regularisation approach (LM-BR) is used. The loss function penalises both the sum of squared errors and sum of squared parameters, see Equation (2.27).
Sets	D^{tr}, D^{va} and D^{te}	Unless stated otherwise, D^{te} is not used. The LM and LM-BR approach divides the data samples $\mathbf{D}$ randomly into training and validation set with ratio $(.8, .2)$ and $(1, 0)$ respectively.

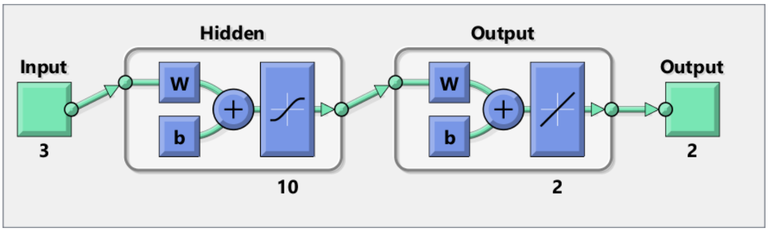


Figure 2.8: A representation of the architecture under consideration. The number of input nodes $P = 3$, hidden layer size $Q = 10$ and number of output nodes $K = 2$ can be specified by the user. The tan-sigmoidal function is used as activation function for the sum (+) of a linear transformation (w) of the input plus a vector of biases (b). The identity function is used as the transfer functions $g_1(\cdot), \ldots, g_K(\cdot)$.

Algorithm 3 Bagging

```
Data sample $\mathbf{D} = \{(\mathbf{x}^1, \mathbf{y}^1) \cdots (\mathbf{x}^n, \mathbf{y}^n)\}$
Generate $B$ bootstrap samples $\mathbf{D}^\star = \{\mathbf{D}^{\star,b}\}_{b=1}^B$
Fit a neural network $f^b(x)$ to each bootstrap sample in $\mathbf{D}^\star$
Let $\Upsilon(\mathbf{x}) = \{f^b(\mathbf{x})\}_{b=1}^B$
Define $f(x)$ as the average of $\Upsilon(\mathbf{x})$                ▷ Bagged predictor
```

predictions. For many real world applications, it is of interest to quantify the uncertainty of predictions. We study how the *bagging* method can be used to obtain a measure of the model uncertainty. The uncertainty estimate and a probabilistic representation of the surrogate model can be used to form confidence intervals for the prediction. This approach is tested on synthetic data.

2.4.1 Bagging

The bagging method, as first proposed by Breiman (1996), derives its name due to the use of *aggregated* predictions from several models, where each of these models is fitted on a *bootstrap sample* of the training set $\mathbf{D}$. Let $\mathbf{D} = (\mathbf{x}, y) = \{(\mathbf{x}^i, y^i)\}_{i=1}^n$ represent the set of all realisations of $y^i = f^{true}(\mathbf{x}^i)$, assuming $K = 1$ for simplicity. Since the observations in $\mathbf{D}$ are assumed to be independent and identically distributed (i.i.d.), any permutation of these are equally likely to be observed. A bootstrap sample of $\mathbf{D}$ is a set of n pairs $\{(\mathbf{x}^{j_i}, y^{j_i})\}_{i=1}^n$ drawn randomly with replacement from $\mathbf{D}$. The bootstrap indices $j_1, \ldots, j_n$ describe the index of the original dataset for the j'th bootstrap sample. Let

$$\mathbf{D}^\star = \{(\mathbf{x}, y)^{\star b}\}_{b=1}^B, \tag{2.32}$$

denote a set of B bootstrap samples of the original set $\mathbf{D}$. All the permutations in $\mathbf{D}^\star$ are equally likely, so inference of $f(\mathbf{x})$ should not only be based on the original set $\mathbf{x}$. Therefore, an ANN is fitted to each of the bootstrap samples in $\mathbf{D}^\star$. Let

$$\Upsilon(\mathbf{x}) = \{f^b(\mathbf{x}; \mathbf{D}_b^\star)\}_{b=1}^B, \tag{2.33}$$

denote the set of B neural network fitted to the bootstrap samples. A more stable surrogate model $f(\cdot)$ is formed by combining these neural networks. The *bagging* technique combines these predictions by averaging over them. Thus, the bagged estimator $f(\mathbf{x})$ may be written as

$$f(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} f^b(\mathbf{x}). \tag{2.34}$$

The bagging procedure is summarised in Algorithm 3 . We assume that each network $f^b(\mathbf{x})$ is an unbiased estimator of the true relation $f^{true}(\mathbf{x})$, i.e. $\mathbb{E}(f^b(\mathbf{x})) = f^{true}(\mathbf{x})$. Thus, the average $f(\mathbf{x})$ is also unbiased since

$$\mathbb{E}(f(\mathbf{x})) = \mathbb{E}\left(\frac{1}{B}\sum_{b=1}^{B} f^b(\mathbf{x})\right) = \frac{1}{B}Bf^{true}(\mathbf{x}) = f^{true}(\mathbf{x}). \tag{2.35}$$

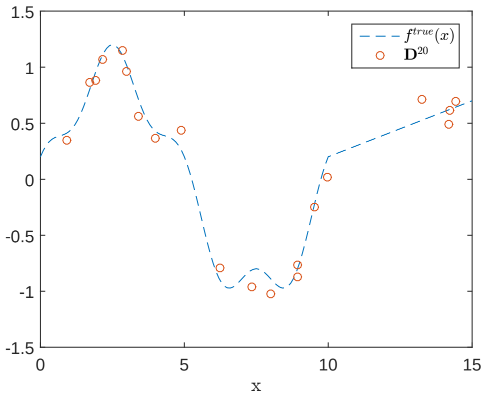


Figure 2.9: The underlying function $f^{true}(x)$ (dotted line) and the data samples $\mathbf{D}^{20}$ ('o').

2.4.2 Uncertainty estimation

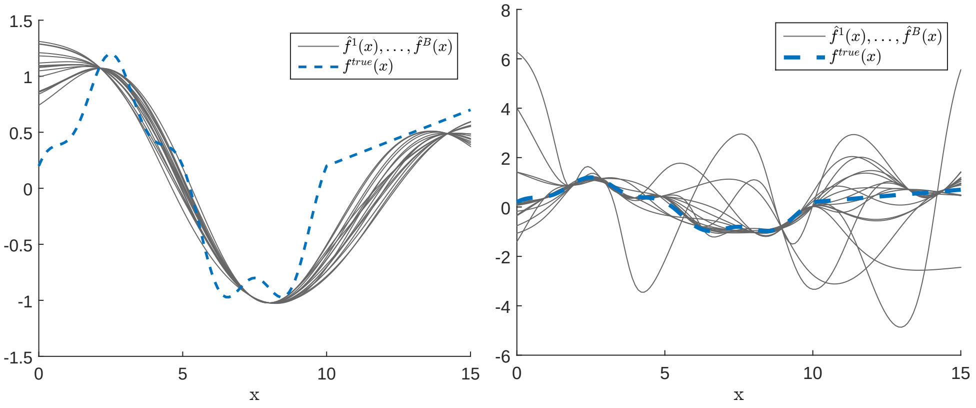
In the following, an approach for quantifying $\sigma(\mathbf{x})$ by applying the bootstrap procedure is studied. The variance $\sigma^2(\mathbf{x})$ can be estimated as the empirical variance of the B predictions $\Upsilon(\mathbf{x})$ predictions around their average $f(\mathbf{x})$ defined in Equation (2.34). This can be expressed as

$$\sigma^2(\mathbf{x}) = \frac{1}{B-1}\sum_{b=1}^{B}(f^b(\mathbf{x}) - f(\mathbf{x}))^2. \tag{2.36}$$

The estimated variance can be used to form confidence intervals for $f(\mathbf{x})$. This estimate reflects the variance of each single ANNs predictions $\Upsilon(\mathbf{x}) = \{f^1(\mathbf{x}), \ldots, f^b(\mathbf{x})\}$ around the average $f(\mathbf{x})$. The predictions of each ANN may vary much more than their average $f(\mathbf{x})$, so the variance estimate based on Equation (2.36) can be biased upwards. As proposed in Carney et al. (1999), ensemble techniques can be used to improve the variance estimate. For the running example, numerical experimentation showed that the most accurate confidence intervals was obtained by using the variance estimate from Equation (2.36). The results are omitted to limit the scope of this project, and without further discussion, the variance estimate from Equation (2.36) is used throughout this project.

As before, let $\mathbf{D}^{20}$ denote a data sample 20 noisy observations of the $f(x^i)$, where $x^1, \ldots, x^{20}$ are drawn uniformly from the domain $\Omega_x = [0, 15]$. For convenience, the realisation of $\mathbf{D}^{20}$ displayed in Figure 2.9 is used throughout this section. Fitting neural networks to $\mathbf{D}^{20}$ generally results in different networks due to random initialisation of the model parameters $\boldsymbol{\theta}^0$. It could be reasonable to fit B ANNs to the same data sample, without using bootstrapping, and use the variability of these predictions to estimate the model uncertainty $\sigma(x)$. In the proceeding, these two approaches are compared. As will become apparent, the approach based on using the same data sample may give a misleading estimate of $\sigma(x)$, and therefore motivates the bagging approach. Let

$$\Upsilon = \{\hat{f}^b(x; \mathbf{D}_b^\star\}_{b=1}^{B}, \tag{2.37}$$



(a) The predictions Υ^{orig}.

(b) The predictions Υ.

Figure 2.10: The predictions based on the B ANNs in Υ^{orig} and Υ^{orig}. The variability of the networks in Υ has more variability since each is fitted to a bootstrap sample of $\mathbf{D}$. Note the difference in the y-axis.

denote the $B = 16$ ANNs fitted to bootstrap samples of $\mathbf{D}^{20}$. Similarly, let

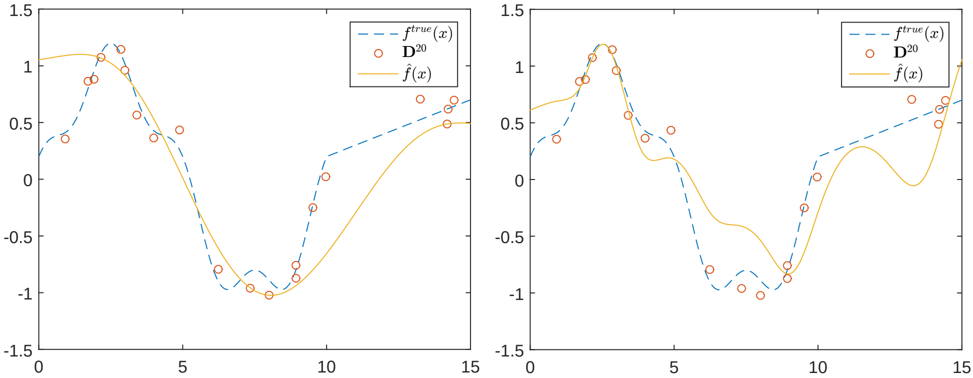
$$\Upsilon^{orig} = \{\hat{f}^b(x; \mathbf{D}^{20}\}_{b=1}^{B}, \tag{2.38}$$

denote $B = 16$ ANNs fitted to the same dataset $\mathbf{D}^{20}$. Figure 2.10 shows the corresponding predictions $\Upsilon(x)$ and $\Upsilon^{orig}(x)$. Note that each single ANN in $\Upsilon(x)$ may deviate significantly from $f^{true}(x)$. The bagged predictor obtained by averaging the $B = 16$ predictors in Υ is more stable, see Figure 2.11b. The average of the $B = 16$ networks in Υ^{orig} is included as a comparison. Both of the averaged predictors are reasonable estimates of the underlying function for the relatively limited data set $\mathbf{D}^{20}$.

The model uncertainty $\sigma(x)$ is estimated for Υ and Υ^{orig} by Equation (2.36). The corresponding bands $f(x) \pm 1.96\sigma(x)$ are shown in Figure 2.12. The model uncertainty $\sigma(x)$ should ideally be high (low) when $|f^{true} - f(x)|$ is high (low). However, Figure 2.12a shows that the band $f(x) \pm 1.96\sigma(x)$ for Υ^{orig} is, for some values of x, very narrow (wide) in regions where the deviation $|f^{true} - \hat{f}(x)|$ is high (low). Thus, the estimate $\sigma(x)$ based on Υ^{orig} does not reflect the model uncertainty.

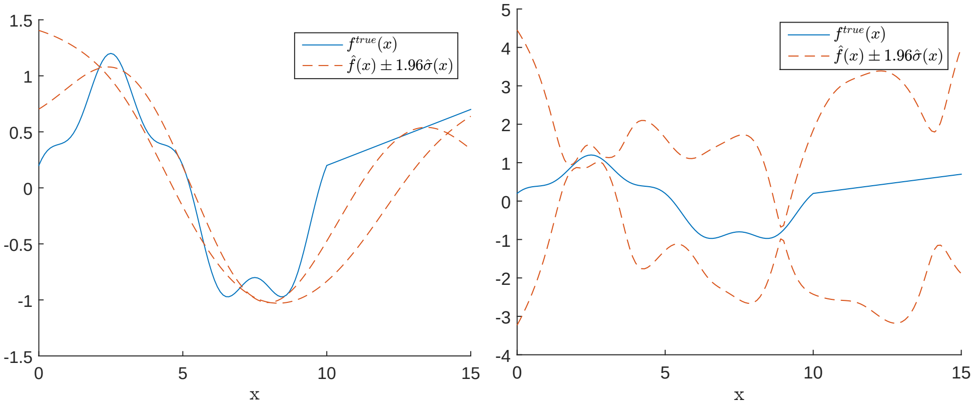
From the band $f(x) \pm 1.96\sigma(x)$ for Υ, shown in Figure 2.12a, it seems that the mentioned effect is less apparent. This motivates the approach of estimating the variance of ANNs fitted to bootstrap samples $\mathbf{D}^{\star}$, instead of ANNs fitted to the original data $\mathbf{D}$. In the proceeding, bands on the form $f(x) \pm c\sigma(x)$, where $c > 0$ is a constant, are assigned a natural interpretation as confidence intervals for the underlying function $f^{true}(x)$.

The distribution of $f^{true}(x)$, around the given the surrogate model prediction $f(x)$, is not analytical tractable. In this section, the distribution is estimated by assuming that $f^{true}(x)$ is normal distributed with mean $f(x)$ and standard deviation $\sigma(x)$. A more thorough discussion of the distribution FFNN fitted to bootstrap samples can be found in Paass (1993). An $(1 - \alpha)$ CI for a quantity should ideally cover the true quantity with probability $1 - \alpha$. Since the CI for $f^{true}(x)$ is analytical intractable, it is of interest



(a) The average $f(x)$ of the B networks Υ^{orig} **(b)** The average the $f(x)$ of the B networks Υ

Figure 2.11: The averaged predictor (solid line) based on Υ^{orig} (left subfigure)and Υ (right subfigure). The data sample $\mathbf{D}^{20}$ and $f^{true}(x)$ are shown as circles and a dashed line. Both predictors are reasonable estimates of $f^{true}(x)$ given the relatively small and noisy data set $\mathbf{D}^{20}$.



(a) Υ^{orig}. **(b)** Υ.

Figure 2.12: The band $f(x) \pm 1.96\sigma(x)$ for the B (dashed lines) for Υ^{orig} (left subfigure) and Υ (right subfigure). Note that band for Υ^{orig} is narrow in some regions where the difference $|f(x) - f^{true}(x)|$ is large. This indicates that using Υ^{orig} is unsuitable for estimating the model uncertainty $\sigma(x)$.

to evaluate the coverage of the approximated CI. The CI for $f^{true}(x)$ is estimated by assuming that

$$f^{true}(x) \sim \texttt{Norm}(f(x), \sigma^2). \tag{2.39}$$

With the above assumption, a $(1-\alpha)$ CI for f^{true} is given by

$$CI_{1-\alpha}(x) = f(x) \pm c_{\alpha/2}\hat{\sigma}(x), \tag{2.40}$$

where the constant $c_{\alpha/2}$ is defined as

$$c_{\alpha/2} = \texttt{P}(f^{true}(x)) \leq c_{\alpha/2}) = \alpha. \tag{2.41}$$

The empirical coverage of the confidence interval can be evaluated on synthetic data. Let $t_1, \ldots, t_F$ denote the index of the F in the regular grid $\mathbf{D}^{grid,151}$. The empirical coverage probability is obtained by computing the fraction of points in the test set $\mathbf{D}^{grid,151}$ that are within the estimated CI. Using the estimated CI, Equation (2.40), this ratio can be expressed as

$$\frac{1}{F}\sum_{i=1}^{F} \texttt{I}\left(f\left(x_{t_i}\right) - c_{\frac{\alpha}{2}}\sigma\left(x_{t_i}\right) \leq f(x) \leq f\left(x_{t_i}\right) + c_{\frac{\alpha}{2}}\sigma\left(x_{t_i}\right)\right), \tag{2.42}$$

where the indicator function $I(A)$ is one (zero) if the expression A is (true) false.

Recall the B ANNs Υ fitted to bootstrap samples of $\mathbf{D}^{20}$. For this particular case, the band $f(x) + \sigma(x)$ covered the underlying function $f^{true}(x)$ for all values of $x \in [0, 15]$. In general, the estimates $f(x)$ and $\sigma(x)$ depends on the data sample $\mathbf{D}$ used in the fitting process. For example, observing 50 noisy observations $\mathbf{D}^{50}$ and performing the same fitting process as for Υ for $\mathbf{D}^{20}$ in Section 2.4.2, results in the CI intervals shown in Figure 2.13, with coverage ratio of 0.9007. Experimental results indicates that the estimated CIs have a desirable coverage ratio for different sample sizes. By increasing (decreasing) α wider (narrower) CIs are obtained and the coverage ratio can be calculated by using Equation (2.42). For different sample sizes and confidence α, the corresponding CIs tend to have a coverage ratio that reflects the desired confidence.

In this chapter, a surrogate model $f(x)$ based on B ANNs fitted to bootstrap samples of the available data samples $\mathbf{D}^n$ has been studied. The bagging procedure can be used to estimated the model uncertainty, which can be useful for estimating CIs. The above aspects have been demonstrated on a one-dimensional function. In the proceeding chapter, an optimise procedure that utilise $f(x)$ and $\sigma(x)$ in the search for x that maximise $f^{true}(x)$ is proposed.

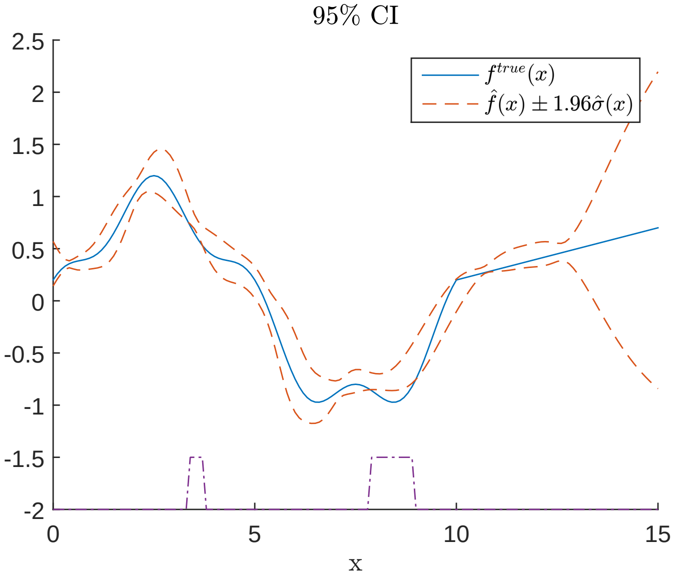


Figure 2.13: The function $f^{true}(x)$ (solid line) and the estimated 95-CI (dotted lie) based on the B ANNs Υ fitted to bootstrap samples of $\mathbf{D}^{50}$. The bottom line (dash-dotted) increases if $f^{true}(x)$ is outside the CI. The estimated CI seem to capture the model uncertainty adequately. Note the CI is wide for x in the upper part of the interval $[0, 15]$, which is a typical behaviour near the boundaries of Ω_x.

Chapter 3

Optimisation by sequential design

3.1 Introduction

Optimisation of $f^{true}(\mathbf{x})$ is not straightforward, due to the lack of an analytical expressions for $f^{true}(\mathbf{x})$ and since evaluations of $f^{true}(\mathbf{x})$ are only available through a time-consuming simulation model. In this chapter, techniques from the surrogate-based optimisation methodology is used in the search of an approximatively global optimiser of $f^{true}(\mathbf{x})$. Such methods use a surrogate model, also called response surface, meta-model and function approximation, to guide the search for the optimum. Since the number of function evaluations is limited, it is of interest to concentrate the function evaluations in regions where $f^{true}(\mathbf{x})$ is high. This may be achieved by first observing a part of the experimental design $\mathbf{D}^n = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$, and iteratively selecting the next design point $\mathbf{x}^{n+1}$ based on the surrogate model $f(\mathbf{x}; \mathbf{D}^n)$ fitted to all available data samples $\mathbf{D}^n$.

Efficient global optimisation typically requires a balance between exploitation and exploration. As sampling strategy, the next point $\mathbf{x}^{n+1}$ is selected as the maximiser of an acquisition, or infill function, $u(\cdot)$ that balances these aspects, i.e.

$$\mathbf{x}^{n+1} = \underset{\mathbf{x} \in \Omega_{\mathbf{x}}}{\operatorname{argmax}} \quad u\left(\mathbf{x}; \mathbf{D}^n\right). \tag{3.1}$$

A general procedure of selecting design point sequentially is shown in Algorithm 4. Common acquisition functions based on the probability of improvement (PI), expected improvement (EI) and an upper confidence bound (UC) use a statistical interpretation of $f^{true}(\mathbf{x})$. For surrogate model based on Gaussian process, the statistical distribution of $f^{true}(\mathbf{x})$ follows from the (assumed) Gaussian process, the observed data $\mathbf{D}^n$ and the estimated model parameters. For surrogate model based on other regression models, for example ANNs, such inherit distribution of $f^{true}(\mathbf{x})$ is seldom available. In the proceeding section, two approaches for estimating the distribution of $f^{true}(\mathbf{x})$ is presented. The two approaches uses different estimates of the cumulative distribution function (CDF). In section 3.3, the EI, PI and UC are derived and illustrated for both approaches.

In section 3.4, we extend these acquisition functions such that several design points can be selected in each iteration. Using several acquisition functions instead of only one

Algorithm 4 Optimisation by sequential design

```
Infill function $u(\cdot)$
Obtain $\mathbf{D}^{n_0} = \{(\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^{n_0}, \mathbf{y}^{n_0})\}$
$n = n_0$
repeat
    Find $\mathbf{x}^{new} = \mathrm{argmax}\, u(\mathbf{x}; \mathbf{D}^n)$
    Sample $y^{new} = f(\mathbf{x}^{new})$
    Augment $\mathbf{D}^n = \{\mathbf{D}^n, (\mathbf{x}^{new}, y^{new})\}$
until Stopping criterion met
return $\mathbf{D}^n$                    ▷ Data samples used for further analysis.
```

have shown promising results (Hoffman et al., 2011), and for the simulation model under consideration, the evaluation of the selected points can be performed efficiently in parallel.

The optimisation of $f^{true}(\mathbf{x})$ may be difficult due to the mentioned time-consuming evaluations and possibly stochastic output. In addition, the optimisation may be restricted to a constraint $c(\mathbf{x})$ that may only be evaluated by an equally time-consuming process. A way to deal with this problem is by estimating the probability that $\mathbf{x}$ satisfy the constraint $c(\mathbf{x})$, and incorporating the estimate into the infill function. See for example Schonlau et al. (1998) for a discussion of such extension of the EI criterion, and Gelbart et al. (2014); Williams et al. (2010) for a more in depth study of sequential designs for constrained optimisation. A drawback of these methods is that they require an additional output variable. In section 3.5, we propose a method that avoids the additional variable by introducing a penalty function for constraint violation.

3.1.1 Notational remarks

In the proceeding, let

$$\mathbf{D}^n = \{(\mathbf{x}^i, y^i)\}_{i=1}^n, \tag{3.2}$$

denote a set of n data samples. For the running example, let

$$\mathbf{D}_{0.1}^{lhc,n} = \{(x^i, f^{true}(x^i) + Z^i)\}_{i=1}^n, \tag{3.3}$$

where

$$Z^i \sim \texttt{Norm}(0, 0.1), \tag{3.4}$$

denote n noisy observations of $f^{true}(x)$ where the design points $x^1, \dots, x^n$ are generated using a Latin Hypercube sampling procedure. For a thorough discussion of such space filling designs, see Santner et al. (2003); Forrester et al. (2008).

