Sequential, Batch and Multi-Objective Bayesian Design of Experiments for Additive Manufacturing Processes

A thesis submitted to University of Sheffield for the degree of Doctor of Philosophy

Guy Alexander Harding

Department of Automatic Control and Systems Engineering

February 2023

Abstract

Research and development of new manufacturing processes require large investments of time, labour, and resources to develop new products. Traditionally, experimentation for manufacturing has been handled using Design of Experiments or Experimental Design (DoE) such as: OVAT, Factorial Designs, and Taguchi Orthogonal Arrays. Whilst these DoE approaches are simple to implement, they select their full experimental budget prior to experimentation which can lead to excessive costs in areas of low experimental value.

In new manufacturing industries such as Additive Manufacturing (AM) experimentation is becoming increasingly expensive, whereby traditional DoE approaches will lead to increased costs. The research challenge is to develop data-efficient DoE methods that iteratively select experiments to maximise information gained whilst minimising the number of experiments to improve the understanding of the underlying processes and/or locate the global optimum.

This thesis investigates the use of Bayesian Optimisation (BO) as a data-efficient DoE method for expensive AM DoE problems and subsequent development of two novel algorithms, Batch Bayesian Experimental Design Optimisation (BB-DoE) and Multi-Objective Batch Bayesian Experimental Design Optimisation (MOBB-DoE). They are then assessed against current state-of-the-art algorithms and/or applied onto expensive AM DoE case studies.

To assess the methodological viability of BO an exploratory investigation is presented. The investigation utilised synthetic benchmarks, theoretical property analysis, and an AM case study to demonstrate BOs capabilities. This investigation also produces the necessary information to contrast each BO cost function through analysis of their properties and performance to determine the suitable cost function to act as the foundation for further algorithmic development.

Both BB-DoE and MOBB-DoE algorithms were shown to have improved or comparable performances against current state-of-the-art algorithms on synthetic problems. BB-DoE also demonstrated favourable performance on an AM case study for a Directed Energy Deposition (DED) process seeking an optimal Dendritic Arm Spacing (DAS) property by locating the optimal DED settings.

Acknowledgements

I would like to first express my gratitude to my supervisor Professor Visakan Kadirkamanthan. In the beginning of my research he guided me on what it meant to be a researcher offering guidance and constructive criticism along my path. His countless ideas and abstract thoughts allowed me to always look at my work from a different point of view giving birth to some of the best ideas of my thesis. However, the most important quality I am thankful for is his compassion and willingness to help when i went through struggles regarding mental health during my research.

I would also like to thank my second supervisor Professor George Panoustos for his useful advice, suggestions and insight from the manufacturers perspective of my thesis.

I also want to give thanks to the many colleague's in the Rolls Royce UTC who provided me the opportunity to be apart of their researching community. To bounce ideas of one another and experience not just my own field of work but many other fields broadening my horizons but also gaining many friends in the process. In particular I would like to thank Andrew Hills for his kind words and guidance through difficult times and his fount of knowledge to seemingly always have an answer to my obscure questions.

Also my friends who always would check up on me to ensure I didn't shrink into reclusion and keep my life balanced as much as possible.

Finally, my parents Paola and Rick for their love, support and plenty of patience during all the stress, seemingly hopeless times they offered to let me bounce ideas off them even if they didn't fully understand the topic. Relentlessly pushing me to achieve this amazing feat, I couldn't ask for anything more.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

A list of the variables and notation used in this thesis is defined below. The definitions and conventions set here will be observed throughout unless otherwise stated. For a list of acronyms, please consult page xxv.

α	GA BLX-a blending constant
$\bar{y}$	Hallucinated Output
β_t	Exploration/Exploitation time-varying trade-off parameter for Optimistic acquisition functions
χ	GA Cost of an individual in the population
δ	Exploration/Exploitation time-varying trade-off parameter kernel parameter
F	Elite set update difference value
ϵ	Random error
η	Polynomial Coefficients
$\Gamma(.)$	Gamma function
γ	GA BLX-a blending operator
$\hat{y}$	Estimation of Output y
κ	Stopping criterion threshold
λ	GP surrogate model hyper-priors
$\mathcal{R}$	Design Space
$\mu_t(x)$	Predictive mean of input x at iteration time step t