3.2 Statistical distribution of $f(\mathbf{x})$

Let $\mathbf{D}^n$ denote all current data samples, and $f(\mathbf{x})$ the surrogate model defined as the average of the B ANNs $\Upsilon = \{f^b(\mathbf{x})\}_{b=1}^B$, each fitted to a bootstrap sample of $\mathbf{D}^n$, as described in Chapter 2. The unknown function $f^{true}(\mathbf{x})$ is interpreted as a random variable, with observations $\Upsilon(\mathbf{x}) = f^1(\mathbf{x}), \ldots, f^B(\mathbf{x})$. Inference based on these observations is used to select $\mathbf{x}^{n+1}$.

In this project, two approaches for estimating the distribution of $Y(\mathbf{x}) = f^{true}(\mathbf{x})$ based on $\Upsilon(\mathbf{x})$, namely by

- Calculating the empirical CDF of $\Upsilon(\mathbf{x})$
- Assuming $Y(\mathbf{x})$ is normal distributed with mean $\mu(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} f^b(\mathbf{x})$ and standard deviation $\sigma(\mathbf{x}) = \hat{\sigma}(\mathbf{x})$

In order to simplify the derivation of the infill functions and emphasise the viewpoint of $f^{true}(\mathbf{x})$ as a random variable, the following notation is adopted. Let $Y(\mathbf{x}) = f^{true}(\mathbf{x})$ represent a random variable, with unknown distribution, and observations $\{y^1(\mathbf{x}), \ldots, y^B(\mathbf{x})\} = \Upsilon(\mathbf{x})$. Assuming that $(y^1(\mathbf{x}), \ldots, y^B(\mathbf{x}))$ are independent, identically distributed random variables with CDF $F_{Y(\mathbf{x})}(y)$, the empirical CDF $\hat{F}_{Y(\mathbf{x})}(y)$ is defined as

$$\hat{F}_{Y(\mathbf{x})}(y) = \frac{1}{B}\sum_{b=1}^{B}\left[y^b(\mathbf{x}) \leq y\right], \tag{3.5}$$

where the notation $[A]$ is one (zero) if A is true (false). An alternative approach of estimating the distribution of $Y(\mathbf{x})$ is by assuming that $Y(\mathbf{x})$ is normal distributed

$$Y(\mathbf{x}) \sim \texttt{Norm}\left(\mu(\mathbf{x}), \sigma^2(\mathbf{x})\right), \tag{3.6}$$

with the two fist moments $\mu(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ estimated by the empirical mean and variance of $\Upsilon(\mathbf{x})$. Note that this approach was used to estimate confidence intervals in Section 2.4.2. By transformation of variables, the corresponding CDF can be expressed as

$$\hat{F}^{norm}_{Y(\mathbf{x})}(y) = \Phi(z), \tag{3.7}$$

where $\Phi(\cdot)$ denotes the CDF for the a standard normal distributed variable, and $z(\mathbf{x}) = \frac{y(\mathbf{x})-\mu(\mathbf{x})}{\sigma(\mathbf{x})}$. The difference between the two approaches is illustrated on an example. Let Υ denote a set of $B = 16$ ANNs fiited to $\mathbf{D}^{10,lhc}_{0.1}$. These observations and ANNs are used throughout the section. Figure 3.1 shows $f^{true}(x)$, $f(x)$, $\mathbf{D}^{10,lhc}$ and Υ. For a specific x, say $x = 5$, $\hat{F}_{Y(5)}(y)$ can be estimated using Equation (3.5) and the $B = 16$ observations

$$\Upsilon(5) = \{-0.974, -0.0456, , \ldots, 0.492\}. \tag{3.8}$$

Correspondingly, $\hat{F}^{norm}_{Y(5)}(y)$ is easily computed for a normal distributed variable with empirical mean $f(5) = 0.134$, and empirical $\sigma(5)^2 = 0.334$. The CDFs are shown in Figure 3.2 and it is clear that the CDF $\hat{F}^{norm}_{Y(5)}(y)$ is smoother than the alternative. Note also that this CDF is zero (one) only asymptotically when $\sigma(\mathbf{x}) > 0$, whereas the alternative reaches these values when the argument is lower (larger) than the smallest (largest) value of $\Upsilon(\mathbf{x})$. In the proceeding, the infill functions are derived using the two approaches for estimation of the distribution of Y.

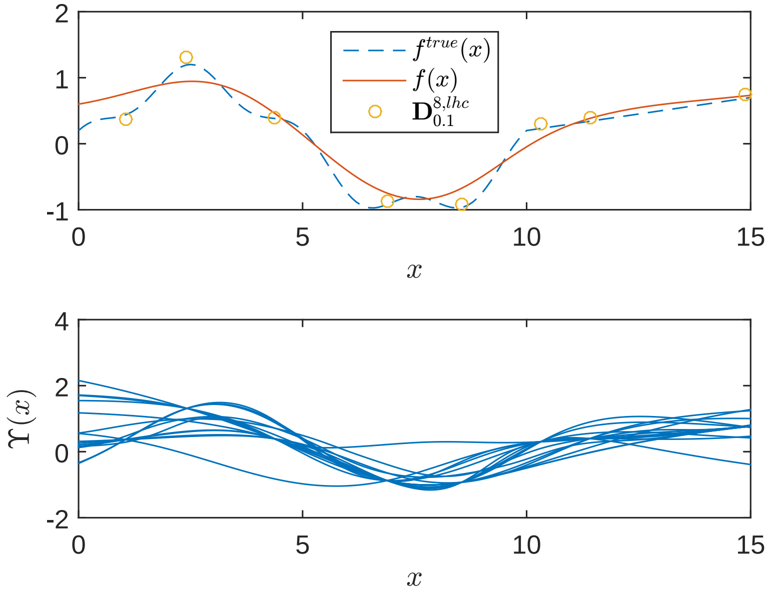


Figure 3.1: The data samples $\mathbf{D}_{0.1}^{8,lhc}$ and the $B = 16$ ANNs Υ. The maximum of $f(x)$ is $\mu^+ = 0.9434$.

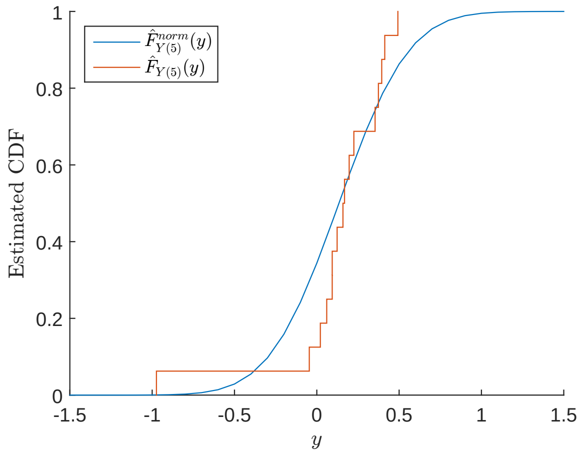


Figure 3.2: The estimated CDF's $\hat{F}_{Y(5)}(y)$ and $\hat{F}^{norm}_{Y(5)}(y)$ obtained by Equation (3.5) and Equation (3.7) respectively. Note that the former are less smooth than the alternative. Both CDFs can be used to estimate the distribution of $f^{true}(x)$

3.3 Infill functions

The performance of a particular infill function is problem-dependent, but typically a balance between exploitation and exploration is necessary in order to avoid getting stuck in local optimums and exhaustive exploration of the domain Ω_x. In this section, infill functions based on the two CDF's are derived. Some iterations of the sequential process of fitting surrogate model and selecting the next point for evaluations is included.

3.3.1 Probability of improvement

Infill functions based on the probability of improvement (PI), first proposed by Kushner (1964), aims at selecting $\mathbf{x}^{n+1}$ such that $Y = f^{true}\left(\mathbf{x}^{n+1}\right)$ is most likely to improve over an incumbent μ^+. For deterministic simulation models, the incumbent is often defined as the best (highest) observed simulation output. For stochastic simulation model under consideration, a more reliable incumbent can be achieved by defining μ^+ as

$$\mu^+ = \max_{i=1,\dots,n} \quad f\left(\mathbf{x}^i\right), \tag{3.9}$$

namely the maximum of the predictions in the current design points $\mathbf{x}^1, \dots, \mathbf{x}^n$. The probability that the random variable Y is an improvement over μ^+ is

$$\begin{aligned} \mathtt{PI}_{\mu^+}(\mathbf{x}) &= \mathtt{P}\left(Y(\mathbf{x}) > \mu^+\right) \\ &= \int_{\mu^+}^{\infty} \mathtt{P}\left(Y(\mathbf{x}) = y\right) \mathrm{d}y \\ &= 1 - \mathtt{P}\left(Y(\mathbf{x}) \leq \mu^+\right) \\ &= 1 - F_{Y(\mathbf{x})}(\mu^+), \end{aligned} \tag{3.10}$$

where $P\left(Y(\mathbf{x}) = y\right)$ and $F_{Y(\mathbf{x})}(y)$ denotes the PDF and CDF of $Y(\mathbf{x})$ respectively.

Inserting for PI as acquisition function $u(\cdot)$ in Equation (3.1), using $\hat{F}_{Y(\mathbf{x})}(y)$ and $\hat{F}^{norm}_{Y(\mathbf{x})}(y)$ from equations (3.5) and (3.7) respectively, yields the following expressions

$$\mathbf{x}^{n+1} = \underset{\mathbf{x}\in\Omega_{\mathbf{x}}}{\operatorname{argmax}} \quad 1 - \sum_{b=1}^{B} \left[y_i \leq \mu^+\right], \tag{3.11}$$

$$\mathbf{x}^{n+1} = \underset{\mathbf{x}\in\Omega_{\mathbf{x}}}{\operatorname{argmax}} \quad 1 - \Phi\left(z(\mathbf{x})\right), \tag{3.12}$$

where $z(\mathbf{x}) = \frac{\mu^+ - f(\mathbf{x})}{\sigma(\mathbf{x})}$. From Equation (3.11) it is easily seen that by using $\hat{F}_{Y(\mathbf{x})}(y)$, $\mathbf{x}^{new}$ is the point where most of the B ANN predictions $f^1(\mathbf{x}^{new}), \dots, f^B(\mathbf{x}^{new})$ is greater than the incumbent μ^+. Alternatively, using the imposed normal distribution, the maximiser $\mathbf{x}^{new}$ in Equation (3.12) is the maximiser of $1 - \Phi\left(z(\mathbf{x})\right)$. Since $\Phi(\cdot)$ is monotonically increasing, $\mathbf{x}^{new}$ is the minimiser of $z(\mathbf{x})$, which is small when $\mu^+ << f(\mathbf{x})$ and $\sigma(\mathbf{x})$ small. This corresponds to high values of $f(\mathbf{x})$ with little variability $\sigma(\mathbf{x})$ across $\Upsilon(\mathbf{x})$.

The two alternative versions of the PI criterion is illustrated on the sample $\mathbf{D}^8_{0.1}$. Figure 3.3 shows the PI using Equation (3.11) and Equation (3.12). Note that the PI following

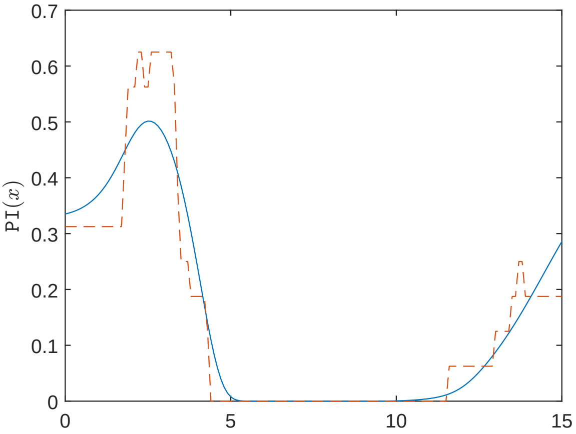


Figure 3.3: The PI computed using $\hat{F}^{norm}_{Y(\mathbf{x})}(y)$ (solid line) and $\hat{F}_{Y(\mathbf{x})}(y)$ (dotted line). The latter is less smooth due to the discontinuous form of $\hat{F}_{Y(\mathbf{x})}(y)$.

from the assumed normal distribution of Y is smoother than the alternative. Moreover, it is always greater than 0 unless $\sigma(\mathbf{x}) = 0$, as opposed to the alternative which is 0 for all $\mathbf{x}$ satisfying

$$\{\mathbf{x} : f^1(\mathbf{x}), \ldots, f^B(\mathbf{x}) \leq \mu^+\}. \tag{3.13}$$

The PI criterion is sensitive with respect to μ^+: perturbing the incumbent with a small amount δ may greatly affect the maximiser x^{n+1}. This is illustrated in Figure 3.4 An increase (decrease) of μ^+ tends to more (less) exploration compared to exploitation In the proceeding, a criterion with less sensitivity to μ^+ is presented. The criterion balance exploration and exploitation by weighting the magnitude of the potential improvement, as opposed to only weighting the probability of improvement.

3.3.2 Expected Improvement

As before, let $Y(\mathbf{x})$ denote a random variable with independent, identically distributed observations $\{y_1(\mathbf{x}), \ldots, y_B(\mathbf{x})\}$ and an assumed known CDF. In order to simplify the derivation of the expected improvement (EI), the reference variable $\mathbf{x}$ is neglected from the notation. Hence, we write the above-mentioned quantities as Y, $\{y_1, \ldots, y_B\}$ and let $F(y)$ denote the corresponding CDF.

Define improvement over an incumbent y^+ as

$$\mathtt{I}(Y) = \begin{cases} y - y^+, & \text{if } y > y^+ \\ 0, & \text{if } y \leq y^+ \end{cases} \tag{3.14}$$

The EI is defined as the expectation of the improvement function, i.e.

$$\mathtt{EI} = \mathtt{E}\{\mathtt{I}(Y)\}. \tag{3.15}$$

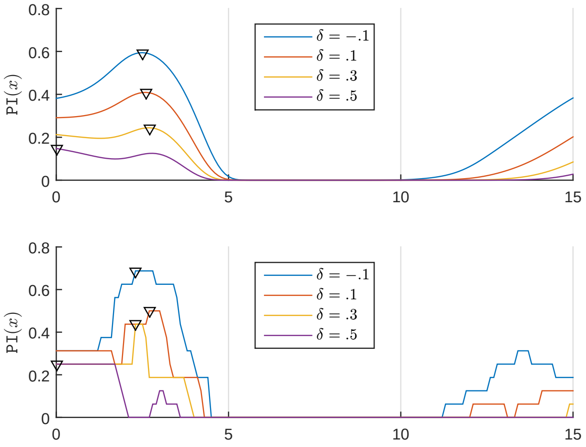


Figure 3.4: The PI computed using $\hat{F}^{norm}_{Y(\mathbf{x})}(y)$ (upper subfigure) and $\hat{F}_{Y(\mathbf{x})}(y)$ (lower subfigure) using incumbent $\mu^+ + \delta$ for different values of δ. Small (large) δ gives large (low) probability of improvement and encourages a higher degree of exploitation (exploration).

From the definition of the expectation operator $\mathtt{E}\{\cdot\}$, this is the same as

$$\mathtt{EI} = \int_{-\infty}^{\infty} \mathtt{I}\,(y)\,\mathtt{P}\,(Y=y)\;\mathrm{d}y, \tag{3.16}$$

which, by inserting for the improvement function $\mathtt{I}\,(y)$ and splitting the integral, is the same as

$$\mathtt{EI} = \int_{y^+}^{\infty} \left(y - y^+\right)\,\mathtt{P}\,(Y=y)\;\mathrm{d}y + 0\int_{-\infty}^{y^+} \mathtt{P}\,(Y=y)\;\mathrm{d}y \tag{3.17}$$

where the second term is obviously 0. By splitting the first term, the EI can be written as

$$\mathtt{EI} = \int_{y^+}^{\infty} y\,\mathtt{P}\,(Y=y)\;\mathrm{d}y\; - y^+\int_{y^+}^{\infty} \mathtt{P}\,(Y=y)\;\mathrm{d}y, \tag{3.18}$$

where the second term is a constant y^+ multiplied by the PI given by Equation (3.10). Inserting for the PI yields

$$\mathtt{EI} = \int_{y^+}^{\infty} y\,\mathtt{P}\,(Y=y)\;\mathrm{d}y - y^+\left(1 - F\left(y^+\right)\right). \tag{3.19}$$

The integral over the weighted PDF, the first term of Equation (3.19), may by complicated depending on the PDF of the random variable Y. However, for the two version of the CDF, this term can be expressed with well-known functions. By assuming the empirical CDF, defined in Equation (3.5), the first term in Equation (3.19) can be expressed as

$$\int_{y^+}^{\infty} y\,\mathtt{P}\left(Y = y\right)\,\mathrm{d}y = \frac{1}{B}\sum_{\{b:y_b>y^+\}} y_b \tag{3.20}$$

$$= \frac{1}{B}\sum_{b=1}^{B} y_b\left[y_b > y^+\right], \tag{3.21}$$

where mass $\frac{1}{B}$ is assigned to each of the B observations $y^1, \ldots, y^B$. By insertion of Equation (3.21) into Equation (3.19), the EI takes the form

$$\begin{aligned}\mathtt{EI} &= \frac{1}{B}\sum_{b=1}^{B} y_b\left[y_b > y^+\right] - y^+\left(1 - F\left(y^+\right)\right)\\ &= \frac{1}{B}\sum_{b=1}^{B}\left[y_b > y^+\right]\left(y_b - y^+\right),\end{aligned} \tag{3.22}$$

where the last transition follows from the definition of the CDF in Equation (3.5). Recall from Equation (3.14) that $[y_b > y^+]\,(y_b - y^+)$ is the improvement $\mathtt{I}(y_b)$ over y^+, hence the EI in Equation (3.22) is simply a sum of the improvements for the predictions $y_1, \ldots, y_B$.

An expression for the EI using the alternative CDF $F_Y^{norm}(y)$, defined in Equation (3.7), is derived in the following. Recall that Y is a normal distributed variable with mean μ and variance σ^2, with PDF

$$\mathtt{P}\left(Y = y\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\{-\frac{\left(y-\mu\right)^2}{2\sigma}\}. \tag{3.23}$$

Inserting $\mathtt{P}\left(Y = y\right)$ and performing the change of variable, $Z = \frac{Y-\mu}{\sigma}$, the first term in Equation (3.19) can be written as

$$\begin{aligned}\int_{y^+}^{\infty} y\,\mathtt{P}\left(Y = y\right)\,\mathrm{d}y &= \frac{1}{\sqrt{2\pi}\sigma}\int_{y^+}^{\infty} y\,\exp\{-\frac{\left(y-\mu\right)^2}{2\sigma^2}\}\,\mathrm{d}y\\ &= \frac{1}{\sqrt{2\pi}}\int_{z}^{\infty}\left(\sigma z + \mu\right)\,\exp\{-\frac{z^2}{2}\}\,\mathrm{d}z\\ &= \frac{\sigma}{\sqrt{2\pi}}\int_{z}^{\infty} z\,\exp\{\frac{z^2}{2}\}\,\mathrm{d}z + \frac{\mu}{\sqrt{2\pi}}\int_{z}^{\infty}\exp\{\frac{z^2}{2}\}\,\mathrm{d}z,\end{aligned} \tag{3.24}$$

with $z = \frac{y^+-\mu}{\sigma}$. By recognising that the second term in the last equation is a constant μ multiplied by the PI for a standard normal variable Z, and integrating the first term, the last equation can be written with familiar functions as

$$\begin{aligned}\int_{y^+}^{\infty} y\,\mathtt{P}\left(Y = y\right)\,\mathrm{d}y &= \frac{\sigma}{\sqrt{2\pi}}\exp\{-\frac{z^2}{2}\} + \mu\left(1 - \Phi\left(z\right)\right)\\ &= \sigma\phi\left(z\right) + \mu\left(1 - \Phi\left(z\right)\right).\end{aligned} \tag{3.25}$$

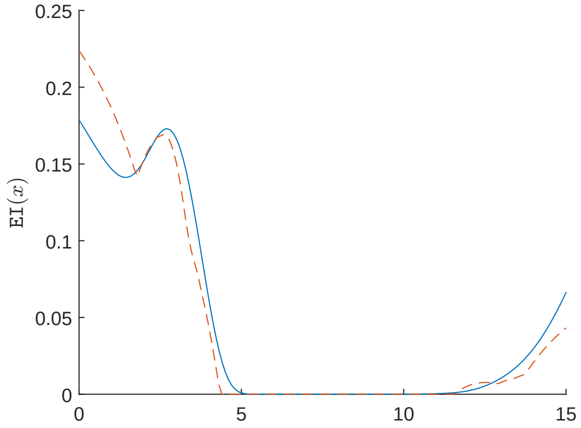


Figure 3.5: The EI computed using $\hat{F}^{norm}_{Y(\mathbf{x})}(y)$ (solid line) and $\hat{F}_{Y(\mathbf{x})}(y)$ (dotted line). Note that the EI criterion typically has several modes as is the case for this example.

Thus, inserting the above expression into Equation (3.19), yields the following expression for the EI

$$\begin{aligned} \mathtt{EI} &= \sigma\phi\left(z\right) + \mu\left(1 - \Phi\left(z\right)\right) - y^{+}\left(1 - \Phi\left(z\right)\right) \\ &= \sigma\left[\phi\left(z\right) + z\left(\Phi\left(z\right) - 1\right)\right]. \end{aligned} \tag{3.26}$$

Figure 3.5 shows the EI over incumbent μ^{+}. The EI have two modes at $x = 0$ and $x \approx 3$. From the illustration of $\Upsilon(x)$ and $f(x)$ in Figure 3.1, it can be seen that these two regions corresponds have high variability of Υ and high prediction $f(x)$. The expected criterion is less sensitive to the incumbent μ^{+}. This is illustrated in Figure 3.6 for $\mu^{+} + \delta$ for different values of δ. Note that the same point $x^{n+1} = 0$ maximises the EI criterion for most combinations of δ and CDF. The EI criterion tends to select x^{n+1} where $f\left(\mathbf{x}\right)$ is high and or $\sigma\left(\mathbf{x}\right)$ is high.

3.3.3 Upper confidence bound

Let the UCB be defined as

$$\mathtt{UCB}\left(\mathbf{x};\kappa\right) = f\left(\mathbf{x}\right) + \kappa\sigma\left(\mathbf{x}\right), \kappa \geq 0. \tag{3.27}$$

High (low) κ gives high (low) degree of exploration. The two special cases $\kappa = 0$ and $\kappa = \infty$ results in maximisation of only $f\left(\mathbf{x}\right)$ and $\left(\mathbf{x}\right)$ respectively. The acquisition function $\mathtt{UC}\left(\mathbf{x};\kappa\right)$ requires the user to determine the parameter κ. The $\mathtt{UC}$ is shown in the bottom left corner of Figure 3.7 Illustration of $f(x)$, Υ, $\mathbf{D}^{8,lhc}_{0.1}$ and the infill function based on $\mathtt{PI}$ and $\mathtt{EI}$ seen previously in this chapter, are included to ease comparison.

3.3.4 Generalised Expected Improvement with constraints

In this section a more general form of the improvement function $\mathtt{I}$ is introduced. The extension were first proposed by Schonlau et al. (1998), and this project used to handle

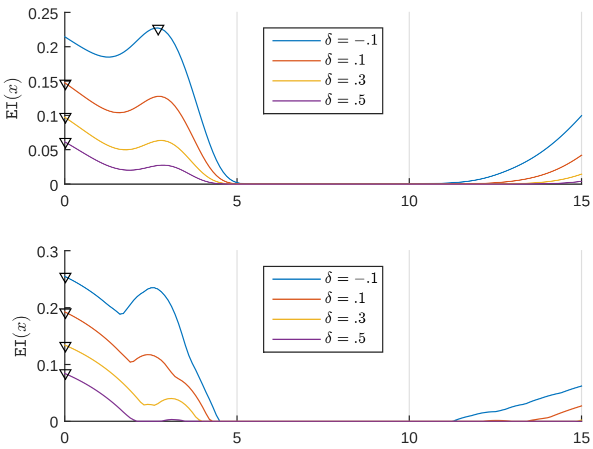


Figure 3.6: The EI computed using $\hat{F}^{norm}_{Y(\mathbf{x})}(y)$ (upper subfigure) and $\hat{F}_{Y(\mathbf{x})}(y)$ (lower subfigure) using incumbent $\mu^{+} + \delta$ for different values of δ. Small (large) δ gives large (low) probability of improvement and encourages a higher degree of exploitation (exploration). Note that the shape of the EI is relatively similar for different values of δ.

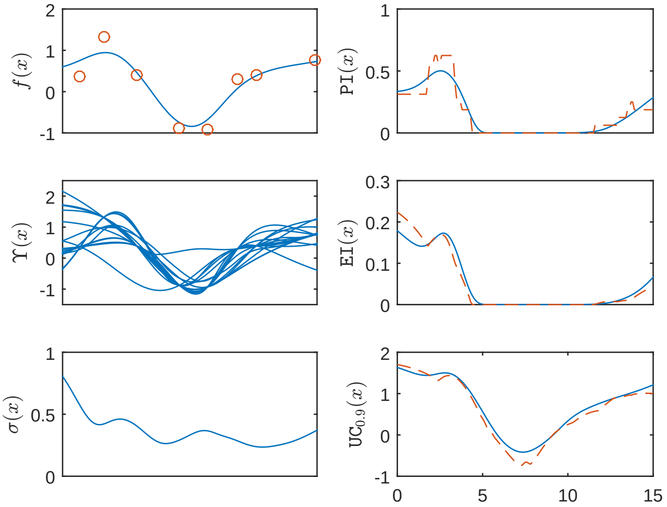


Figure 3.7: The left column shows aspects related to the surrogate model, namely $f(x)$, $\Upsilon(x)$ and σ, and the data sample $\mathbf{D}^{8,lhc}_{0.1}$. The right column shows the corresponding infill functions based on $\texttt{PI}$, $\texttt{EI}$ and $\texttt{UC}_{90}$. The $\texttt{PI}$ is highest near the maximum of $f(x)$, and the two alternatives are highest in the high variance region near $x = 0$.

improvements subjected to constraints on an additional response variable. Let the improvement function $\mathtt{I}^{\rho}(\mathbf{x})$ be defined as

$$\mathtt{I}^{\rho}(\mathbf{x}) = \begin{cases} (y(\mathbf{x}) - y^{+})^{\rho}, & \text{if } y(\mathbf{x}) > y^{+} \\ 0, & \text{otherwise,} \end{cases} \tag{3.28}$$

i.e. the improvement from Equation (3.14) raised to the power of $\rho \geq 0$. For $\rho > 1$, the improvements $(Y - \mu^{+})^{\rho} >> (Y - \mu^{+})$, so large improvements $(Y - \mu^{+})$ are assigned more weight for increasing ρ. Thus, the general improvement function $\mathtt{I}^{\rho}$ may be used to balance exploration and exploitation in a more systematic manner. In this project, the parameter values $\rho = \{0, 1\}$ are used. For these values, the expectation of $\mathtt{I}^{\rho}$ is the PI and EI derived in Section 3.3.1 and Section 3.3.2 respectively. Recall that the PI and EI were derived using CDFs based on an imposed normal distribution and an empirical CDF. For the latter, $\mathtt{I}^{\rho}$ may be easily obtained by taking the ρ'th power of the term $(y_b - y^{+})$ in Equation (3.22). The alternative requires more cumbersome derivation that can be found in Schonlau et al. (1998).

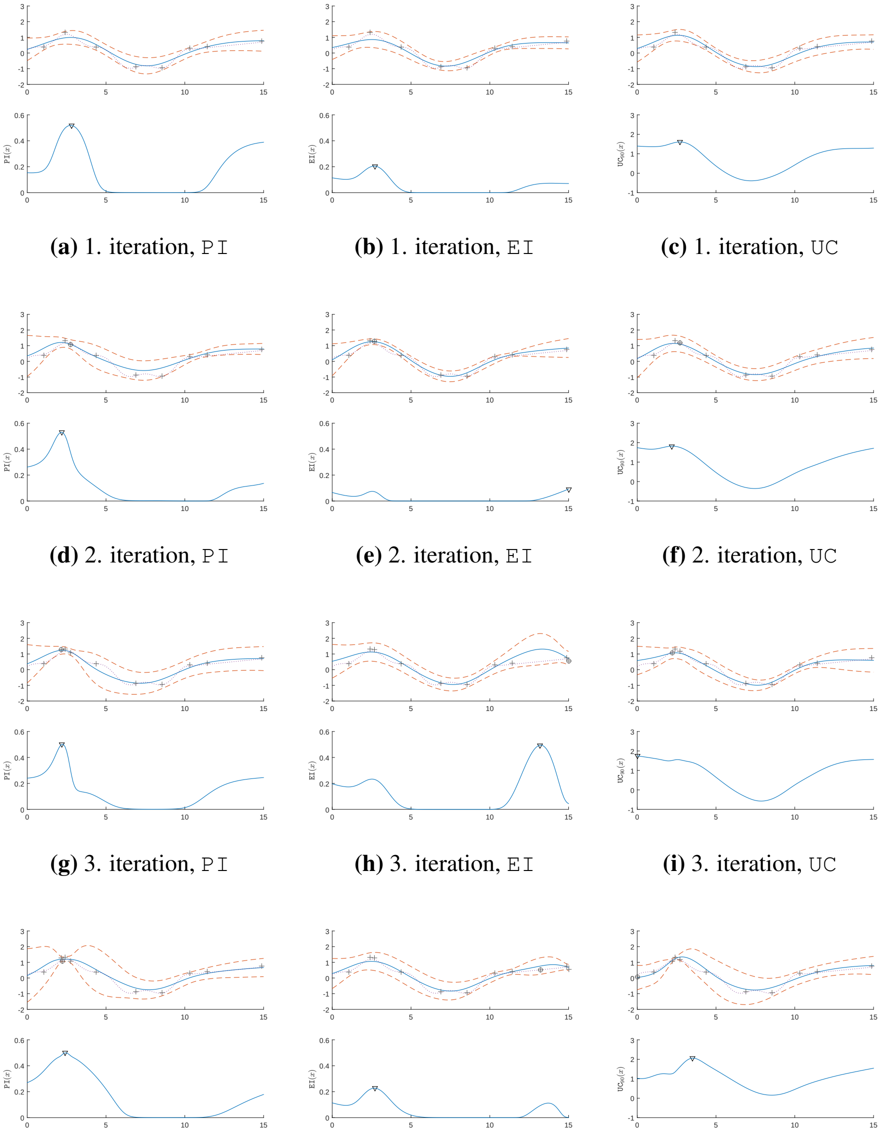
3.3.5 Sampling paths for synthetic data

The criteria PI,EI and UC are reasonable candidates for infill functions. Assessing the performance of each of them may be cumbersome, since the performance is dependent on the initial sample $\mathbf{D}^{n_0}$, the problem under consideration, and possible random effects in the fitting of the ANNs used as surrogate model fitting. In this section, we illustrate some iteration of iterations of a sequential method in order to capture the characteristics of the infill function under consideration.

In this section, only one point is sampled in each iteration, i.e $S = 1$. For notational simplicity, let $f^{n}(x)$ and $\sigma^{n}(x)$ denote the surrogate model and the associated model uncertainty at the i'th iteration. Figure 3.8 show four iterations of the sequential method for the running example with an initial data sample $\mathbf{D}^{8,lhc}$. For all $i = 1, \ldots, 4$ iterations $f^{n}(x)$, $\sigma^{n}(x)$ and the infill function $u(x)$ and the current data sample $\mathbf{D}^{n}$ is displayed.

The `PI` criterion is characterised by less global search than the alternatives. Note that the model uncertainty $\sigma^{n+1}(x;)$ tend to be lower than $\sigma^{n}(x)$ near the last observed data sample (x^{n+1}, y^{n+1}). Since the initial sample $\mathbf{D}_{0.1}^{8,lhc}$ is relatively large, the initial surrogate models has much knowledge of the underlying function $f^{true}(x)$. Thus, the benefit of infill functions with high degree of exploration is less apparent.

By starting with a smaller initial sample $\mathbf{D}_{0.1}^{3,lhc}$, a higher degree of exploration may be necessary for efficient global optimisation. Figure 3.9 shows 20 iterations of the sequential method with $\mathbf{D}_{0.1}^{3,lhc}$ as initial sample. For all cases, the rightmost region has highest predicted value in the first $i = 1, \ldots, 16$ iterations. Note that the `PI` infill function samples points almost exactly at the predicted maximum, and fails to identify the global maxima. The sample paths obtained with `EI` and `UC` as criterion, are spread more evenly in the domain $[0, 15]$. Around the 15. and 16. iteration, the sampling strategy based on the two criteria identifies the region around the global optimum, as opposed to the `PI` criterion. This indicates that infill functions with a high degree of exploration are more suitable for problems where the initial sample is sparse.



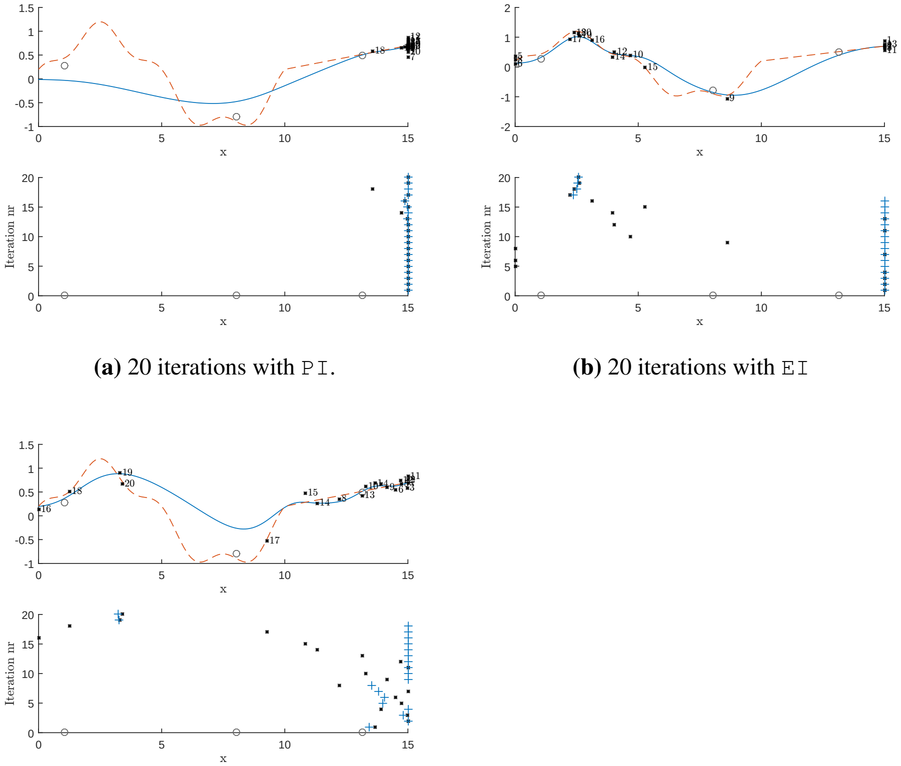
(a) 1. iteration, PI **(b)** 1. iteration, EI **(c)** 1. iteration, UC

(d) 2. iteration, PI **(e)** 2. iteration, EI **(f)** 2. iteration, UC

(g) 3. iteration, PI **(h)** 3. iteration, EI **(i)** 3. iteration, UC

(j) 4. iteration, PI **(k)** 4. iteration, EI **(l)** 4. iteration, UC

Figure 3.8: Four iterations of the sequential optimisation method using $\mathbf{D}_{0.1}^{8,lhc}$ as initial sample, surrogate model $f^n(x)$ based on $B = 20$ ANNs, using PI (left column), EI (middle column) and UC$_{90}$ as infill function respectively. The upper sub-figures shows $f^n(x)$ (solid line), $f^n(x) \pm \sigma^n(x)$ (dashed line), $f^{true}(x)$ (dotted line), observations $\mathbf{D}^n$ $(+)$ where last observed point is enclosed in a circle. The lower subfigures shows the infill function and its maximiser (downward triangle). The infill function are based on $\hat{F}^{norm}_{Y(\mathbf{x})}(y)$. The points selected using PI as infill function, are near the maximiser of $f^n(x)$. The points selected by the EI criterion have more spread. Note that the EI$(\cdot)$ is reduced in regions near the last sampled points due to reduced model uncertainty $\sigma(\cdot)$. The UC$_{90}$ criterion selects points where $f(x)$ and/or $\sigma(x)$ is high.



(a) 20 iterations with `PI`.

(b) 20 iterations with `EI`

(c) 20 iterations with `UC`

Figure 3.9: The upper sub-figures shows $f^{true}(x)$ (dashed line), the initial sample $\mathbf{D}^{lhc,3}_{0.1}$ (circles), the pairs (x^i, y^i) (small diamond marked with the iteration number $i = 1, \ldots, 20$) and the prediction $f(x)$ (solid line) for the last iteration. The points $x^1, \ldots, x^{20}$ are selected using `PI`, `EI` and `UC` as infill function. The lower sub-figures shows the selected points x^i (small $*$) and the predicted maxima $x^{max,n} = \operatorname{argmax} f^n(x)$ (+). The sequential optimisation method based on PI as infill fails to identify the global maxima due to the high degree of exploitation.

Table 3.1: Explanation of different approaches for selecting S points $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}$.

Technique	Explanation	Challenges
Joint maximisation	Select $\mathbf{x}^1, \ldots, \mathbf{x}^S$ that jointly maximise an infill function $u(\mathbf{x}^1, \ldots, \mathbf{x}^S; \mathbf{D})$	Computationally demanding, can be avoided by using an sequential approach see for example (Schonlau et al., 1998).
Sequential maximisation	Compute $\mathbf{x}^1, \ldots, \mathbf{x}^{S'}$ such that $\mathbf{x}^s$ maximise $u_s(\mathbf{x}; \mathbf{D})$	$\mathbf{x}^1, \ldots, \mathbf{x}^S$ may contain several (approximately) equal points. Can be avoided by applying clustering techniques, see for example Ponweiser et al. (2008); Ginsbourger et al. (2007).

3.4 Several input points

It may be of interest to select several points $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}$ between each update of the surrogate model. Sequential optimisation strategies such as the Efficient Global optimisation (EGO) algorithm (Jones et al., 1998) selects only one design point $\mathbf{x}^{n+1}$ between each refitting of the model. Selecting $S > 1$ design points in each iteration may offer benefits as

- reduced number of surrogate model updates,
- $f(\mathbf{x}^{n+1}), \ldots, f(\mathbf{x}^{n+S})$ evaluated in parallel.

The simulation model under consideration can be evaluated efficiently in parallel. Therefore, selecting $S > 1$ points in each iteration can significantly reduce the time consumption of the sequential process described in Algorithm 4. In the following sections, two approaches for selecting the S points are discussed. They are distinguished by whether one or several infill functions are used to select the $S > 1$ points. A short description and possible challenges of the two approaches are described briefly in Table 3.1. For surrogate models based on a GP, a covariance structure is available and can be utilised to select the S points. Since such structure is unavailable for the surrogate model under consideration, some alternative methods for selecting the S points $\mathbf{x}^1, \ldots, \mathbf{x}^S$ are proposed. The methods are relatively simple compared to similar heuristics (Ponweiser et al., 2008; Ginsbourger et al., 2007). A thorough analysis and discussion of these methods are outside the scope of this project. Instead, they are illustrated for the running example using infill functions based on `EI`.

3.4.1 Joint maximisation

Let the S infill points as be defined as

$$\{\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}\} = \operatorname{argmax} u\left(\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}\right), \tag{3.29}$$

namely the S point that jointly maximise $u(\cdot)$. Note that the infill function $u(\cdot)$ in Equation (3.29) is defined with S points as argument. The joint distribution of $\mathbf{x}^{n+1}, \dots, \mathbf{x}^{n+S}$ may be complex and analytical intractable, and the numerical optimisation time-consuming. For example, the generalised improvement function $\mathtt{I}^{\rho}$, Equation (3.28), can be augmented to

$$\mathtt{I}^{\rho}_{S} = \left[max\{0, y(\mathbf{x}^1) - y^+, \dots, y(\mathbf{x}^S) - y^+\}\right]^{\rho} \tag{3.30}$$

for the case where S points are selected. This extension was first defined by Schonlau et al. (1998) as the "ρ-step `EI`" and the criteria is studied more in depth in Ginsbourger et al. (2007), where it is referred to as the multi-points `EI`. Calculating the expectation of the above expression is difficult since the multivariate distribution may be complex or unknown and the set of combinations $\mathbf{x}^1, \dots, \mathbf{x}^S$ is large. However, approximations of the maximisers $\mathbf{x}^1, \dots, \mathbf{x}^S$ of Equation (3.29) can be achieved by applying suitable simplifications. As proposed by Schonlau et al. (1998), a reasonable simplification is to select the S points sequentially as the univariate maximisers

$$\mathbf{x}^{n+i} = \operatorname{argmax} u\left(\mathbf{x}; \mathbf{S}^i\right), \quad \text{for } i = 1, \dots, S, \tag{3.31}$$

conditioned on the previous selected points $\mathbf{S}$ defined as

$$\mathbf{S}^i = \begin{cases} \{\}, & \text{if } i = 1 \\ \mathbf{x}^{n+1}, \dots, \mathbf{x}^{n+i-1}, & \text{if } i = 2, 3, \dots, S \end{cases} \tag{3.32}$$

where $\{\}$ denote the empty set. However, for the surrogate model considered in this project, the selected points $\mathbf{S}^i$ does not affect the prediction $f\left(\mathbf{x}\right)$ and $\sigma\left(\mathbf{x}\right)$ unless the function f^{true} is evaluated at the points $\mathbf{S}^i$ and the model refitted. Thus, the points $\mathbf{x}^{n+1}, \dots, \mathbf{x}^{n+S}$ maximising Equation (3.31) can be written as

$$\mathbf{x}^{n+i} = \operatorname{argmax} u\left(\mathbf{x}\right), \quad \text{for } i = 1, \dots, S, \tag{3.33}$$

and are therefore all equal. Note that this is not the case for a Gaussian process, since the estimated variance depends on the design points $\mathbf{S}^i$ (Schonlau et al., 1998).

Maximisation of (3.29) by the sequential simplification results in S duplicates, and is therefore not a good strategy since it is desirable that $\mathbf{x}^1, \dots, \mathbf{x}^S$ has some degree of spread. The latter can be achieved by augmenting the expression for the S joint maximisers, Equation (3.29), to

$$u\left(\mathbf{x}^{n+1}, \dots, \mathbf{x}^{n+S}\right) = \sum_{i=1}^{S} u\left(\mathbf{x}^i\right) - L^{spread}\left(\mathbf{x}^{n+1}, \dots, \mathbf{x}^{n+S}\right), \tag{3.34}$$

where the term L^{spread} is a loss-function that penalise points $\mathbf{x}^{n+1}, \dots, \mathbf{x}^{n+S}$ that are close. If $L^{spread}(\cdot) << u(\cdot)$ the effect of the additional term $L^{spread}(\cdot)$ is low, and vice versa. In the following, loss.functions based on a measure of the pairwise distances between the points $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}$ are proposed. Let $\mathbf{d}^S$ be defined as

$$\begin{aligned}\mathbf{d}^S = \{&d(\mathbf{x}^{n+1}, \mathbf{x}^{n+2}), \ldots, d(\mathbf{x}^{n+1}, \mathbf{x}^{n+S}),\\ &d(\mathbf{x}^{n+2}, \mathbf{x}^{n+3}), \ldots, d(\mathbf{x}^{n+2}, \mathbf{x}^{n+S}),\\ &\vdots\\ &d(\mathbf{x}^{n+S-1}, \mathbf{x}^{n+S})\},\end{aligned} \tag{3.35}$$

the set of distances $d(\mathbf{x}^i, \mathbf{x}^j)$ between all pairs of points $\mathbf{x}^i, \mathbf{x}^j \in \{\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}\}$. The distance $d(\mathbf{x}^i, \mathbf{x}^j)$ is assumed to satisfy

$$d(\mathbf{x}', \mathbf{x}'') = d(\mathbf{x}'', \mathbf{x}') \geq 0, \tag{3.36}$$

and only the euclidean distance is considered in this project. The set $\mathbf{d}^S$ contains $|\mathbf{d}^S| = S(S-1)$ elements, and for notational convenience let $d_1, \ldots, d_{|\mathbf{d}^S|}$ denote the elements of $\mathbf{d}^S$ defined in Equation (3.35). Functions $L^{spread}(\cdot)$ that only depends on the distances $\mathbf{d}\left(\mathbf{x}^1, \ldots, \mathbf{x}^S\right)$ can be expressed as

$$L^{spread}\left(\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}\right) = L^{spread}\left(\mathbf{d}^S\right). \tag{3.37}$$

A natural choice of $L^{spread}\left(\mathbf{d}\left(\mathbf{x}^1, \ldots, \mathbf{x}^S\right)\right)$ is a function that penalise all distances that are less than a threshold η, i.e.

$$L^{spread}\left(\mathbf{d}^S\right) = C^{spread} \sum_{i=1}^{|\mathbf{d}^S|} [d_i < \eta], \tag{3.38}$$

where the notation $[A]$ is one (zero) if A is true (false) and C^{spread} a non-negative constant. An illustration of $[d < \eta]$ compared to e^{-d} is shown in Figure 3.10. Inserting the above equation into Equation (3.34) gives the following expression for the S joint maximisers

$$u\left(\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}\right) = \sum_{i=1}^{S} u\left(\mathbf{x}^i\right) - C^{spread} \sum_{i=1}^{|\mathbf{d}^S|} [d_i < \eta]. \tag{3.39}$$

Thus, the maximisers $\{\mathbf{x}^1, \ldots, \mathbf{x}^S\}$ are strongly (weakly) encouraged to have distances of at least $\geq \eta$ from each other for C^{spread} large (small). In some sense, η is a *range* parameter that reflects the desired minimum pairwise distance, while C^{spread} affects the balance between exploitation of $u()$ and spread of $\mathbf{x}^1, \ldots, \mathbf{x}^S$. In the following, the set of joint maximisers obtained using the above equation is illustrated on the running example using the EI as infill function $u()$ for different values of S, η and C^{spread} and the surrogate model fitted to $\mathbf{D}_{0.1}^{8,lhc}$.

The EI generally decreases as $\mathbf{D}$ increases, therefore it may be reasonable to let C^{spread} depend on the current iteration. Let u^{max} be defined as

$$u^{max} = \max_{\mathbf{x} \in \Omega_{\mathbf{x}}} \quad u\left(\mathbf{x}\right), \tag{3.40}$$

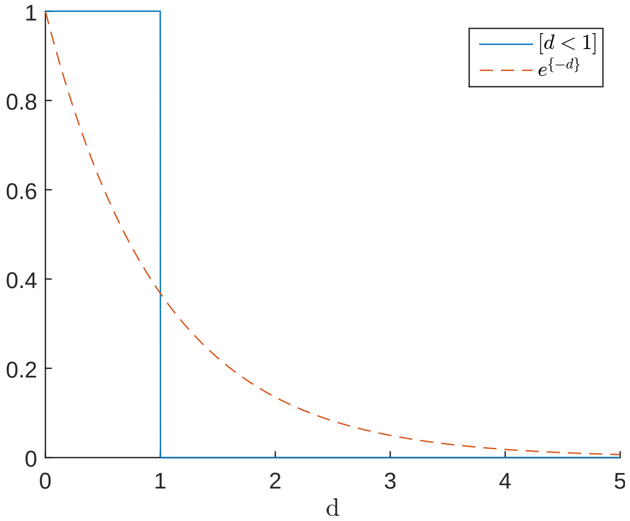


Figure 3.10: The indicator function $[d < \eta]$ (solid line) with $\eta = 1$ and the exponential function e^{-d} (dashed line) as a function of d. The former is zero for all $d \geq \eta$ and the latter approaches zero fast for increasing d. The exponential function e^{-d} increases as $d > 0$ approaches zero, i.e. small distances d corresponds to large penalties $C^{spread}e^{-C^{decay}d}$. In contrast, the term $C^{spread}\,[d_i < \eta]$ is equal for all $d < \eta$.

and let

$$C^{spread} \propto u^{max}, \tag{3.41}$$

i.e. C^{spread} is a fraction of the maximum of $u(\cdot)$ at the current iteration. This may be used to balance the two terms in Equation (3.39) in a meaningful way. The values of C^{spread}, η used in the illustrations are listed in Table 3.2 , with references to figures showing $\mathbf{S}^S$ for $S = 2, \ldots, 5$. The chosen parameter values $C^{spread} = \{u^{max}, \frac{1}{5}u^{max}\}$ corresponds to very high and moderate penalty, respectively, for all pairwise distances in $\mathbf{d}$ smaller than η. Figure 3.11 illustrate the S selected points obtained for the former parameter value.

Note that for all values of S and η, the selected points $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}$ are unique and have pairwise distances greater than or equal to η. This is due to the large value of C^{spread}, which strongly encourage set of points with pairwise distances at least η.

Figure 3.12 shows the selected points using a smaller C^{spread}. Comparing Figure 3.11 and 3.12, it is clear that the sampled points are similar for η small (middle subplots). However, for η large, the sampled points obtained using a smaller C^{spread} are not unique. This may be understood by observing that when C^{spread} is small, the right sum in Equation (3.39) is small, hence the joint maximisers may have pairwise distances less than η. Moreover, the selected points may be duplicates since the penalty $C^{spread}\,[d < \eta]$ is constant for all $d < \eta$. To summarise, the joint maximisers $\mathbf{x}^1, \ldots, \mathbf{x}^S$, obtained by using the proposed penalty function, seem to have a reasonable spread that may be controlled by the parameters η and C^{spread}. A possible drawback of the penalty function is that $\mathbf{x}^1, \ldots, \mathbf{x}^S$ may contain duplicates.

Table 3.2: Values of the parameters C^{spread} and η from Equation (3.39) used to determine the joint maximisers $\mathbf{x}^1, \ldots, \mathbf{x}^S$. The refereed figures shows the joint maximsiers for $S = 2, \ldots, 5$. The values for η corresponds to a fraction of $\frac{1}{30}$ and $\frac{1}{10}$ of the range $x \in [0, 15]$.

C^{spread}	η	Interpretation of L^{spread}	Figure
u^{max}	0.5	High penalty for all pairwise distances less than a small size	3.11, middle subfigure
u^{max}	1.5	High penalty for all pairwise distances less than a moderate size	3.11, lower subfigure
$\frac{1}{5}u^{max}$	0.5	Moderate penalty for all pairwise distances less than a small size	3.12, middle subfigure
$\frac{1}{5}u^{max}$	1.5	Moderate penalty for all pairwise distances less than a moderate size	3.12, lower subfigure

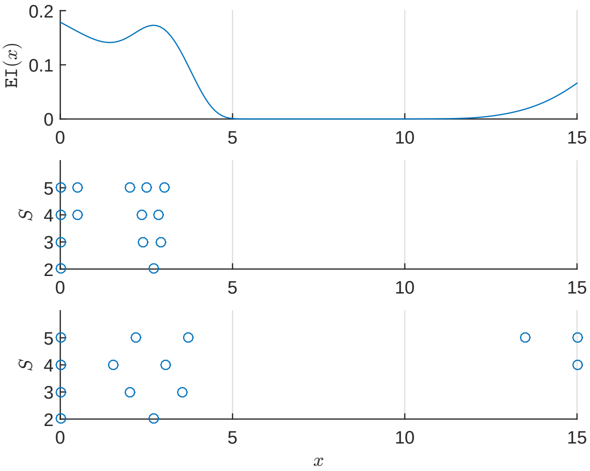


Figure 3.11: The infill function $u(x)$ (upper subfigure) and the S selected points $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}$ for $S = 2, \ldots, 5$ using $C^{spread} = u^{max}$ and $\eta = \frac{1}{30}$ (middle subfigure) and $\eta = \frac{1}{10}$ (lower subfigure).

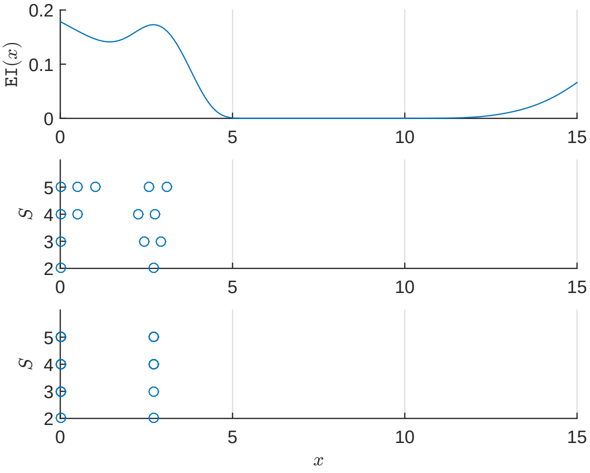


Figure 3.12: The infill function $u(x)$ (upper subfigure) and the S selected points $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}$ for $S = 2, \ldots, 5$ using $C^{spread} = \frac{1}{5}u^{max}$ and $\eta = \frac{1}{30}$ (middle subfigure) and $\eta = \frac{1}{10}$ (lower subfigure).

Motivated by the above discussion, an alternative penalty function $L^{spread}\left(\mathbf{d}^S\right)$ is proposed. A smoother version of the penalty function from Equation (3.38) can be obtained by replacing the indicator function $[d_i < \eta]$ with $e^{-C_2 d_i}$, i.e.

$$L^{spread}\left(\mathbf{d}^S\right) = C^{spread} \sum_{i=1}^{|\mathbf{d}^S|} e^{-C^{decay} d_i}. \tag{3.42}$$

The constant C^{decay} is redundant since it could be incorporated in the distance measures d_i, but is included to emphasise that the rate of decay can be controlled. Note that the (reciprocal) of C^{decay} affects the range of the penalty function, similar to η from Equation (3.38), with small (large) values giving a rapid (slow) diminishing penalty $L^{spread}(\mathbf{d})$ for distances $\mathbf{d}$. The function e^{-d} approaches zero (one) for large (small) d. Thus, the penalty function is approximately zero if all distances in $\mathbf{d}$ are large, and approximately $C^{spread}|\mathbf{d}^S|$ if all distances in $\mathbf{d}$ are close to zero. Note that this asymptotic behaviour is similar to the penalty function defined in Equation (3.38). Moreover, each pairwise distance d_i has an exponential contribution, thus $L^{spread}(\cdot)$ is small only if all distances $\mathbf{d}$ are large. By inserting $L^{spread}(\cdot)$ from Equation (3.42) into Equation (3.34), the joint maximisers is given by

$$u\left(\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}\right) = \sum_{i=1}^{S} u\left(\mathbf{x}^i\right) - C^{spread} \sum_{i=1}^{|\mathbf{d}^S|} e^{-C^{decay} d_i}. \tag{3.43}$$

In the proceeding, $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}$ are computed for $S = 2, \ldots, 5$, a fixed value for $C^{spread} = 2u^{max}$ and different values of C^{decay} for the running example $\mathbf{D}_{0.1}^{8,lhc}$. The joint maximisers obtained for three different values of C^{decay} are shown in Figure 3.13a The points $\mathbf{x}^1, \ldots, \mathbf{x}^S$ tend to be centred in regions where $u()$ is high, while keeping a reasonable spread. Note that the sets $\mathbf{x}^1, \ldots, \mathbf{x}^S$ does not contain duplicates, contrary to what was observed for the penalty function based on the indicator function.

An alternative loss-function could be defined as

$$L^{spread}\left(\mathbf{d}^S\right) = C^{spread} e^{C^{decay}\sum_{i=1}^{|\mathbf{d}^S|} -d_i}, \tag{3.44}$$

penalising the sum of pairwise distances. However, this penalty term is small as long the sum of distances is large. Therefore, a set of points $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}$ where all pairwise distances $\mathbf{d}^S$ are small, except one, results in a large sum and consequently a small penalty $L^{spread}\left(\mathbf{d}^S\right)$. Thus, the loss-function in the above equation may not be suitable for selecting $\mathbf{S}$ with a desired amount of spread.

3.4.2 Pool of infill functions

In this section, methods that utilise several infill functions to select the points $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}$ are studied. The rationale is that since there is no guarantee that a particular infill function has overall best performance, it is reasonable to consider several functions. As demonstrated in Hoffman et al. (2011), considering several acquisition functions as criterion for selecting $\mathbf{x}^{n+1}$, often outperforms strategies where a single infill function is used in all iterations.

However, in order to utilise parallel computing, we are interested in selecting $S > 1$ in each iteration. This can be achieved by determining the $S' \geq S$ maximisers

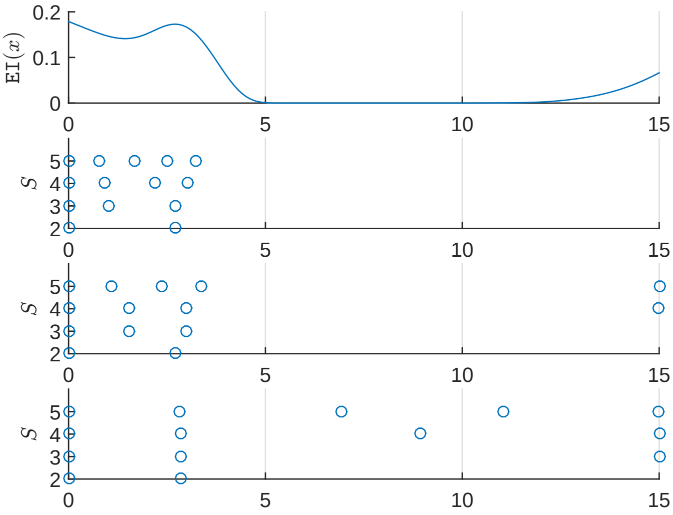
$$\mathbf{x}^{n+i} = \operatorname{argmax} u^i\left(\mathbf{x}; \mathbf{D}\right), \quad \text{for } i = 1, \ldots, S', \tag{3.45}$$

of $u_1, \ldots, u_{S'}$, and selecting a subset of these points as the S design points $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}$. The S' maximisers may be equal, or approximately equal, hence a criteria that ensures that the S selected points $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}$ have some spread may be reasonable. Examples of clustering techniques for selecting the S points can be found in Ponweiser et al. (2008); Jones (2001). Possible drawbacks of these techniques is that they may be computationally expensive, the performance may depend on several user-specified parameters and the lack of analytical tractability.

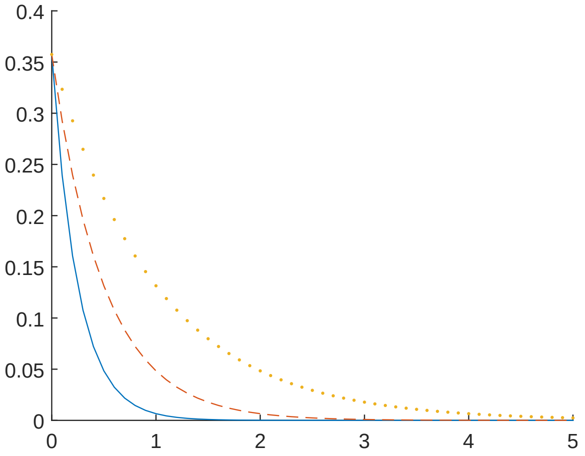
An alternative approach is proposed in the following. Let $L^{penalty}\left(\mathbf{x}; \mathbf{S}^i\right)$ denote a function that is high when $\mathbf{x}$ is close to previously selected points $\mathbf{S}^i$, defined in Equation (3.32). An sequential approach for selecting the S point is to select the i'th points as the maximiser of

$$\mathbf{x}^{n+i} = \operatorname{argmax} u^i\left(\mathbf{x}; \mathbf{D}\right) - L_i^{spread}\left(\mathbf{S}^i\right), \quad \text{for } i = 1, \ldots, S. \tag{3.46}$$

The above expression can be interpreted as follows. The first point $\mathbf{x}^{n+1}$ is simply the maximiser of the infill function $u^1(\cdot)$. The second point $\mathbf{x}^{n+2}$ is the maximiser of $u^2(\cdot)$ adjusted with a penalty $L_i^{spread}(\mathbf{x}^1)$ for the closeness of $\mathbf{x}^{n+2}$ to the set $\mathbf{S}^1 = \mathbf{x}^{n+1}$. The process is repeated $i = 1, \ldots, S$ times, and the set $\mathbf{S}^i$ and the infill function $u^i(x)$ is used in the i'th iteration. In order to balance the two terms in (3.46) in a sensible way, it is reasonable to let $L_i(\cdot)$ depend on the i'th infill function $u_i(\cdot)$.



(a) The infill function $u(x)$ (upper subfigure) and $\mathbf{S}^S$ for $S = 2, \ldots, 5$ using $C^{decay} = \{4, 2, 1\}$ (2nd, 3.rd and 4.th subfigure from the top respectively).



(b) The function $C^{spread}e^{-C^{decay}d}$ for $C^{decay} = \{4, 2, 1\}$ (solid, dashed and dotted lines respectively). Large (small) C^{decay} gives fast (slow) decay, and thus large (small) penalty $L^{spread}(\mathbf{d})$ for large (small) distances $\mathbf{d}$.

Figure 3.13: The upper subfigures show the selected points obtained using Equation (3.43) for $C^{spread} = 2u^{max}$ fixed and different values of C^{decay}. The bottom figure shows the corresponding functions $C^{spread}e^{-C^{decay}d}$. For $C^{decay} = 4$ the points $(\mathbf{x})^1, \ldots, (\mathbf{x})^S$ are spread in the region where $u()$ is high. Decreasing to $C^{decay} = 2$ gives slower decay of $e^{-C^{decay}d}$, and therefore a greater spread is encouraged. For $C^{decay} = 1$ the penalty $L^{spread}(\mathbf{d})$ is large for relatively large $\mathbf{d}$, thus the points are spread evenly in the domain $\Omega_x = [0, 15]$.

Recall the penalty functions L^{spread}, based on the indicator function and the exponential function, studied in the previous section. The latter was defined as

$$L^{spread}\left(\mathbf{d}\left(\mathbf{x}^1,\ldots,\mathbf{x}^S\right)\right) = C^{spread}\sum_{i=1}^{|\mathbf{d}|} e^{-C^{decay}d_i}. \tag{3.47}$$

where the parameter C^{spread} was chosen to be proportional to u^{max} defined in Equation (3.40). Let L_i^{spread} be defined as

$$\begin{aligned} L_i^{spread}\left(\mathbf{d}^i\left(\mathbf{x}^1,\ldots,\mathbf{x}^i\right)\right) &= C_i^{spread}\sum_{i=1}^{|\mathbf{d}|} e^{-C^{decay}d_i}, \\ C_i^{spread} \propto u_i^{max} &= \max_{\mathbf{x}\in\Omega_{\mathbf{x}}} \quad u_i\left(\mathbf{x};\mathbf{D}\right) \end{aligned} \tag{3.48}$$

where the function $L_i^{spread}\left(\mathbf{x}^1,\ldots,\mathbf{x}^i\right)$ depends on the number of selected points $(i-1)$ at the i'th iteration and C_i^{spread}. As discussed ib the previous section it may be reasonable to select a constant proportional to the maximum of the current infill function. Inserting L_i^{spread} into Equation (3.46) gives the following expression for the i'th selected point

$$\mathbf{x}^{n+i} = \operatorname{argmax} u^i\left(\mathbf{x};\mathbf{D}\right) - C_i^{spread}\sum_{i=1}^{|\mathbf{d}|} e^{-C^{decay}d_i}. \tag{3.49}$$

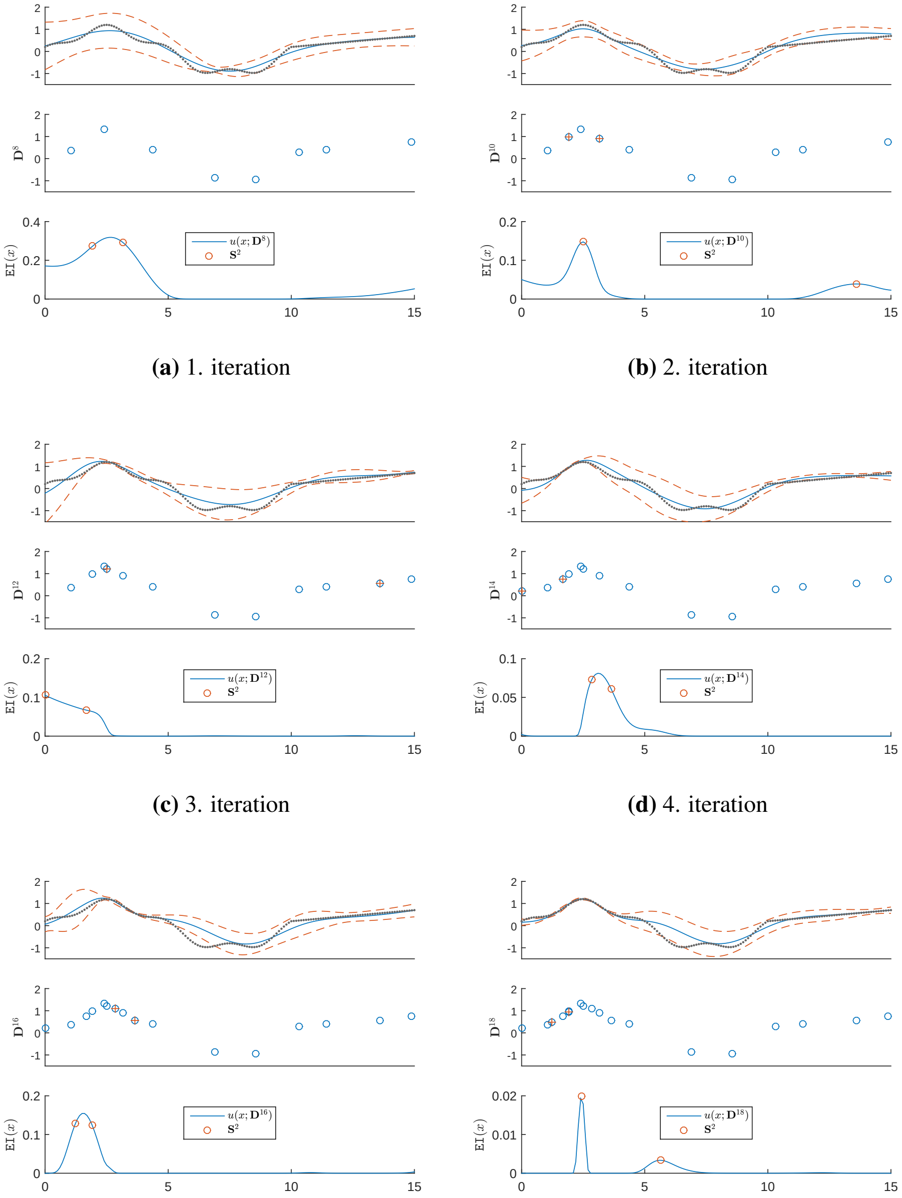
As it is difficult to analyse the performance of a single acquisition function in general, it is difficult to assess the performance of the proposed sequential approach for selecting $\mathbf{x}^1,\ldots,\mathbf{x}^S$. However, assuming that the computer intensive function f^{true} can be evaluated efficiently in parallel, the proposed method can be used to utilise parallel processing. The proposed sequential method is rather simple, and may be an adequate approach for selecting S design points with a reasonable spread.

3.4.3 Sample paths for synthetic data

In the proceeding, six iterations of the sequential optimisation method using S infill points selected jointly, as described in Section 3.4.1, are illustrated on the running example. The data samples $\mathbf{D}_{0.1}^{3,lhc}$ are used as initial sample. For each iteration, the current data samples $\mathbf{D}^n$, the prediction $f^n(x)$ $f^n(x) \pm \sigma^n(x)$ and the set of joint maximisers $\mathbf{S}^S$ are recorded. The latter quantity is of particular interest since it gives information of whether the joint maximisers of Equation (3.43) have a reasonable balance between maximisation of the univariate EI criterion and spread.

Figure 3.14 shows these aspects obtained by selecting $S = 2$ in each iteration. Note that for the second and sixth iteration, the two selected points are in the maxima of two different modes of the EI. For the first, third, fourth and fifth iteration, the EI criterion has one significant mode, and the two points are sampled from this mode with some spread. It seems that the $S = 2$ selected points have a reasonable balance between spread and maximisation of the EI criterion for all iterations.

Similarly, Figure 3.15 shows $f^n(x)$, $f^n(x) \pm \sigma^n(x)$, $\mathbf{D}^n$ and the $S = 3$ joint maximis-



(a) 1. iteration

(b) 2. iteration

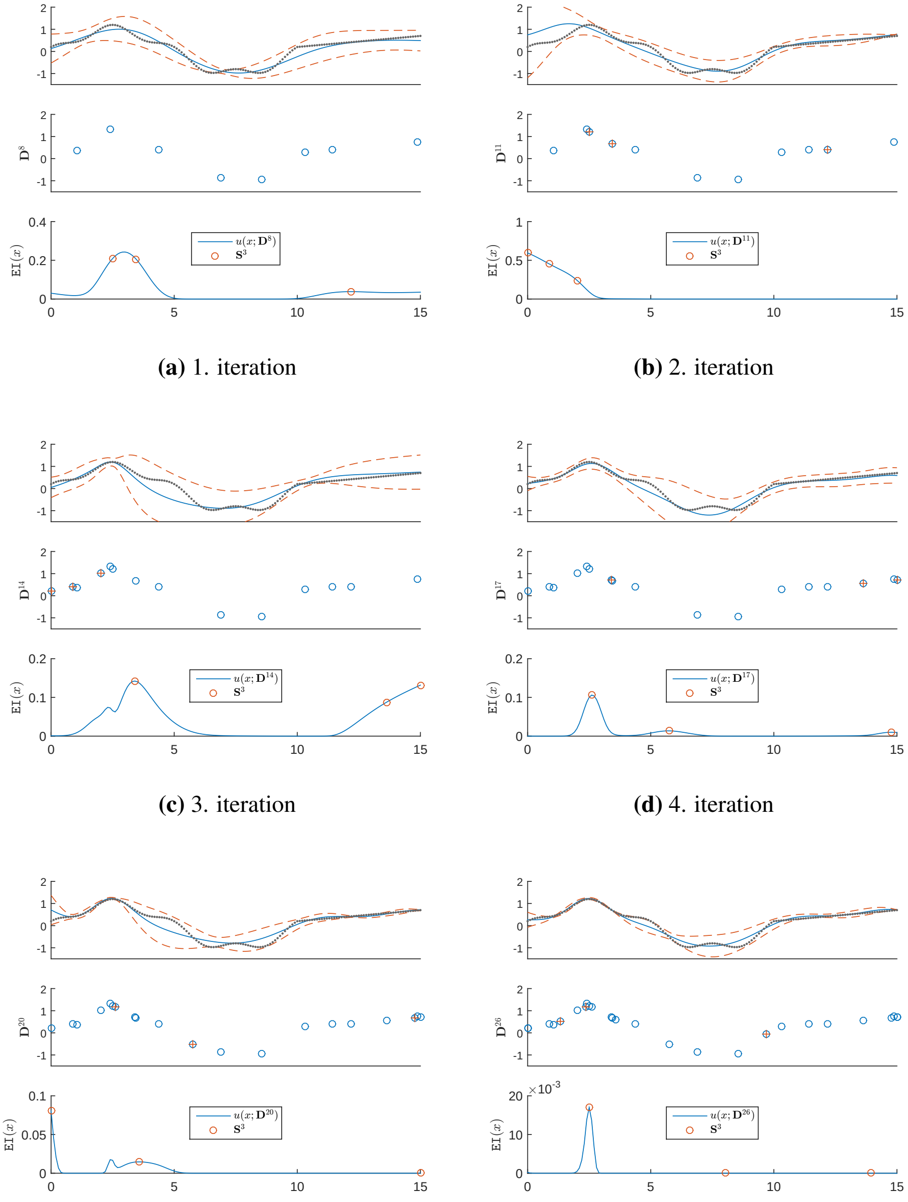
(c) 3. iteration

(d) 4. iteration

(e) 5. iteration

(f) 6. iteration

Figure 3.14: Upper subfigures: $f^n(x)$, $f^n(x) \pm \sigma^n(x)$ and $f^{true}(x)$ shown as solid, dashed and dotted lines respectively. Middle subfigures: observations $\mathbf{D}^n$ at iteration i. For $i > 1$, the previously S observed points are marked with a $+$ sign. Lower subfigure; the $\texttt{EI}$ infill function $u(x)$ and the set of joint maximisers $\mathbf{S}^S$ obtained by finding an approximate solution of Equation (3.43).



(a) 1. iteration

(b) 2. iteration

(c) 3. iteration

(d) 4. iteration

(e) 5. iteration

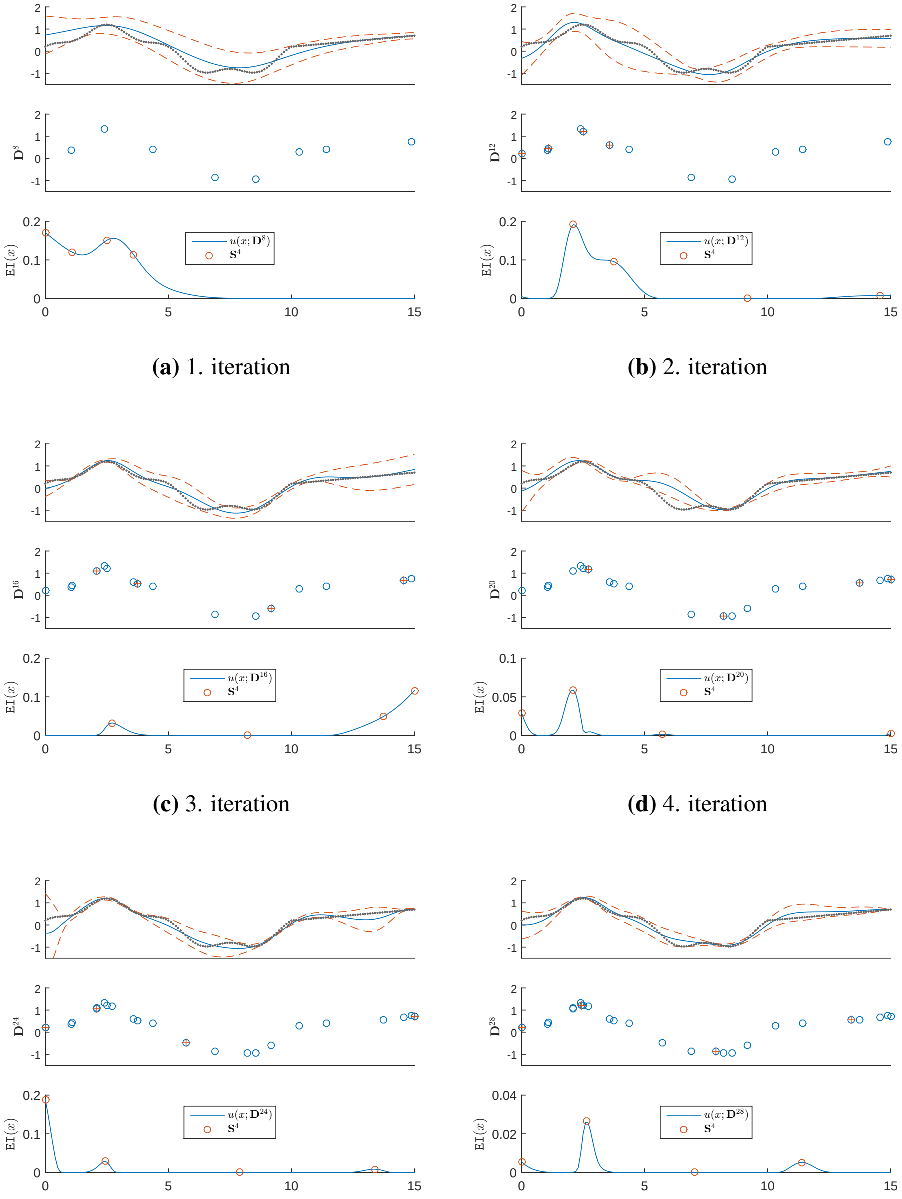
(f) 6. iteration

Figure 3.15: Upper subfigures: $f^n(x)$, $f^n(x) \pm \sigma^n(x)$ and $f^{true}(x)$ shown as solid, dashed and dotted lines respectively. Middle subfigures: observations $\mathbf{D}^n$ at iteration i. For $i > 1$, the previously S observed points are marked with a $+$ sign. Lower subfigure; the $\mathtt{EI}$ infill function $u(x)$ and the set of joint maximisers $\mathbf{S}^S$ obtained by finding an approximate solution of Equation (3.43).

ers for 6 iterations. Note that in the fourth iteration, the three joint maximisers are selected from three different modes of the EI criterion. In the first and third iteration the points are selected from two different modes. Note that in the sixth iteration, only one point is selected from the single mode. This mode is very spiky, hence the penalty of sampling more than one point in this mode is large due to the small pairwise distances. For all iterations the selected points seem to have a reasonable degree of spread.

The same quantities are displayed in Figure 3.16 using $S = 4$ infill points in each of the 6 iterations. In the first iteration, the four joint maximisers are in the leftmost region. Note that for all remaining iterations, the maximisers of the modes are always in the set of joint maximisers.

As is demonstrated for $S = \{2, 3, 4\}$, the sets of joint maximisers $\mathbf{S}$ have a reasonable balance between maximisation of the EI criterion and pairwise distances. This indicates that the multipoint criterion defined in Equation (3.43) is suitable for selecting $S > 1$ infill points.



(a) 1. iteration

(b) 2. iteration

(c) 3. iteration

(d) 4. iteration

(e) 5. iteration

(f) 6. iteration

Figure 3.16: Upper subfigures: $f^n(x)$, $f^n(x) \pm \sigma^n(x)$ and $f^{true}(x)$ shown as solid, dashed and dotted lines respectively. Middle subfigures: observations $\mathbf{D}^n$ at iteration i. For $i > 1$, the previously S observed points are marked with a $+$ sign. Lower subfigure; the $\mathtt{EI}$ infill function $u(x)$ and the set of joint maximisers $\mathbf{S}^S$ obtained by finding an approximate solution of Equation (3.43).

3.5 Constrained optimisation

A simulation model may contain constraints on the input $\mathbf{x}$. It may require simulations in order to evaluate if the input satisfy the constraints. Such (unknown) constraints poses a challenge for surrogate based optimisation, and we refer to Schonlau et al. (1998); Gelbart et al. (2014); Williams et al. (2010); Lindberg and Lee (2015); Gramacy et al. (2016) for different approaches for incorporating constraints in surrogate based optimisation methods. In this section, two methods for handling constraints are studied. The method introduced in 3.5.1 is based on the work by Schonlau et al. (1998), and requires an additional response variable to model the probability that a constraint is satisfied. The method proposed in section 3.5.2 use a penalty term for constraint violation, and does not require modelling of additional output variables.

3.5.1 Constrained expected improvement

Following the approach in Schonlau et al. (1998), let $c(\mathbf{x})$ denote a constraint on an additional response variable Y_c. Let the improvement function $\mathtt{I}_c^{\rho}(\mathbf{x})$, subjected to $c(\mathbf{x})$, be defined as

$$\mathtt{I}_c^{\rho}(\mathbf{x}) = \begin{cases} (y(\mathbf{x}) - y^{+})^{\rho}, & \text{if } y(\mathbf{x}) > y^{+} \text{ and } a \leq c(\mathbf{x}) \leq b \\ 0, & \text{otherwise,} \end{cases} \tag{3.50}$$

i.e. the improvement $\mathtt{I}_c^{\rho}$ are non-zero if and only if $y(\mathbf{x}) > y^{+}$ and the variable $y_c(\mathbf{x})$ satisfy the constraint $c(\mathbf{x})$. Let $Y^1(\mathbf{x})$ and $Y^2(\mathbf{x})$ denote the two response variables under consideration. By assuming that two variables Y and Y_c are statistical independent, the expectation of Equation (3.50) can be written as

$$\mathtt{E}\left[\mathtt{I}_c^{\rho}(\mathbf{x})\right] = \mathtt{E}\left[\mathtt{I}^{\rho}(\mathbf{x})\right] \mathtt{P}\left(a \leq c(\mathbf{x}) \leq b\right) \tag{3.51}$$

The above equation is a scaled version of $\mathtt{E}\left[\mathtt{I}^{\rho}(\mathbf{x})\right]$. By estimating the probability of constraint satisfaction $\mathtt{P}\left(a \leq c(\mathbf{x}) \leq b\right)$, the constrained expected improvement for $\mathtt{E}\left[\mathtt{I}_c^{\rho}\right]$ can be obtained by simple multiplication of the expected improvement $\mathtt{E}\left[\mathtt{I}^{\rho}\right]$.

In the proceeding, the constrained expected improvement is illustrated for the ANNs $\Upsilon(x)$ shown in Figure 3.1. For simplicity, assume $\rho = 1$ and that the probability of constraint satisfaction is given by

$$\mathtt{P}\left(a \leq c(x) \leq b\right) = \frac{2x}{15^2}, \tag{3.52}$$

i.e. it increases linearly from zero to one on the domain $x \in [0, 15]$. Thus, $\mathtt{E}\left[\mathtt{I}_c^1(\mathbf{x})\right]$ can be easily obtained by inserting for $\mathtt{E}\left[\mathtt{I}^1(\mathbf{x})\right]$ and $\mathtt{P}\left(a \leq c(x) \leq b\right)$ into Equation (3.51). $\mathtt{E}\left[\mathtt{I}^1(\mathbf{x})\right]$ is given by Equation (3.22) and Equation (3.26) for CDF based on the emprical CDF and an assumed normal distribution respectively. Figure 3.17. shows $\mathtt{E}\left[\mathtt{I}_c^1(\mathbf{x})\right]$ computed for the two CDFs and the constraint in Equation (3.52). Comparing the unconstrained and constrained expected improvement, the former shown in Figure 3.5, it is clear that $\mathtt{P}\left(a \leq c(x) \leq b\right)$ encourages exploration of x with high probability for constraint satisfaction.

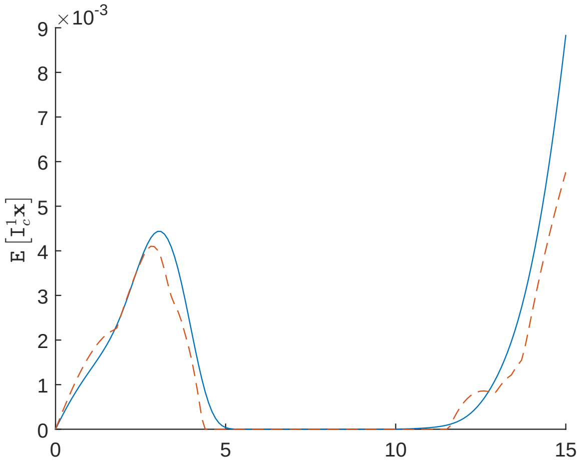


Figure 3.17: The expectation of the constrained improvement $\mathbb{E}\left[\mathbb{I}_c\left(\mathbf{x}\right)\right]$ computed using $\hat{F}^{norm}_{Y(\mathbf{x})}(y)$ (solid line) and $\hat{F}_{Y(\mathbf{x})}(y)$ (dotted line) and $\mathbb{P}\left(a \leq c(x) \leq b\right)$ from Equation (3.52). The latter is small (large) for x low (high). Note that $\mathbb{E}\left[\mathbb{I}_c\left(\mathbf{x}\right)\right]$ is high where the constraint $c\left(\mathbf{x}\right)$, defined in Equation (3.52), is high.

3.5.2 Penalty method

The approach proposed in this section utilise a penalty function $L^c\left(\mathbf{x}\right)$ to adjust the simulation output $f^{true}\left(\mathbf{x}\right)$. An appealing feature of this approach is that it does not require modelling the probability that a strategy $\mathbf{x}$ satisfy $c\left(\mathbf{x}\right)$, as opposed to the method discussed in the previous section. The rationale for including the term $L^c\left(\mathbf{x}\right)$ is that it can be used to reflect a penalty for constraint violation. Thus, optimisation of the simulation model will tend to avoid regions where $c(\cdot)$ is not satisfied. In the following, the method is explained for a generic maximisation problem and penalty function $L^c\left(\mathbf{x}\right)$.

Let $f^{true}\left(\mathbf{x}\right)$ denote the function to be maximised. Let $c\left(\mathbf{x}\right)$ denote the unknown constraint available through evaluation of $f^{true}\left(\mathbf{x}\right)$. Let $L^c\left(\mathbf{x}\right) \geq 0$ denote a function that penalises violation of $c\left(\mathbf{x}\right)$, and assume high (low) degree of constraint violation corresponds to high (low) values of $L^c\left(\mathbf{x}\right)$. Define the adjusted observations y^i of $\mathbf{x}^i$ as

$$f^{adj}\left(\mathbf{x}^i\right) = f^{true}\left(\mathbf{x}^i\right) - L^c\left(\mathbf{x}^i\right). \tag{3.53}$$

The above equation is high when $f^{true}\left(\mathbf{x}\right)$ is high and $L^c(\mathbf{x})$ is low, and the maximiser is the point with best balance between $f^{true}\left(\mathbf{x}^i\right)$ and $L^c\left(\mathbf{x}^i\right)$. This approach is used to adjust the output for a simulation model in Section 4.1.2.

Chapter 4

Case study on NOWIcob simulation model

In this chapter, the surrogate based optimisation method is tested on a real-world problem that require evaluation of a time-consuming simulation model. The problem under consideration is to reduce the costs related to operation and maintenance (O&M) tasks for offshore wind farms. These costs constitute about one third of the overall cost during the lifetime of a wind farm. Thus, finding good strategies for performing the O&M tasks are essential. In practice it is infeasible, or at best extremely costly, to evaluate the performance of strategies by physical experiments. There are developed several simulation models that mimic the logistics related to O&M tasks. The sequential optimisation method is utilised on such a model called NOWIcob. The aim is to identify combinations of vessel fleet and personnel that gives low O&M cost. An introduction to O&M operations is given in Section 4.1, and the decision problem is described more in detail in Section 4.2.

4.1 Introduction

SINTEF has developed a simulation model called NOWIcob (**N**orwegian **o**ffshore **wi**nd power life cycle **c**ost and benefit model). The simulation model mimics the daily operations on a wind farm in order to evaluate the performance of different O&M strategies. In this section, main features of the model are described. See Hofmann et al. (2014) for a more in depth description of this simulation model, and Hofmann (2011) for a review of similar simulation models.

One of the key features of the simulation model is that different types of turbine-failure occurs. These failures cause the windmill to stop or reduce the produced energy until they are fixed. Failures may require technicians, spare parts and vessels with special abilities in order to be restored. Since the wind farm is offshore, the vessels have to access the wind farm in order to fix the failures. The different vessels have limits that restricts what kind of weather they can be operated in, and if the weather is too rough the repair

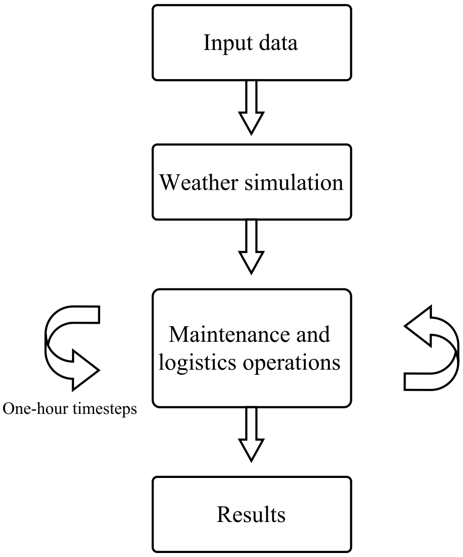


Figure 4.1: Simplified flow scheme of the NOWIcob simulation model, adopted from a similar illustration in Hofmann et al. (2014). The wind farm and the related O&M strategy $\mathbf{x}$ are specified by the input data. The weather can be either historical records or synthetic data generated from such records. Operations related to maintenance and logistics are simulated with one hour resolution. The results contain many output variables. The profit $\pi(\mathbf{x})$ is used throughout this project.

must be rescheduled. The simulation model distinguish between *corrective* and *preventive* maintenance tasks. The former type refers to maintenance tasks related to failures that randomly occurs. The failure times are assumed to follow a homogeneous Poisson process with an average yearly rate. This rate can be different for different years, which can be used to model for example a bathtub curve with higher intensities the first few and last few years. The latter type is time-based maintenance tasks that have to be performed at predetermined time steps. For convenience, such tasks are modelled as one annual service task that occurs each year for each turbine. The setup of the NOWIcob model used in this project does not penalise O&M strategies that are not able to complete all annual service tasks. This is discussed more in detail in Section 4.1.2 due to the implications this feature has on the sequential optimisation method. A more in depth discussion of the failures and related logistics can be found in Hofmann et al. (2014).

The user of the model can specify many input parameters that together describe all aspects that are used by the simulation model. Figure 4.1 illustrates the steps in the simulation model. The weather can be chosen as historical records, or synthetically weather generated from such records by using a Markov chain process. The power curve of the turbines depends on the wind speed, and vessels may have weather limits that restricts the time they can be used.

In the proceeding, let $\mathbf{x}$ denote all input parameters. By running the simulation model for particular values of the input parameters $\mathbf{x}$ a resulting output y is produced, which is stochastic if either failure times or the weather is treated as stochastic. The failure times

Table 4.1: Explanation of some important input and output parameters.

Input parameter	Definition
Weather	Time series for wave height and wind speed are generated from historical weather from the wind farm location.
Turbine type	Properties as power curve, physical dimensions, cut-in and cut-off speeds differs for different types.
Turbine number	The total number of turbines at the wind farm
Distance to location	The shortest distance from the wind farm to the location(s) with personnel accommodation.
Simulation horizon	The simulated lifetime of the wind farm
Personnel available	The average number of maintenance or technician personnel available each shift. These are stationed at an onshore or offshore location.
Failure type	The different failure types have different consequences. They may partly or fully reduce the turbines ability to produce energy, or they can be annual services in order to prevent future failures.
Rate	The different failures types are assumed to occur randomly with some intensity. Maintenance tasks that are performed regularly, as annual service, can be performed at predetermined dates.
Total direct O&M cost	The sum of all costs related to vessels, repair, personnel and location.
Total energy production	Takes into account availability, loss and downtime.

refers to the times when corrective failures occurs.

Some examples of different types of input $\mathbf{x}$ and output y are shown in Table 4.1. A reader unfamiliar with wind farms and simulation models similar to NOWIcob, may ease the further reading by getting familiar with the input and output explanations in the mentioned table. Other parameter are listed in Table 6.2, 6.3, 6.4 and 6.5 in Appendix A. A throughout descriptions of all input, output and assumptions used in the simulation model is outside the scope of this paper, and we refer to Hofmann et al. (2014) for a more detailed description.

Before proceeding with demonstrations of the simulation model, a more mathematical language is adopted for describing the relation between different wind farms with specific O&M strategies and their performance.

Let the vector $[\mathbf{x}', \mathbf{x}^-]$, where $\mathbf{x}' = (x_1, \ldots, x_P)^\top$ and $\mathbf{x}^- = (x_{P+1}, \ldots, x_{\tilde{P}})^\top$ contain all input parameters necessary to describe any wind farm with specific O&M strategies. The two vectors $\mathbf{x}'$ and $\mathbf{x}^-$ are a partition of the input parameters categorised by whether a user of the model finds the parameters of interest or not. The former type of variables are referred to as *decision variables*. The nuisance parameters are assumed to be fixed. In order to ease the readability, we may adopt notation that excludes the fixed parameters. Hence, let $\mathbf{x} = [\mathbf{x}'; \mathbf{x}^-] = [x_1, \ldots, x_p; \mathbf{x}^-]$ denote the values of the P varying parameters, given the value $\mathbf{x}^-$ for the fixed parameters.

Moreover, let the associated performance for a specific choice of variables be denoted by $y(\mathbf{x})$. With this notation, we can model the stochastic simulation model as

$$y(\mathbf{x}) = f^{true}(\mathbf{x}) + \epsilon(\mathbf{x}) \tag{4.1}$$

where $f^{true}(\mathbf{x})$ may be interpreted as the *true relation* and $\epsilon(\mathbf{x})$ as noise due to the randomness in weather and failure times. If both weather and failure time are deterministic, the noise term could be omitted. However, this approach would give information on how strategies perform for a particular weather time series with known failure times, and hence may be disadvantageous since it fails to favour strategies with overall good performance for different realisations of the assumed random weather and failure times. In some cases, it may be of interest to use deterministic weather, either historical weather or the same synthetic weather, and/or let the failure times be the same for all simulations with the same input. Before illustrating the effect of the stochastic aspects, the different outputs of the simulation model are introduced.

In Banks (1998) several advantages of simulation models are mentioned, and most of them concern the ability to learn about the underlying system, in this case a wind farm and its related operations. A decision maker may learn some characteristics of different O&M strategies, and be enabled to answer what-if questions. The NOWIcob simulation model produces a lot of output that may help the user draw inference. Some of these outputs are listed in Table 6.5 in Appendix A. Two of these outputs, namely O&M cost and produced amount of energy should ideally be low and high respectively. Let the profit $\pi(\mathbf{x})$ represent the users desired trade off between the two aspects, and be defined here as

$$\pi(\mathbf{x}) = I^{\text{ener}}(\mathbf{x}) - C^{\text{OM}}(\mathbf{x}) - C^{inve} \tag{4.2}$$

where $C^{OM}(\mathbf{x})$ and $I^{\text{ener}}(\mathbf{x})$ denotes the total cost and income from energy production for a particular O&M strategy $\mathbf{x}$ over the whole lifetime, and C^{inve} denotes the investment cost of the wind farm. Unless stated otherwise, it is assumed that $C^{inve} = 0$ since a fixed investment cost does not affect maximisation of Equation (4.2). The income from energy production $I^{\text{ener}}(\mathbf{x})$ implicitly assumes some (future) price scenario for electricity. A constant price scenario is assumed for simplicity. The overall focus is on maximisation of the profit $\pi(\mathbf{x})$, which can be calculated from the two outputs $C^{OM}(\mathbf{x})$ and $I^{\text{ener}}(\mathbf{x})$ by insertion into Equation (4.2). Results from the simulation model may indicate if a given O&M strategy has high associated O&M cost, produced energy and profit. However, due to stochastic weather and failure times, it may be hard to draw conclusions since the promising (poor) results may be caused by favourable (unfavourable) realisations of the stochastic components. The effect of the stochastic failure times and the (possible) stochastic weather on the profit $\pi(\mathbf{x})$ may be assessed by performing several simulation with the same input $\mathbf{x}$. Figure 4.2 shows two boxplots of $\pi(\mathbf{x})$ obtained by performing 8 simulations for a strategy $\mathbf{x}$, with deterministic and stochastic weather respectively. Note that the variability of $\pi(\mathbf{x})$ is smaller (higher) when the weather is deterministic (stochastic). The reason is that some weather time series have calm (harsh) weather such that maintenance tasks are less (more) restricted to the wind speed and weather height. Moreover, the turbines potential energy production depends on the wind speed, which also effect

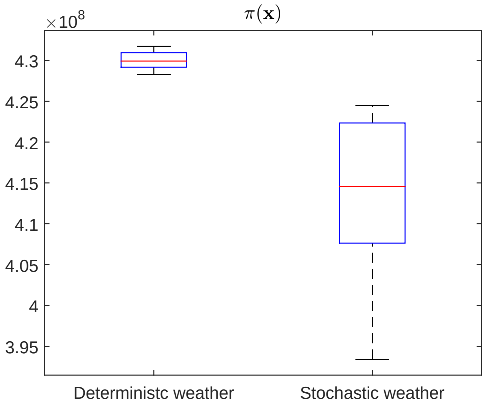


Figure 4.2: The left (right) figure shows boxplots of the profit $\pi(\mathbf{x})$ obtained through 8 repeated simulations using deterministic (stochastic) weather time series and the same strategy $\mathbf{x}$. The simulation horizon is 5 years, and the failure times are stochastic. The deterministic weather is a sequence of historic records, whereas the stochastic weather sequences are (unique) synthetically generated weather sequences. Using deterministic (stochastic) weather gives low (high) variability for the profit $\pi(\mathbf{x})$.

the amount of potential energy that can be harvested. The effect of stochastic weather is discussed more in detail in Section 4.1.1.

When comparing the performance of different strategies, it is not obvious which strategy that is best due to the stochastic output. An approach may be to perform several simulations for each input to reduce the variability. This is time consuming, so different kinds of analysis could be applied in order to reduce the need of many simulations. In the master thesis of Hagen (2013) sensitivity analysis was performed for analysing the results from the NOWIcob simulation model, and is an integrated part of this model. Such analysis aims at answering how the output is affected by changes in one or more of the input parameters. A surrogate model can be used to approximate $f^{true}(\mathbf{x})$ in Equation (4.1). If the surrogate model is accurate, we may obtain knowledge of how changes in one or several input parameters affect the output.

Some input parameters have a trivial effect on the output, so processing of previously simulations could be used to infer what the output would be for changes in these inputs. Examples of such parameters are all cost parameters. Parameters that do not have a trivial relation are those which affect the daily operations or the electrical availability of the wind farm, like the failure rates, prioritisation of maintenance tasks, vessel fleet, weather etc. In this project we will focus on the latter type of parameters. Parameters that either have a known and easy relation to the output, or a negligible effect, are given less consideration than parameters with unknown and great effect on the output. The reason is to avoid problems related to the ”curse of dimensionality”. The term refers to problems that arises when the dimensionality increases and computer demands exceed what is feasible.

Some other assumptions that may affect the analysis in this project is discussed in Appendix A.

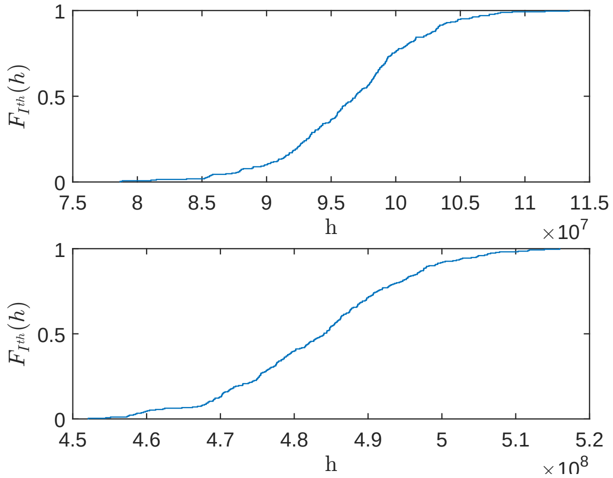


Figure 4.3: The empirical CDF of $I^{theo}(w)$ calculated from $w_1, \ldots, w_{270}$ weather time series over 1 (upper subfigure) and 5 (lower subfigure) years. Note the difference in x-axis: the relative variability of $I^{theo}(w)$ is higher for the short time series compared to the longer time series. This illustrates that more stable results may be obtained by increasing the simulation horizon. Longer simulation horizons corresponds to more time-consuming simulations, hence the user may balance stability and time consumption.

4.1.1 Maximal theoretical income

The variability increases if different weather time series w is used in the different simulations. Some weather time series may have more harsh weather conditions compared to other time series, such that maintenance tasks are more difficult to perform. However, this effect depends on the strategy $\mathbf{x}$ since some strategies are less sensitive to weather than others. In the following, a quantity that is useful for understanding the effect of weather on the simulation output is discussed. Define the maximum theoretical income $I^{theo}(w)$ as the income if all turbines were working all the time for weather time series w . This measure of the potential of the wind farm is independent of the strategy $\mathbf{x}$, and is useful for explaining the variability of the simulation output. $I^{theo}(w)$ is one of the many output parameters of the NOWIcob model, and it is calculated by taking into account the wind speed, power curve of the turbines and losses due to wake effect and electrical infrastructure (Hofmann et al., 2014).

Figure 4.3 shows the empirical CDFs $F_{I^{theo}(w)}$ obtained by computing $I^{theo}(w)$ for 270 weather time series $w_1, \ldots, w_{270}$ over a period of one and five years. From the figure, it is clear that $F_{I^{theo}(w)}$ has more spread for simulations performed over one-year than the alternative. This illustrates that the more stable output from the simulation model may be achieved by increasing the simulation horizon. The time consumption increases for increasing simulation horizon. By specifying the simulation horizon, the user can balance a simulations time consumption and the outputs variability.

4.1.2 Annual service constraint

Annual service tasks are a type of maintenance tasks that must be performed for each turbine every year. In this section, the sequential method is extended to handle a constraint related to the number of such tasks that has to be completed. It is reasonable to require that only strategies with a high degree of completed annual service tasks are valid strategies. Evaluating if a strategy satisfy this constraint may only be assessed through simulation. Annual service require technicians assessing the turbines, spare parts and shut-down of the turbine during work, see Appendix 6.7. The NOWIcob does not penalise strategies that fails to complete the service tasks.

The first few runs of the sequential method optimised only the profit, and the method converged to strategies that had high profit due to low O&M cost and high amount of produced energy. More careful analysis of the simulation output for these strategies showed that the amount of completed annual service tasks were too low. Such (unknown) constraints poses a challenge for surrogate based optimisation, and we refer to Schonlau et al. (1998); Gelbart et al. (2014); Williams et al. (2010); Lindberg and Lee (2015); Gramacy et al. (2016) for different approaches for incorporating constraints in surrogate based optimisation methods. In the proceeding, we propose an approach for handling the annual service constraint by adjusting the profit for strategies that fail to complete the annual services. The motivation for this approach is that it is simpler than the above-mentioned approaches, and such penalty methods are commonly used in different optimisation methods.

Let $T^{as}\left(\mathbf{x}\right) \in [0,1]$ denote the fraction of annual service tasks that are completed for strategy $\mathbf{x}$. Since strategies are only valid if most annual service tasks are completed, it is of interest that T^{as} is exactly, or very close to, 1. Define an adjusted profit measure $\pi^{\star}\left(\mathbf{x}\right)$ as

$$\pi^{\star}\left(\mathbf{x}\right) = \pi\left(\mathbf{x}\right) - L^{as}\left(\mathbf{x}\right) \tag{4.3}$$

where $L^{as}\left(\mathbf{x}\right)$ is a loss-function for constraint violation. A thorough discussion of the effect of constraint violation is outside the scope of this project. However, since it is of interest to consider strategies with T^{as} high, it may suffices with any penalty function T^{as} that encourages exploration of strategies where T^{as} is likely to be high. For simplicity, assume that $L^{as}\left(\mathbf{x}\right)$ depends linearly on $T^{as}\left(\mathbf{x}\right)$ as

$$L^{as}\left(\mathbf{x}\right) = C^{as}\left(1 - T^{as}\left(\mathbf{x}\right)\right) \tag{4.4}$$

where $C^{as} > 0$ is a constant that ensure a sufficiently high penalty for service tasks not completed. Note that if all annual service tasks are completed, i.e. $T^{as} = 1$, Equation (4.4) is 0. From Equation (4.4), it is clear that by multiplying the constant C^{as} with a factor, corresponds to multiplying $L^{as}\left(\mathbf{x}\right)$ with the same factor. The value of C is set such that the penalty of not completing a service task, is higher than an estimate of the total cost of completing the task. In the following, the profit $\pi\left(\mathbf{x}\right)$ and the adjusted $\pi^{\star}\left(\mathbf{x}\right)$ are computed for an example case with 30 different strategies. Figure 4.4 illustrates that the profit for strategies with low $T^{as}\left(\mathbf{x}\right)$ are adjusted by a penalty $L^{as}\left(\mathbf{x}\right)$. A higher (lower) value of the constant C^{as} results in a higher (lower) penalty. The magnitude of the penalty seem reasonable. A low penalty may results in a sequential method that converges towards an optimal strategy $\mathbf{x}$ where $T^{as}\left(\mathbf{x}\right) < 1$. A too high penalty may cause greater variability

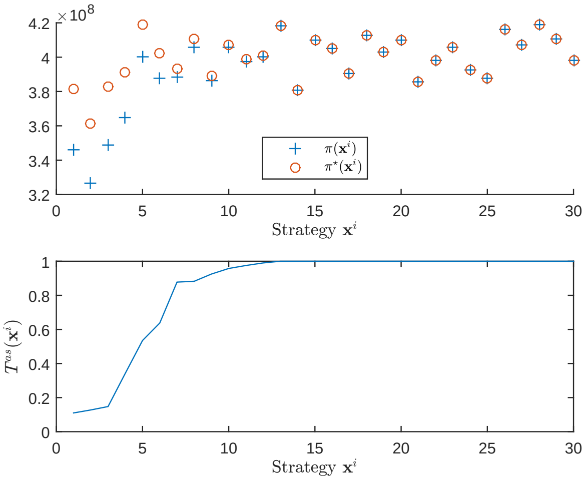


Figure 4.4: The upper subfigure shows $\pi(\mathbf{x})$ ('o') and $\pi^\star(\mathbf{x})$ ('+') obtained through simulation of strategies $\mathbf{x}^1, \ldots, \mathbf{x}^{30}$ and by applying equations (4.2) and (4.3) respectively. The lower subfigure shows the corresponding values of $T^{as}(\mathbf{x}^i)$ for $i = 1, \ldots, 30$. The strategies $\mathbf{x}^i$ represent different choices of number of personnel and vessels, and the constant C^{as} corresponds to a penalty of 100.000 GBP per annual service that is not completed. Note especially that the fifth strategy has very high profit $\pi(\mathbf{x})$, whereas the adjusted profit $\pi^\star(\mathbf{x})$ is lower since only 53.5% percent of the service tasks are completed. The strategies $\mathbf{x}^1, \ldots, \mathbf{x}^{n_0}$ are generated by using a Latin Hypercube design for the three variables CTV, SES and technicians

Table 4.2: Parameters related to personnel and the vessel types CTV and SES. Both vessels have capacity of transporting 12 personnel. They can operate one shift before returning to the harbour. Fuel consumption is neglected. The SES vessels is faster, more robust and more expensive than the alternative. The set of feasible strategies $\Omega_{\mathbf{x}}$ is all possible combinations of x_1, x_2 and x_3.

	CTV	SES	Personnel
Fixed cost [GBP/year]	638.750	1875.000	80.000
Speed [knots]	20	30	-
Weather limits access (wave heigh, wind speed)	1.5 m, 16 m/s	2 m, 16m /s	-
Feasible range	$[0, 1, \ldots, 4]$	$[0, 1, \ldots, 4]$	$[6, 7, \ldots, 50]$

in the data samples since the magnitude of the stochastic term $L^{as}(\mathbf{x})$ increases. In the proceeding, the maximisation of the simulation model output is done with respect to $\pi^\star(\mathbf{x})$ from Equation (4.3), with $C^{as} = 100.000$.

4.2 Vessel fleet optimisation-problem

In this section, the sequential method is tested on a problem of selecting a combination of Crew Transfer Vessels (CTVs) and technicians. CTVs are used to transport personnel that can perform maintenance tasks. Different vessel types may have different abilities as speed, capacity weather restrictions etc. The weather affects how and when the vessels can operate. Therefore, it is not straightforward to analyse which fleet mix gives the best balance between O&M cost and produced amount of energy. Different approaches have been applied to analyse problems related to vessel fleet selecting. For example, a simulation model is in Dalgic et al. (2014) used to study the decision problem of selecting a combination of two types of CTVs. A similar problem is studied in Tande et al. (2013) and Gundegjerde et al. (2015) by applying deterministic and stochastic mathematical optimisation respectively.

The above-mentioned approaches show promising results for selecting combinations of several vessels, helicopters etc. In this project, we extend the decision problem by adding workforce as a decision variable. The sub problems of selecting vessel fleet and work force are highly connected, and it is therefore of interest to consider them as a single decision problem. In the following, the decision problem is formulated in a more precise manner. Let

$$\mathbf{x} = (x_{CTV}, x_{SES}, x_{PER}) \tag{4.5}$$

denote a strategy represented by the number of CTVs, Surface Effects Ships (SES) and personnel respectively. The vessel types CTV and SES can be used to transport personnel, and the SES is defined as a more robust and expensive version of the CTV. Costs and other related parameters for the decision variables are listed in Table 4.2. The objective is to find $\mathbf{x}^{opt}$ that maximises the adjusted profit $\pi^\star(\cdot)$, defined in Equation (4.3), i.e.

$$\mathbf{x}^{opt} = \underset{\mathbf{x} \in \Omega_{\mathbf{x}}}{\operatorname{argmax}} \, \pi^\star(\mathbf{x}) \tag{4.6}$$

Table 4.3: Explanation of the three instances of the VFO optimisation case. The main difference is whether the same (historical) weather time series is used or unique (synthetic) time series are used for the different simulations. All other parameters used by the simulation model is equal for the three cases. The number of iteration of the sequential method n^{iter} is low (high) depending on degree of variability in output.

	Case 1	Case 2	Case 3
Lifetime	1 year	5 years	5 years
Weather	Deterministic	Deterministic	Stochastic
Failure times	Deterministic	Stochastic	Stochastic
n^{iter}	40	60	80
Simulation characteristics	Rapid simulations, deterministic output	Slow, stochastic output with moderate variability	Slow, stochastic output with high variability
Interpretation: Modelling of O&M tasks and related logistics over...	1 year with the same (historical) weather time series and equal failure times for replicates of the same strategy	5 years with the same (historical) weather time series and different failure times for replicates of the same strategy	5 years with unique (synthetic) weather time series and different failure times for replicates of the same strategy

where $\Omega_{\mathbf{x}}$ denotes the set of feasible strategies. For notational ease, let $f^{true}(\mathbf{x})$ denote the observed adjusted profit $\pi^{\star}(\mathbf{x})$ for strategy $\mathbf{x}$. It is of particular interest to gain knowledge of the sequential methods performance on problems with different amount of noise. Therefore, the method is tested on three instances of the VFO problem with different degree of variability in the simulation output. The instances are referred to as case1, case2 and case3, and a description of each can be found in Table 4.3. The cases differ by whether the failure times and/or the weather are treated as deterministic or stochastic. The simulation output for case 1 is noise-free, and case 2 and case 3 has moderate and high degree of noise respectively. In the proceeding section, the sequential method is used to optimise the simulation output $\pi^{\star}(\cdot)$ for the three cases.

.

4.3 Optimisation algorithm for the VFO problem

The sequential method utilised for optimising the different cases of the VFO problem is summarised in Algorithm 5 . The initial design points $\mathbf{x}^1, \ldots, \mathbf{x}^{n_0}$ are selected by a Latin Hypercube sampling procedure, see Santner et al. (2003); Forrester et al. (2008) for thorough discussion of such space filling procedures. The same set of parameters are used for the different cases, except the number of iterations n^{iter}. This parameter is used as stopping criteria, and is set to $40, 60$ and 80 for case 1, case 2 and case 3 respectively.

Algorithm 5 Optimisation by sequential design

```
Define:
    wind farm with the fixed parameters $\mathbf{x}^{-}$
    decision variables $\mathbf{x} = (x_{CTV}, x_{SES}, x_{PERS})$
    feasible domain $\Omega_{\mathbf{x}}$
Specify surrogate model parameters:
    number of ANNs $B = 10$
Specify sampling procedure:
    number of infill points $S = 4$
    joint infill function $u\left(\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}\right)$ defined by Equation (3.43)
      with univariate infill function $u\left(\mathbf{x}\right) = \mathtt{EI}\left(\mathbf{x}\right)$
Obtain initial sample
    initial sample size $n_0 = 30$
    Select $\mathbf{x}^1, \ldots, \mathbf{x}^{n_0}$ by Latin Hypercube sampling
    Evaluate $y^j = f^{true}(\mathbf{x}^j)$ in parallel for $j = 1, \ldots, n_0$
    $\mathbf{D}^n = \{(\mathbf{x}^j, y^j)\}_{j=1}^{n_0}$
Iteration $i = 1$
repeat
    Fit surrogate model $f^n\left(\mathbf{x}\right)$ to $\mathbf{D}^n$
    Find $\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S} = \mathrm{argmax}_{\mathbf{x} \in \Omega_{\mathbf{x}}}\, u\left(\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+S}\right)$
    Sample $y^{n+j} = f(\mathbf{x}^{n+j})$ in parallel for $j = 1, \ldots, S$
    Augment $\mathbf{D}^n = \{\mathbf{D}^n, \left(\mathbf{x}^{n+1}, y^{n+1}\right), \ldots, \left(\mathbf{x}^{n+S}, y^{n+S}\right)\}$
    Record:
        $\mathbf{x}^{max,i} = \mathrm{argmax}_{\mathbf{x} \in \Omega_{\mathbf{x}}}\, f^n\left(\mathbf{x}\right)$
        $u^{max,i} = \mathrm{argmax}_{\mathbf{x} \in \Omega_{\mathbf{x}}}\, u\left(\mathbf{x}\right)$
    $i = i + 1$
until $i > n^{iter}$
return $\mathbf{D}^n$                                   ▷ Data samples used for further analysis.
```

Chapter 5

Experimental results

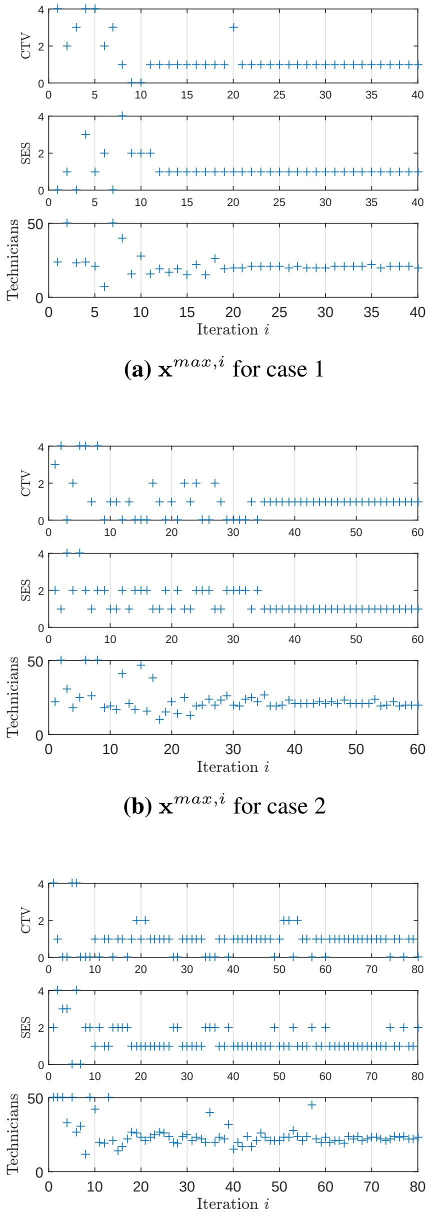
In this chapter, the sequential optimisation procedure presented in section 4.3 is used to optimise the three instances of the VFO problem. It is of interest both to assess aspects related to the sequential optimisation method and the results available after the method has converged. These two aspects are studied in Section 5.1 and Section 5.2 respectively.

5.1 The optimisation process

Since the surrogate model is tested on problems without a known optimal solution, it can be difficult to measure the performance. In order to better understand how the surrogate model learns which strategies that are favourable, the predicted maximiser $\mathbf{x}^{max,i}$ is recorded at each iteration $i = 1, \ldots, n^{iter}$. The predicted maximisers $\mathbf{x}^{max,i}$ is interesting since it shows which strategies the surrogate model $f^n(\mathbf{x})$ predicts is the best at iteration i. Figure 5.1 shows $\mathbf{x}^{max,i}$ for $i = 1, \ldots, n^{iter}$ for the three cases. From the figure, it can be seen that the $\mathbf{x}^{max,i}$ is very unstable the first few iterations i. This indicates that the surrogate model $f^n(\mathbf{x})$ changes significantly between each iteration. As the number of iterations increases, $\mathbf{x}^{max,i}$ stabilises at some equilibrium point. Note that such convergent behaviour occurs fast for case 1, and slower for case 2 and case 3 which have moderate and high degree of variability in the simulation output.

An alternative method for visualising $\mathbf{x}^{max,i}$ is shown in Figure 5.2 where the frequencies of $\mathbf{x}^{max,i}$ are shown as three dimensional histograms. The histograms illustrates the pairwise distribution of $\mathbf{x}^{max,i}$, with respect to the components $x_{CTV}, x_{SES}, x_{PER}$, for the three cases. From the figures, it is clear that $x_{CTV} = 1$ and $x_{SES} = 1$ is the most common combination of vessels for all cases. Moreover, the most common number of personnel is within the interval $[20, 30]$.

Another quantity that can be of interest is the maximum expected improvement $u^{max,i}$. This quantity is implicitly used to select the S maximisers, see (3.43) and recall that the constant C^{spread} was defined as $2u^{max,i}$. The sample paths of $u^{max,i}$ for the three cases are shown in Figure 5.3 as a function of i. The maximum EI decreases, which suggests that using $C^{spread} \propto u^{max,i}$ is reasonable. Low EI indicates that large improvements

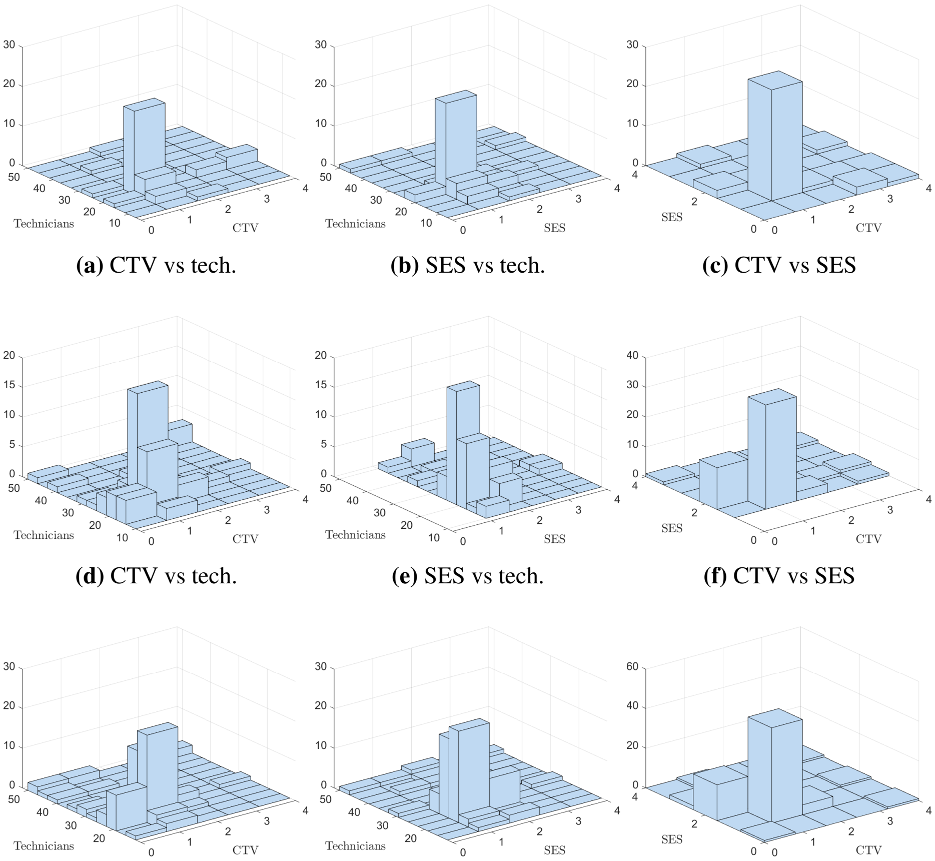


(a) $\mathbf{x}^{max,i}$ for case 1

(b) $\mathbf{x}^{max,i}$ for case 2

(c) $\mathbf{x}^{max,i}$ for case 3

Figure 5.1: The figures show the components $(x_{CTV}^{max,i}, x_{SES}^{max,i}, x_{PER}^{max,i})$ of the predicted maximisers $\mathbf{x}^{max,i}$ as a function of iteration $i = 1, \ldots, n^{iter}$ for case 1 (upper subfigure), case 2 (middle subfigure) and case 3 (bottom subfigure). Note that $\mathbf{x}^{max,i}$ stabilises for increasing i. This reflects that the surrogate model becomes more confident in which strategy $\mathbf{x}$ that maximise the simulation output. The convergence towards equilibrium point(s) occurs faster for case 1 (deterministic) and slower for the alternatives. Note that the number of iterations differs for the three cases.



(a) CTV vs tech.

(b) SES vs tech.

(c) CTV vs SES

(d) CTV vs tech.

(e) SES vs tech.

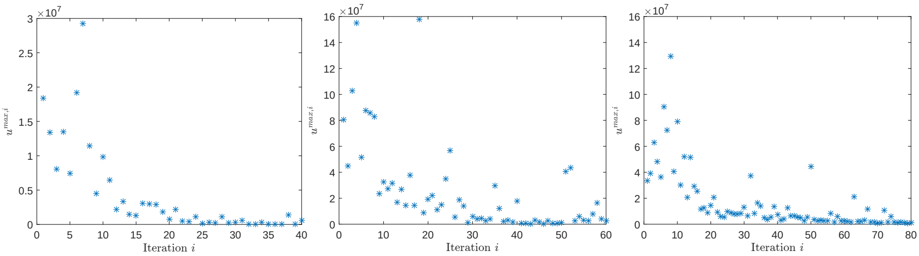
(f) CTV vs SES

(g) CTV vs tech.

(h) SES vs tech.

(i) CTV vs SES

Figure 5.2: The subfigures show histograms of the pairwise components of $\mathbf{x}^{max,i}$ for $i = 1, \ldots, n^{iter}$ for case 1(upper row), case 2 (middle row) and case 3 (bottom row). The histograms illustrates the number of times the components of $\mathbf{x}^{max,1}, \ldots, \mathbf{x}^{max,n^{iter}}$ are within certain bins. For all cases, most of the predicted maximisers have few CTVs and SESs and around 25 personnel. Note that the histograms are more centred for case 1 (upper row) than the alternative. This indicates that the predicted maximiser are similar between the iterations $i, \ldots, n^{iter}$. Note that the frequencies (z-axis) are not directly comparable due to different number of iterations for the different cases.



(a) Case 1.

(b) Case 2.

(c) Case 3.

Figure 5.3: The figures show the predicted maximum EI $u^{max,i}$ as a function of iteration $i = 1, \ldots, n^{opt}$ for case 1 (left), case 2 (middle) and case 3(right). $u^{max,i}$ reflects the expected improvement of performing a simulation for strategy $\mathbf{x}$. Note that $u^{max,i}$ generally decreases for increasing iterations. This indicate that the surrogate model becomes certain that only very small improvements over the current best predicted strategy $\mathbf{x}^{max,i}$ is likely. Note the difference in y-axis due to difference in simulation horizon between case 1 and the alternatives.

over the predicted best strategy is unlikely. Thus, decreasing $u^{max,i}$ may indicate that the sequential method predicted optimiser converges to the global optimiser as the number of iterations increases. Recall that the $S = 4$ points selected for simulations are selected using a criterion that encourages exploration. This strengthen the belief that the predicted optimiser indeed converges to the (approximate) global optimiser.

5.2 Optimal strategies

After the optimisation procedure has terminated, it is of interest to gain knowledge of which strategies $\mathbf{x}$ that are favourable. A more accurate surrogate model $f^n(\mathbf{x})$, using 20 instead of 10 ANNs, are fitted to the data samples $\mathbf{D}^n$. The strategy that maximises the surrogate model can easily be computed. However, a decision maker may be interested to gain more knowledge of how the optimal strategy compares to other good strategies, the related uncertainty and other aspects. For the decision problem at hand, the following results are computed for each of the cases

- the $n^{opt} = 20$ best strategies,
- the related profit and uncertainty

The n^{opt} predicted best strategies are defined as

$$\mathbf{x}^{opt,j} = \begin{cases} \operatorname{argmax}_{\mathbf{x}\in\Omega_{\mathbf{x}}} f\ (\mathbf{x})\,, & \text{if } j = 1 \\ \operatorname{argmax}_{\mathbf{x}\in\Omega_{\mathbf{x}}} f\left(\mathbf{x}; \mathbf{x} \notin \left\{\mathbf{x}^{opt,1}, \ldots, \mathbf{x}^{opt,j-1}\right\}\right), & \text{if } j = 2, 3, \ldots, n^{opt} \end{cases} \tag{5.1}$$

i.e. as the n^{opt} unique maximiser of the surrogate model $f(\mathbf{x})$. A search heuristic is used to identify the 20 predicted best strategies defined in Equation 5.1, hence the prediction $f\left(\mathbf{x}^{opt,j}\right)$ may not necessarily decrease for increasing j.

Table 5.1: The table show the components $\mathbf{x}^{opt,j} = x^{opt,j}_{CTV}, x^{opt,j}_{SES}, x^{opt,j}_{PER}$, the predicted profit $f\left(\mathbf{x}^{opt,j}\right)$ and prediction uncertainty $\sigma\left(\mathbf{x}^{opt,j}\right)$ for the 20 best strategies $\mathbf{x}^{opt,1}, \ldots, \mathbf{x}^{opt,20}$ for case 1. The 90% confidence interval $[\mathtt{CI}^j_{0.05}, \mathtt{CI}^j_{0.95}]$ is estimated by $f\left(\mathbf{x}^{opt,j}\right) \pm 1.645\sigma\left(\mathbf{x}^{opt,j}\right)$ for $j = 1, \ldots, 20$. A visual representation of $\mathbf{x}^{opt,j}$ and $f\left(\mathbf{x}^{opt,j}\right)$ with confidence interval $[\mathtt{CI}^j_{0.05}, \mathtt{CI}^j_{0.95}]$ is shown in figures 5.4a and 5.4b respectively.

j	$x^{opt,j}_{CTV}$	$x^{opt,j}_{SES}$	$x^{opt,j}_{PER}$	$10^{-6}f\left(\mathbf{x}^{opt,j}\right)$	$10^{-6}\sigma\left(\mathbf{x}^{opt,j}\right)$	$10^{-6}\mathtt{CI}_{0.05}$	$10^{-6}\mathtt{CI}_{0.95}$
1	1	1	21	84.86	0.08	84.72	84.99
2	1	1	20	84.84	0.08	84.72	84.97
3	1	1	22	84.84	0.11	84.66	85.02
4	1	1	23	84.79	0.13	84.56	85.01
5	1	1	19	84.77	0.11	84.59	84.96
6	1	1	24	84.72	0.15	84.46	84.97
7	1	1	18	84.64	0.18	84.35	84.93
8	1	1	25	84.63	0.17	84.36	84.91
9	1	1	26	84.54	0.18	84.24	84.84
10	2	1	21	84.49	0.18	84.18	84.79
11	2	1	20	84.47	0.19	84.16	84.78
12	2	1	22	84.46	0.19	84.15	84.77
13	1	1	27	84.45	0.20	84.12	84.77
14	1	1	17	84.43	0.25	84.02	84.84
15	2	1	23	84.40	0.20	84.08	84.72
16	1	1	28	84.35	0.22	83.99	84.71
17	2	1	19	84.40	0.21	84.05	84.75
18	1	1	29	84.26	0.25	83.85	84.66
19	2	1	24	84.32	0.20	83.99	84.66
20	2	1	18	84.24	0.26	83.82	84.66

Table 5.1 shows the 20 predicted best strategies $\mathbf{x}^{opt,1}, \ldots, \mathbf{x}^{opt,20}$, the predicted profit $f(\mathbf{x}^{opt,j})$, the prediction uncertainty $\sigma(\mathbf{x}^{opt,j})$ and the estimated 90% confidence intervals for case 1. The strategies are very similar, most of them differ only by small differences in the number of technicians. Note that there are only two vessel combinations among the 20 best predicted strategies, namely

- 1 CTV, 1 SES,
- 2 CTV, 1 SES.

Strategies based on the above vessel combinations has relatively low O&M cost and are able to complete corrective maintenance task effectively when the number of technicians are larger than approximately 18. Moreover, such strategies most often complete all annual service. Note that the predicted profit for all of the 20 strategies are high and have only small differences.

The confidence intervals may be used to assess whether a strategy is significantly better than an alternative. For example, the 90% confidence interval for the best predicted strategy has lower bound $10^{-6}84.72$ which is higher than the predicted profit for all strategies $\mathbf{x}^{opt,j}$ for $j > 6$. This indicates that the best predicted strategy is indeed better than all strategies that is not among the six predicted best.

The reader should be aware that some of the predicted best strategies are redundant. For example, strategies with a higher number of technicians than the total capacity of the available vessels could be removed since these extra technicians contributes to the O&M cost without affecting the energy production. Each of the vessel types have capacity of twelve technicians. However, in order to emphasise that the sequential optimisation method works without using prior knowledge of technicians, vessels and other aspects of a wind farm, such strategies are not removed.

The same process of fitting a high fidelity surrogate model to the available data samples $\mathbf{D}^n$, identifying the 20 best strategies with corresponding prediction, uncertainty estimate and confidence intervals are repeated for case 2 and case 3. The results are shown in Table 5.2 and Table 5.3 respectively. In order to ease the comparison across cases, the numerical results from the tables are visualised in Figure 5.4 . Comparing the 20 predicted best strategies for case 1 and case 2, either using the tables 5.1 and 5.2 or the figures 5.4a and 5.4c, it is clear that the same favourable range of technicians and vessel combinations are identified. Note also that the vessel combination

- 0 CTV, 2 SES

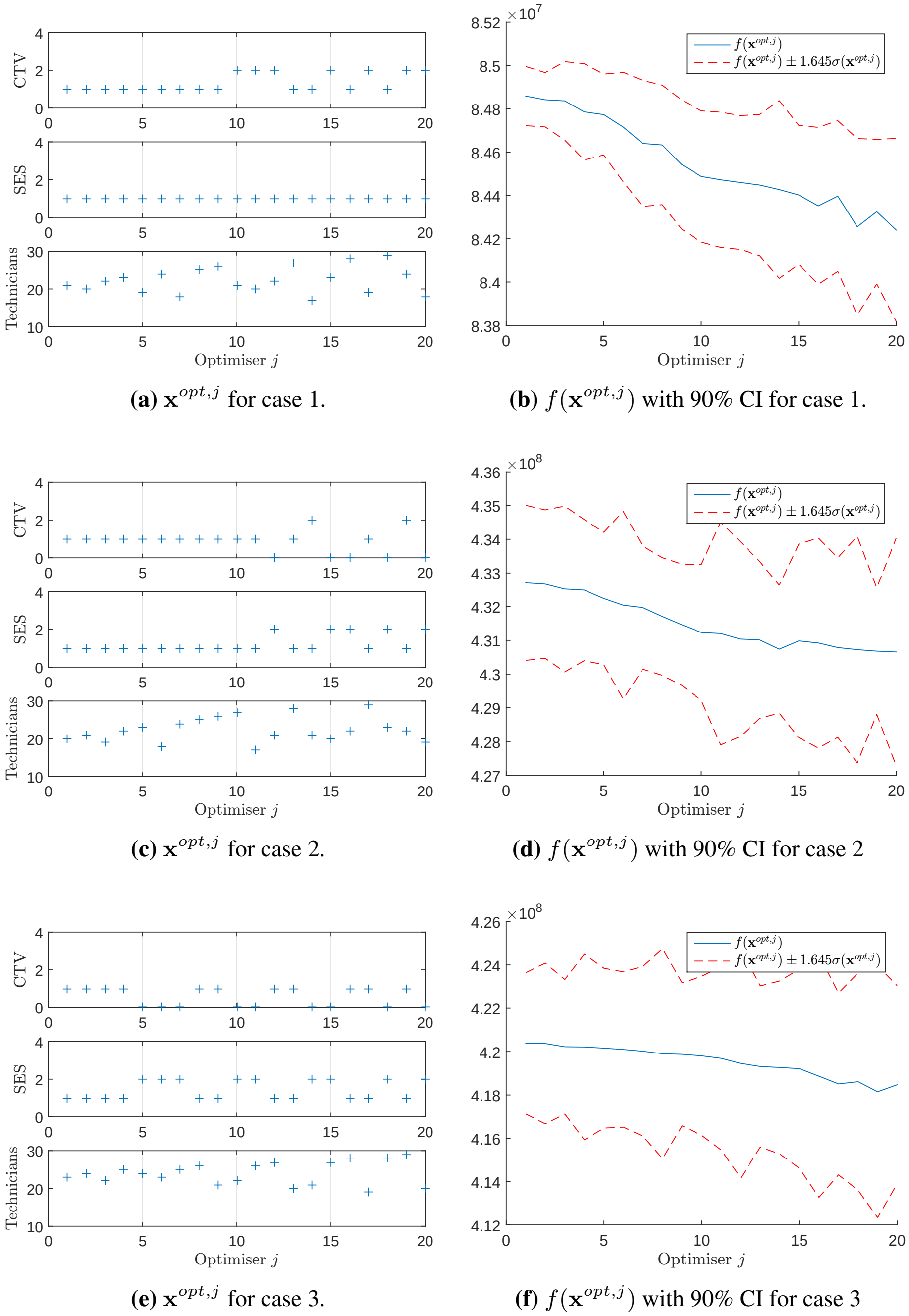
is among the 20 best predicted strategies for case 2 and case 3, but not for case 1. The confidence intervals for case 2 and case 3 are wider than for case 1 since stochastic simulations increases the prediction uncertainty $\sigma(\mathbf{x})$. The scatter plots in Figure 5.5 illustrates the prediction accuracies of the surrogate model for the three cases. The prediction and the observations are overall very similar. The accuracy is lower for the two case 1 and case 2, due to the stochastic output. Note that, for all cases, the majority of observations and predictions are much lower than the the predicted profit for the 20 optimal candidates. This indicates that the sets of predicted bests strategies are indeed among the best feasible strategies.

Table 5.2: The table show the components $\mathbf{x}^{opt,j} = x_{CTV}^{opt,j}, x_{SES}^{opt,j}, x_{PER}^{opt,j}$, the predicted profit $f\left(\mathbf{x}^{opt,j}\right)$ and prediction uncertainty $\sigma\left(\mathbf{x}^{opt,j}\right)$ for the 20 best strategies $\mathbf{x}^{opt,1}, \ldots, \mathbf{x}^{opt,20}$ for case 2. The 90% confidence interval $[\mathtt{CI}_{0.05}^{j}, \mathtt{CI}_{0.95}^{j}]$ is estimated by $f\left(\mathbf{x}^{opt,j}\right) \pm 1.645\sigma\left(\mathbf{x}^{opt,j}\right)$ for $j = 1, \ldots, 20$. A visual representation of $\mathbf{x}^{opt,j}$ and $f\left(\mathbf{x}^{opt,j}\right)$ with confidence interval $[\mathtt{CI}_{0.05}^{j}, \mathtt{CI}_{0.95}^{j}]$ is shown in figures 5.4c and 5.4d respectively.

j	$x_{CTV}^{opt,j}$	$x_{SES}^{opt,j}$	$x_{PER}^{opt,j}$	$10^{-7}f\left(\mathbf{x}^{opt,j}\right)$	$10^{-7}\sigma\left(\mathbf{x}^{opt,j}\right)$	$10^{-7}\mathtt{CI}_{0.05}$	$10^{-7}\mathtt{CI}_{0.95}$
1	1	1	20	43.27	0.14	43.04	43.50
2	1	1	21	43.27	0.13	43.05	43.49
3	1	1	19	43.25	0.15	43.01	43.50
4	1	1	22	43.25	0.13	43.04	43.46
5	1	1	23	43.22	0.12	43.03	43.42
6	1	1	18	43.20	0.17	42.93	43.48
7	1	1	24	43.20	0.11	43.01	43.38
8	1	1	25	43.17	0.11	43.00	43.35
9	1	1	26	43.15	0.11	42.97	43.33
10	1	1	27	43.12	0.12	42.92	43.32
11	1	1	17	43.12	0.20	42.79	43.45
12	0	2	21	43.10	0.18	42.81	43.39
13	1	1	28	43.10	0.14	42.87	43.33
14	2	1	21	43.07	0.12	42.88	43.26
15	0	2	20	43.10	0.17	42.81	43.39
16	0	2	22	43.09	0.19	42.78	43.40
17	1	1	29	43.08	0.16	42.81	43.35
18	0	2	23	43.07	0.20	42.74	43.41
19	2	1	22	43.07	0.11	42.88	43.26
20	0	2	19	43.07	0.21	42.73	43.40

Table 5.3: The table show the components $\mathbf{x}^{opt,j} = x_{CTV}^{opt,j}, x_{SES}^{opt,j}, x_{PER}^{opt,j}$, the predicted profit $f\left(\mathbf{x}^{opt,j}\right)$ and prediction uncertainty $\sigma\left(\mathbf{x}^{opt,j}\right)$ for the 20 best strategies $\mathbf{x}^{opt,1}, \dots, \mathbf{x}^{opt,20}$ for case 3. The 90% confidence interval $[\mathtt{CI}_{0.05}^{j}, \mathtt{CI}_{0.95}^{j}]$ is estimated by $f\left(\mathbf{x}^{opt,j}\right) \pm 1.645\sigma\left(\mathbf{x}^{opt,j}\right)$ for $j = 1, \dots, 20$. A visual representation of $\mathbf{x}^{opt,j}$ and $f\left(\mathbf{x}^{opt,j}\right)$ with confidence interval $[\mathtt{CI}_{0.05}^{j}, \mathtt{CI}_{0.95}^{j}]$ is shown in figures 5.4e and 5.4f respectively.

j	$x_{CTV}^{opt,j}$	$x_{SES}^{opt,j}$	$x_{PER}^{opt,j}$	$10^{-7} f\left(\mathbf{x}^{opt,j}\right)$	$10^{-7} \sigma\left(\mathbf{x}^{opt,j}\right)$	$10^{-7}\mathtt{CI}_{0.05}$	$10^{-7}\mathtt{CI}_{0.95}$
1	1	1	23	42.04	0.20	41.71	42.36
2	1	1	24	42.04	0.23	41.67	42.41
3	1	1	22	42.02	0.19	41.71	42.33
4	1	1	25	42.02	0.26	41.59	42.45
5	0	2	24	42.02	0.22	41.65	42.38
6	0	2	23	42.01	0.22	41.65	42.37
7	0	2	25	42.00	0.24	41.61	42.39
8	1	1	26	41.99	0.29	41.51	42.47
9	1	1	21	41.99	0.20	41.66	42.32
10	0	2	22	41.98	0.22	41.61	42.35
11	0	2	26	41.97	0.26	41.55	42.39
12	1	1	27	41.95	0.32	41.42	42.47
13	1	1	20	41.93	0.23	41.56	42.30
14	0	2	21	41.93	0.24	41.53	42.33
15	0	2	27	41.92	0.28	41.46	42.38
16	1	1	28	41.89	0.34	41.33	42.45
17	1	1	19	41.85	0.26	41.43	42.27
18	0	2	28	41.86	0.30	41.36	42.36
19	1	1	29	41.81	0.35	41.23	42.40
20	0	2	20	41.85	0.28	41.39	42.30



(a) $\mathbf{x}^{opt,j}$ for case 1.

(b) $f(\mathbf{x}^{opt,j})$ with 90% CI for case 1.

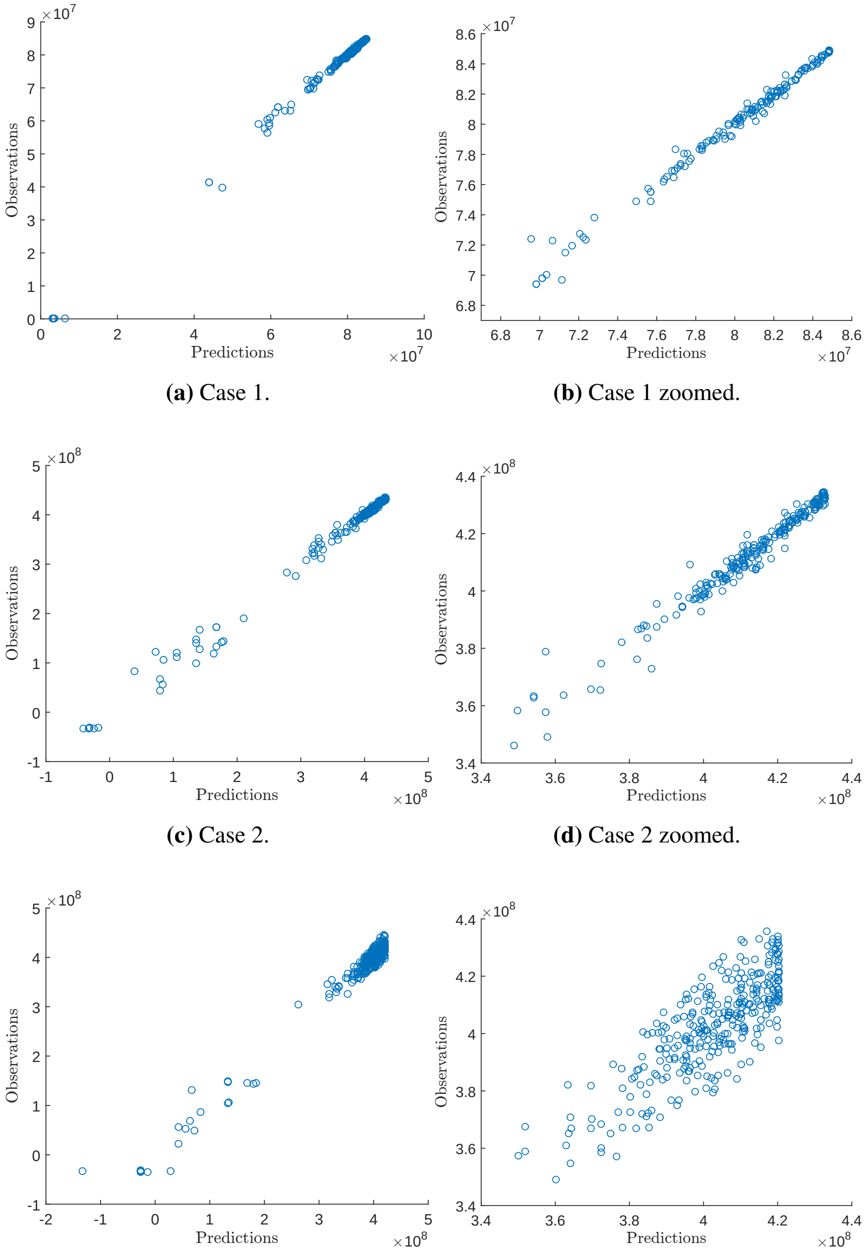
(c) $\mathbf{x}^{opt,j}$ for case 2.

(d) $f(\mathbf{x}^{opt,j})$ with 90% CI for case 2

(e) $\mathbf{x}^{opt,j}$ for case 3.

(f) $f(\mathbf{x}^{opt,j})$ with 90% CI for case 3

Figure 5.4: The left subfigures shows the 20 predicted optimal strategies $\mathbf{x}^{opt,j}$ for $j = 1, \ldots, 20$, and the right subfigures shows the predictions $f(\mathbf{x}^{opt,j})$ with corresponding 90% CI $f(\mathbf{x}^{opt,j}) \pm 1.645\sigma f(\mathbf{x}^{opt,j})$ for case 1 (upper row), case 2 (middle row) and case 3 (bottom row). For all cases, the predicted profits $f(\mathbf{x}^{opt,1}), \ldots, f(\mathbf{x}^{opt,20})$ are relatively similar. Note the difference in the y axis for the right-hand subfigures.



(a) Case 1.

(b) Case 1 zoomed.

(c) Case 2.

(d) Case 2 zoomed.

(e) Case 3.

(f) Case 3 zoomed.

Figure 5.5: Scatter plot of observations $\mathbf{D}$ and corresponding predictions $f(\mathbf{x}; \mathbf{x} \in \mathbf{D})$ for case 1 (upper subfigures), case 2 (middle subfigures) and case 3 (bottom subfigures). Note the difference in y-axis for the different cases. The right-hand figures shows the whole range of observations and predictions, whereas the left-hand figures are zoomed to the most profitable ranges.

5.2.1 Comparison of fleet mix

In the proceeding section, the surrogate model was used to identify the 20 most profitable strategies. As were discussed, the 20 most profitable strategies for all three scenarios had one of the following vessel combinations

- one CTV, one SES,
- two CTVs, one SES,
- two SES.

A decision maker may be interested in studying these vessel combinations more in detail. Figure 5.6 shows the predicted profit $f(\mathbf{x})$ and the band $f(\mathbf{x}) \pm \sigma(\mathbf{x})$ as a function of the number of technicians for the tree vessel combinations for all cases. The figures illustrates the effect of changes in the number of technicians. For case 1, the vessel combination $(x_{CTV}, x_{SES}) = (1, 1)$ seem to be better than the alternatives. The same results holds for case 2, although the differences are smaller. For case 3, both combinations $(x_{CTV}, x_{SES}) = (1, 1)$ and $(x_{CTV}, x_{SES}) = (0, 2)$ are marginally better than the alternative.

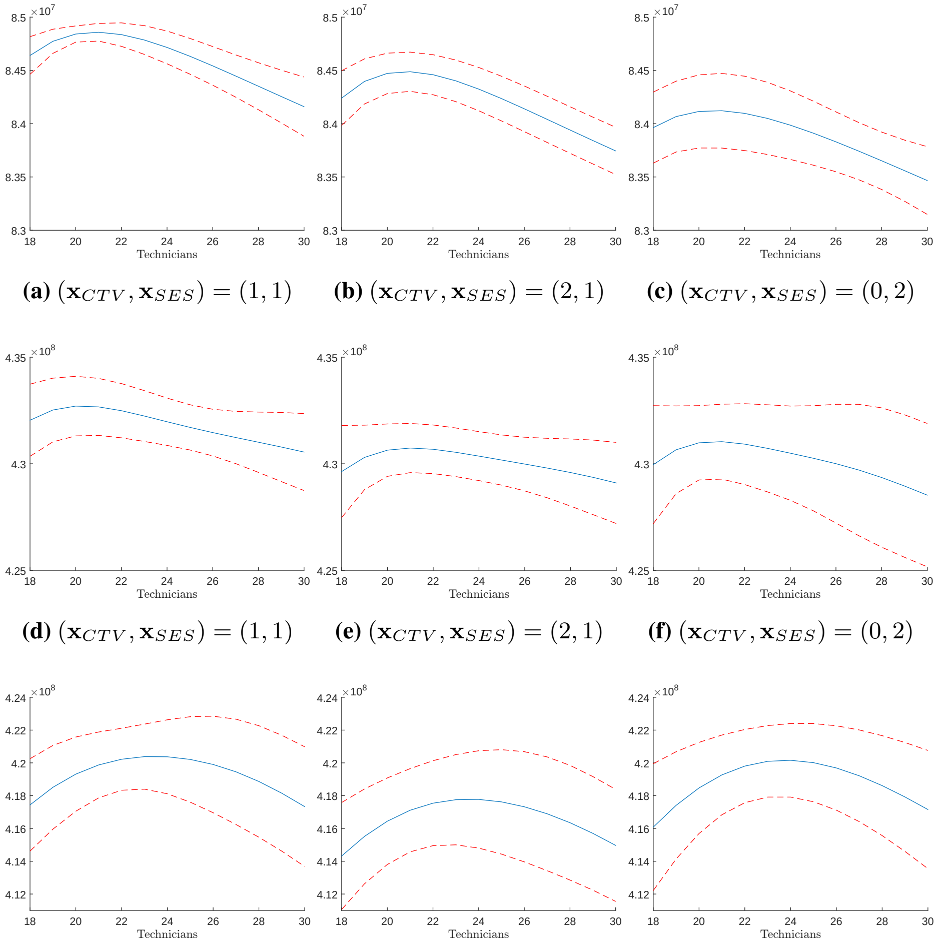
Note that for the combinations $(x_{CTV}, x_{SES}) = (1, 1)$ and $(x_{CTV}, x_{SES}) = (0, 2)$, $f(\mathbf{x})$ decreases linearly for $x_{PER} \geq 24$ for all cases. This is reasonable, since each vessel has capacity of transporting 12 technicians, hence adding more technicians corresponds to adding the fixed costs of technicians without increasing the energy production. The surrogate model captures this behaviour without any prior knowledge of the problem at hand. This indicates that the surrogate model is able to learn the relation $f^{true}(\mathbf{x})$ adequately using only the data samples $\mathbf{D}^n$.

From Figure 5.6, it is clear that the predicted profit has a non-linear relation with respect to the decision variable x_{PER}. In other words, the sensitivity of increasing or decreasing x_{PER} depends on the value of x_{PER}.

As has been demonstrated in this section, the proposed sequential optimisation method is able to identify favourable strategies for all cases. The surrogate model can be used to predict the profit and related uncertainty for different strategies. This can be used to compare different strategies and to gain knowledge of how changes in one or several input variables affect the output. The optimisation method used only information of the range of each of the decision variables $\mathbf{x}$ and the input-output $\mathbf{D}$. Thus, the same method may be used for other decision problems by changing only the range of the related decision variables.

5.3 Limitations

Recall that the vessel types CTV and SES were defined as two vessel types with difference in cost, speed and weather limit when accessing turbines, see Table 4.3. However, the vessels types used in the three cases had difference in two other parameters as well. These parameters are the weather limits for transportation and the time used for transporting technicians from the vessel to the wind turbine. The effect of the difference in the latter parameter has a negligible effect, while the former may have greater effect. The CTV



(a) $(\mathbf{x}_{CTV}, \mathbf{x}_{SES}) = (1, 1)$ **(b)** $(\mathbf{x}_{CTV}, \mathbf{x}_{SES}) = (2, 1)$ **(c)** $(\mathbf{x}_{CTV}, \mathbf{x}_{SES}) = (0, 2)$

(d) $(\mathbf{x}_{CTV}, \mathbf{x}_{SES}) = (1, 1)$ **(e)** $(\mathbf{x}_{CTV}, \mathbf{x}_{SES}) = (2, 1)$ **(f)** $(\mathbf{x}_{CTV}, \mathbf{x}_{SES}) = (0, 2)$

(g) $(\mathbf{x}_{CTV}, \mathbf{x}_{SES}) = (1, 1)$ **(h)** $(\mathbf{x}_{CTV}, \mathbf{x}_{SES}) = (2, 1)$ **(i)** $(x_{CTV}, x_{SES}) = (0, 2)$

Figure 5.6: The figures shows the surrogate prediction $f(\mathbf{x})$ (solid line) and $f(\mathbf{x}) \pm \sigma(\mathbf{x})$ (dotted line) for the three best vessel combinations as a function of the number of personnel $\mathbf{x}_{PER}$. The vessel combination (1,1), (2,1) and (0,2) are shown in the left, middle and right column respectively, for case 1 (upper row), case 2 (middle row) and case 3 (bottom row).

was specified without weather limits for travelling, whereas the SES was specified to only operate in weather with wave height less than 4 meter and wind speeds less than 30 meter per second.

The effect of this difference in weather limits are most likely small since the weather limits for the SES are large, and both vessels are restricted to the more conservative weather limits for accessing the turbines. This parameter should have been specified more consistently in order to ease comparison of the two vessel types.

Chapter 6

Conclusion

In this thesis, a surrogate-based optimisation method based on ANNs was proposed. The surrogate model used the previous simulation results and a multipoint criteria to select a set of input points for the next simulations. The method was applied for a decision problem of selecting vessel fleet and personnel for a wind farm. The results indicated that the optimisation method was able to identify favourable strategies.

The method used only information of the feasible range of the decision parameters and the input-output relations. This suggests that a sequential method based on ANNs may be used to solve other decision problems without high degree of tailoring for each problem.

As was demonstrated, a surrogate model fitted to all available data samples, can be used to identify the best strategies or predict the output and related uncertainty for arbitrary strategies. For a decision-maker, the former may be particular useful for identifying favourable regions of the input space, whereas the latter may be utilised to compare different strategies and assess the sensitivity of the output with respect to input variables.

The proposed multipoint criterion enabled better utilisation of parallel processing due to independence between simulations. The criterion used a relatively simple penalty function in order to achieve a desirable amount of spread between the selected points.

6.1 Possible improvements

It may be of interest to study the performance of the proposed sequential optimisation method on benchmark cases. Since the method was tested on a real-world problem without a known optimal solution, it may be difficult to compare the performance to alternative methods.

The points selected by the multipoint criterion was not studied in detail for the vessel fleet problem. A more thorough study of the proposed multipoint criterion should be carried out in order to assess its usability for surrogate based optimisation.

Bibliography

Álvarez, M. A., Lawrence, N. D., Jul. 2011. Computationally efficient convolved multiple output gaussian processes. J. Mach. Learn. Res. 12, 1459–1500.
URL `http://dl.acm.org/citation.cfm?id=1953048.2021048`

Banks, J., 1998. Handbook of Simulation : Principles, Methodology, Advances, Applications, and Practice. John Wiley and Sons, Inc.

Beale, M. H., Hagan, M. T., Demuth, H. B., 2016. Neural network toolbox $^{\text{TM}}$.
URL `http://se.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf`

Bekirev, A. S., Klimov, V. V., Kuzin, M. V., Shchukin, B. A., Jul. 2015. Payment card fraud detection using neural network committee and clustering. Opt. Mem. Neural Netw. 24 (3), 193–200.
URL `http://dx.doi.org/10.3103/S1060992X15030030`

Breiman, L., 1996. Bagging Predictors. Machine Learning 24 (2), 123–140.

Carney, J., Cunningham, P., Bhagwan, U., 1999. Confidence and prediction intervals for neural network ensembles. Proceedings of the International Joint Conference on Neural Neworks 2, 1215–1218.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems 2 (4), 303–314.

Dalgic, Y., Dinwoodie, I. A., Lazakis, I., McMillan, D., Revie, M., September 2014. Optimum ctv fleet selection for offshore wind farm o&m activities. In: Safety and Reliability: Methodology and Applications. pp. 1177–1185.

Dinwoodie, I., Endrerud, O., Hofmann, M., Martin, R., Sperstad, I., 2015. Reference Cases for Verification of Operation and Maintenance Simulation Models for Offshore Wind Farms. Wind Engineering 39 (1), 1–14.

Fine, T., Lauritzen, S. L., Lawless, J., 1999. Feedforward Neural Network Methodology. Springer.

Forrester, A. I. J., Sbester, A., Keane, A. J., 2008. Engineering Design via Surrogate Modelling: A Practical Guide. John Wiley & Sons, Ltd, Chichester.

Gelbart, M. A., Snoek, J., Adams, R. P., 2014. Bayesian Optimization with Unknown Constraints. In: Proceedings of the Thirtieth Annual Conference on Uncertainty in Artificial Intelligence (UAI-14). AUAI Press, Corvallis, Oregon, pp. 250–259.

Ginsbourger, D., Riche, R. L., Carraro, L., 2007. A multi-points criterion for deterministic parallel global optimization based on gaussian processes.
URL `https://hal.archives-ouvertes.fr/hal-00260579`

Gramacy, R., Lee, H., 2009. Adaptive Design and Analysis of Supercomputer Experiments. Technometrics 51, 130–145.

Gramacy, R. B., Gray, G. A., Digabel, S. L., Lee, H. K. H., Ranjan, P., Wells, G., Wild, S. M., 2016. Modeling an augmented lagrangian for blackbox constrained optimization. Technometrics 58 (1), 1–11.
URL `http://dx.doi.org/10.1080/00401706.2015.1014065`

Gundegjerde, C., Halvorsen, I. B., Halvorsen-Weare, E. E., Hvattum, L. M., Nons, L. M., 2015. A stochastic fleet size and mix model for maintenance operations at offshore wind farms. Transportation Research Part C: Emerging Technologies 52, 74 – 92.
URL `http://www.sciencedirect.com/science/article/pii/S0968090X15000078`

Hadsell, R., Sermanet, P., Ben, J., Erkan, A., Scoffier, M., Kavukcuoglu, K., Muller, U., LeCun, Y., 2009. Learning long-range vision for autonomous off-road driving. Journal of Field Robotics 26 (2), pp. 120–144.
URL `http://www.scopus.com/inward/record.url?eid=2-s2.0-67649219352&partnerID=40&md5=9f2d6ea0c1e614a40b924dce724e8297`

Hagan, M., Demuth, H., Mark, H., De Jess, O., 2003. Neural Network Design, 2nd Edition. eBook.
URL `hagan.okstate.edu/nnd.html`

Hagen, B. A. L., 2013. Sensitivity Analysis of OM Costs for Offshore Wind Farms. Master's thesis, NTNU.

Han, J., Moraga, C., 1995. From Natural to Artificial Neural Computation: International Workshop on Artificial Neural Networks Malaga-Torremolinos, Spain, June 7–9, 1995 Proceedings. Springer Berlin Heidelberg, Ch. The influence of the sigmoid function parameters on the speed of backpropagation learning, pp. 195–201.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning : Data Mining, Inference, and Prediction. Springer.

Higdon, D., 2002. Quantitative methods for current environmental issues. In: Anderson, C., Barnett, V., Chatwin, P., El-Shaarawi, A. (Eds.), Quantitative Methods for Current Environmental Issues. Springer London, pp. 37–56.
URL `http://dx.doi.org/10.1007/978-1-4471-0657-9_2`

Hoffman, M. W., Brochu, E., de Freitas, N., 2011. Portfolio allocation for Bayesian optimization. In: Uncertainty in Artificial Intelligence. pp. 327–336.

Hofmann, M., 2011. A Review of Decision Support Models for Offshore Wind Farms with an Emphasis on Operation and Maintenance Strategies. Wind Engineering 35 (1), 1–16.

Hofmann, M., Sperstad, I., Kolstad, M., 2014. Technical documentation of the NOWIcob tool (D5.1-66). SINTEF Energy Research.

Jones, D. R., 2001. A taxonomy of global optimization methods based on response surfaces. Journal of Global Optimization 21 (4), 345–383.
URL http://dx.doi.org/10.1023/A:1012771025575

Jones, D. R., Schonlau, M., Welch, W. J., 1998. Efficient global optimization of expensive black-box functions. Journal of Global Optimization 13 (4), 455–492.
URL http://dx.doi.org/10.1023/A:1008306431147

Knerr, S., Personnaz, L., Dreyfus, G., 1992. Handwritten digit recognition by neural networks with single-layer training. IEEE Transactions on Neural Networks 3 (6), 962–968.
URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=165597

Kushner, H., 1964. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. Journal of Basic Engineering 86, 97–106.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

LeCun, Y., Bottou, L., Orr, G., Mller, K.-R., 2012. Efficient backprop. Lecture Notes in Computer Science 7700, 9–48.
URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84872543023&partnerID=40&md5=93e968254b07b58d44cd66db89e0a79c

Lindberg, D. V., Lee, H. K., 2015. Optimization under constraints by applying an asymmetric entropy measure. Journal of Computational and Graphical Statistics 24 (2), 379–393.
URL http://dx.doi.org/10.1080/10618600.2014.901225

MacKay, D. J. C., May 1992. A practical bayesian framework for backpropagation networks. Neural Computation 4 (3), 448–472.

Maier, R., Dandy, G., 2000. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environmental Modelling & Software 15 (1), 101 – 124.
URL http://www.sciencedirect.com/science/article/pii/S1364815299000079

Menon, A., Mehrotra, K., Mohan, C., Ranka, S., 1996. Characterization of a class of sigmoid functions with applications to neural networks. Neural Networks 9 (5), 819 – 835.

Mirowski, P., Madhavan, D., LeCun, Y., Kuzniecky, R., November 2009. Classification of patterns of eeg synchronization for seizure prediction. Clinical Neurophysiology 120 (11), 1927–1940.
URL http://www.sciencedirect.com/science/article/pii/S1388245709005264

Osborne, M. A., Roberts, S. J., Rogers, A., Ramchurn, S. D., Jennings, N. R., April 2008. Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes. In: Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on. pp. 109–120.

Paass, G., 1993. Assessing and improving neural network predictions by the bootstrap algorithm. In: Hanson, S. J., Cowan, J. D., Giles, C. L. (Eds.), Advances in Neural Information Processing Systems 5. Morgan-Kaufmann, pp. 196–203.
URL http://papers.nips.cc/paper/659-assessing-and-improving-neural-net pdf

Pomerleau, D. A., 1989. Alvinn: An autonomous land vehicle in a neural network. In: Advances in Neural Information Processing Systems 1. Morgan-Kaufmann, pp. 305–313.

Ponweiser, W., Wagner, T., Vincze, M., June 2008. Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). pp. 3515–3522.

Santner, T., Williams, B., Nots, W., 2003. The design and analysis of computer experiments. Springer series in statistics. Springer.

Schonlau, M., Welch, W., Jones, D., 1998. Global versus local search in constrained optimization of computer models. Vol. 34 of Lecture Notes–Monograph Series. Institute of Mathematical Statistics, Hayward, CA, pp. 11–25.
URL http://dx.doi.org/10.1214/lnms/1215456182

Shafiee, M., 2015. Maintenance logistics organization for offshore wind energy: Current progress and future perspectives. Renewable Energy 77, 182193.

Shan, S., Wang, G., 2010. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Structural and Multidisciplinary Optimization 41 (2), 219–241.

Staffel, I., 2015. Wind turbine power curves.
URL http://www.academia.edu/1489838/Wind_Turbine_Power_Curves

Storlie, B., Reich, B., Helton, J., Swiler, L., Sallaberry, C., 2013. Analysis of computationally demanding models with continuous and categorical inputs. Reliability Engineering and System Safety 113 (1), 30–41.

Tande, J. O. G., Kvamsdal, T., Muskulus, M., Halvorsen-Weare, E. E., Gundegjerde, C., Halvorsen, I. B., Hvattum, L. M., Nons, L. M., 2013. Vessel fleet analysis for maintenance operations at offshore wind farms. Energy Procedia 35, 167 – 176.
URL http://www.sciencedirect.com/science/article/pii/S1876610213012563

Williams, B. J., Santner, T. J., Notz, W. I., Lehman, J. S., 2010. Statistical Modelling and Regression Structures: Festschrift in Honour of Ludwig Fahrmeir. Physica-Verlag HD, Heidelberg, Ch. Sequential Design of Computer Experiments for Constrained Optimization, pp. 449–472.
URL http://dx.doi.org/10.1007/978-3-7908-2413-1_24

Zissis, D., Xidias, E., Lekkas, D., 2015. A cloud based architecture capable of perceiving and predicting multiple vessel behaviour. Applied Soft Computing 35, 652 – 661.
URL http://www.sciencedirect.com/science/article/pii/S1568494615004329

Appendix

A Input data and model assumptions

In all case studies considered in this project, it have been assumed that all turbines is working when the simulation starts. This assumption may be too optimistic, and it has a significant effect for simulations performed over a short time horizon. For longer simulations, the effect stabilises during the first years. The investment cost is neglected, which results in high profit for most O&M strategies. However, the investment cost does not depend on the decision variables, and could be incorporated as a shift in the profit.

Explanations of the most central input parameters related to wind farm specification, workforce, turbine failures and vessels are listed in tables 6.1 , 6.2 , 6.3 and 6.4 respectively. Some the available simulation output are listed in Table 6.5 . The values of the different input variables related to wind farm and simulation setup, workforce, failures and vessels used for all simulations are listed in tables 6.6 , 6.7 , 6.8 A set of historical weather records is used for case 1, and synthetic weather generated from these records are used for case 2 and case 3.

Table 6.1: Explanation of input parameters related to the wind farm and simulation setup

Input parameter	Definition
Weather	Time series for wave height and wind speed are generated from historical weather from the wind farm location.
Turbine type	Properties as power curve, physical dimensions, cut-in and cut-off speeds differs for different types.
Turbine number	The total number of turbines at the wind farm
Distance to location	The shortest distance from the wind farm to the location(s) with personnel accommodation.
Distances for (un)planned tasks	The average distance vessels travels between turbines for planned and unplanned maintenance tasks.
Electricity price	Defines the price of electricity each month through the entire lifetime of the wind farm. It can be constant, seasonal etc.
Simulation horizon	The simulated lifetime of the wind farm
Time resolution	The time unit that defines the smallest time step
Weather resolution	The minimum difference in wind speed and wave height that are considered
Simulation runs	How many simulations that are performed for the same input case. Since the weather and failure times are stochastic, the result of several simulations differs.
Wake loss	Loss in produced energy due to wake effects
Electrical loss	Loss in produced energy due to electrical infrastructure
Downtime loss	Loss in produced energy due to downtime in electrical infrastructure
Discount rate	Used to calculate the net present value of future income and costs
Fuel price	Price for the fuel used by the vessels. For some chartered vessels the fuel cost is included in the charter rate.

Table 6.2: Explanation of input parameters related to the workforce

Input parameter	Definition
Daily shifts	The number of shifts each day.
Working hours	The number of working hours in each shift.
Shift start	The time each shift starts.
Minimum working hours	The maintenance task will be postponed if the available time is less than the minimum working hours.
Personnel available	The average number of maintenance or technician personnel available each shift. These are stationed at an onshore or offshore location.
Personnel cost	The fixed cost for each personnel.
Prioritisation	Defines which maintenance task have highest priority. For example maintenance tasks that are already started, but not finished, could have higher prioritisation than corrective tasks.

Table 6.3: Explanation of input parameters related to turbines and failures

Input parameter	Definition
Failure type	The different failure types have different consequences. They may partly or fully reduce the turbines ability to produce energy, or they can be annual services in order to prevent future failures. Maintenance, either remotely or one site, are required to restore the turbine after failure.
Rate	The different failures types are assumed to occur randomly with some intensity. Maintenance tasks that are performed regularly, as annual service, can be performed at predetermined dates.
Repair cost	Some maintenance tasks requires replacement of spare parts. These have an associated cost.
Lead time	Most spare parts are always be available, but others may have a lead time associated to the ordering process.
Work duration	The expected number of hours used to perform the maintenance tasks.
Personnel needed	Denotes the number of personnel needed on the structure to perform the maintenance tasks.
Abilities	Replacement of heavy parts can only be performed by vessels with special abilities, as for example jack up rig etc.
Access needed	Most failures requires technicians assessing the turbine, while some may be performed remotely.
Logistics time	The time needed to transfer crew or equipment before the work can start.

Table 6.4: Explanation of input parameters related to vessels.

Input parameter	Definition
Vessel type	The vessels available for charter or purchase have different properties that affect their operation and maintenance abilities. Some of the vessels are used merely to transport personnel to the offshore wind farm, while others have abilities that either are required or may be useful to perform the operational tasks efficiently.
Number	The number of available vessels of each type.
Day rate	Daily cost for chartering a specific vessel.
Mobilisation cost	Cost related to the process of preparing a vessel.
Lead time	Ships that are chartered sporadically often have an associated lead time before they are available in the wind farm.
Weather limits	The vessels are assumed to have some limits related to the significant wave height and wind speed. These limits often may be depend on whether vessels are traveling or accessing a turbine.
Speed	At which speed the vessels can travel to and within the wind farm.
Space	The vessels have an upper limit on the number of personnel that are transferred or offered accommodation.
Abilities	Some vessels have special abilities that are required to perform certain tasks, or are useful in other ways. Vessels with jack-up abilities could be used to lift heavy parts, while others have support for a helicopter.

Table 6.5: Explanation of some output parameters.

Output parameter	Definition
Total direct O&M cost	The sum of all costs related to vessels, repair, personnel and location.
Total energy production	Takes into account availability, loss and downtime.
Annual direct O&M cost	The total direct O&M cost divided by the simulation horizon.
Annual energy production	The total amount of produced energy divided by the simulation horizon.
Time-based availability	Defined here as the percentage of time the turbines are operative, including availability of electrical infrastructure.
Electricity-based availability	The actual electricity production measured relative to the theoretical production. The latter takes into account losses due to wake effects and electrical infrastructure, while the former also consider the downtime of turbines and in the electrical infrastructure.

Table 6.6: Specification of input related to the wind farm specification and simulation setup. All losses, discount rate and fuel price are neglected and therefore not listed in this table.

Input parameter	Value
Weather	FINO dataset
Turbine type	Vestas V90 3.0MW (Staffel, 2015)
Turbine number	80
Distance to location	50
Distances for (un)planned tasks	0
Simulation horizon	1 and 5 years
Time resolution	1 hour
Weather resolution	0.1 m and 0.1 m pr s
Runs	1
Electricity price	90 GBP pr MWH

Table 6.7: Specification of input related to failures. Lead time and the logistics time are neglected, and access to the wind farm are required for all failure types. These parameters are therefore not listed in the table.

Failure type	Manual reset	Minor repair	Medium repair	Annual service
Rate [pr year]	7.5	3.6	0.34	1
Repair cost [GBP]	0	1000	185000	18500
Work duration [hours]	3	7.5	22	60
Personnel needed on structure	2	2	3	3
Stop during repair	Yes	Yes	Yes	Yes
Stop at failure	Yes	Yes	Yes	No

Table 6.8: Specification of input related to the workforce.

Input parameter	Value
Daily shifts	1
Working hours	12 hours
Shift start	At 06:00
Minimum working hours	0.5 hours
Personnel available	20
Personnel cost	80.000 GBP pr year
Prioritisation	Corrective tasks