Ω	Uncertainty Region
ω	Total number of Monte Carlo Estimations
$\Phi(.)$	Cumulative distribution function
$\varphi(.)$	Probability density function
ψ	Total number of sobol sequence points
Ψ_{T}	Total number of stopping criterion threshold iterations
Ψ_{t}	Stopping criterion threshold iteration counter
ρ	Stopping criterion difference value
σ_{b}	Periodic kernel uncertainty off-set hyper-parameter
σ_{f}	Signal Variance hyper-parameter
σ_{n}	Noise hyper-parameter
$\sigma_{\mathrm{t}}(\mathrm{x})$	Predictive standard deviation of input x at iteration time step t
τ	Target Improvement in Improvement function
$\mathbf{c}$	Vector set of optimisation constants
$\mathbf{o}$	Vector of Optimisation settings
θ	Hyper-parameter vector set
$\Upsilon(.)$	Average fitness function
ζ	EI trade-off constant
B	Batch set of experiments
$\mathrm{BS}_{\mathrm{size}}$	Boundary stretch coefficient
BT	Maximum experimentation budget counter
C	Individual in GA population
c	Linear Kernel off-set hyper-parameter
CI(.)	Confidence Interval function
D	Data set

d	Total number of input variables
$D_{penalty}$	GA Death Penalty
E	GA elite population
$E[.]$	Expectation Function
E_{size}	GA Elite population size
$Euc_d(.)$	Euclidean Distance function
$f(x)$	function output
$g(.)$	Inequality constraints
G	GA Generation Number
$Gen_d(.)$	Generational Distance function
$h(.)$	Equality Constraints
H	Total number of hyper-parameter sets
$H[.]$	Differential entropy of probability distribution p(.)
HV	Hyper-volume indicator
$I(.)$	Improvement function
I	Identity Matrix
IBC_{con}	Non-Greedy Batch Constraint, Intra-Batch Constraint
j	Member of a set
$k(.,.)$	Covariance function
K	Covariance Matrix
k	Number of cross-validation folds
k_{plat}	Rate constant of the plateau function
L	Total number of uncontrollable variables
L_d	Length Scale hyper-parameter
$\log(.)$	Natural Logarithm function

m(.)	Mean function
M	Total number of output variables
M_r	Mutation Rate
N	Total number of data points
N_{batch}	Total number of batches
N_{Gen}	Total number of GA generations
n_s	Total number of Latin hyper-cube samples
NF_{evals}	Number of function evaluations
O	GA Population offspring
ODC_{con}	Non-Greedy Batch Constraint, Observed Data Constraint
P(.)	Probability function
p	Periodic Kernel hyper-parameter
P_r	Pareto Rank
P_{size}	GA Population size
Q	Total number of experiments in a batch
R_t	Regret at interval t
rnd	Random number
S	GA Tournament size
$S_{penalty}$	Severe Reduction penalty
T	Experimentation budget
t	Time step or interval step
TL	Target level for the improvement function
u(.)	Acquisition function
$u(r_i)$	Total count for rank i
U	Hyper-rectangle

U	Uncertainty Hyper-rectangle
v^{max}	Maximum or Upper limit of variable
v^{min}	Minimum or Lower limit of variable
v^{norm}	Feature scaled variable
v^{score}	Standard Score
W	Total number of weight vectors
w	Weighting
w^{i}	weight vector i
x	Input variable
x^{*}	Optimum Input
x^{+}	Incumbent Input
x^{L}	Minimum or Lower limit of x
x^{M}	GA Mutated input variable
x^{U}	Maximum or Upper limit of x
y	Output variable
y^{L}	Minimum or Lower limit of y
y^{U}	Maximum or Upper limit of y
YP	Plateau function output
YP_{max}	Maximum plateau output
YP_{min}	Minimum plateau output
Z	Total number of noise variables
z	Uncontrollable variable
z^{*}	Reference point

Acronyms

AM Additive Manufacturing. i, 2–6, 11, 19, 25, 28, 30, 52, 53, 56, 67, 68, 70, 93, 101–103, 112, 113, 116–118, 131, 161, 162, 164, 170, 178, 190, 203, 212–219

ANOVA Analysis of Variance. 23, 25–27, 70

ARSM Adaptive Response Surface Methodology. 28–30, 94, 128

B-DoE Bayesian Experimental Design Optimisation. ix–xi, xvii, 3, 5, 8, 56, 59, 70, 72, 75, 83, 94–96, 98, 101–109, 111–118, 126, 128–131, 144, 149, 158, 171, 208, 214, 216, 217, 220

BB-DoE Batch Bayesian Experimental Design Optimisation. i, xii–xv, xvii, xviii, 3–7, 53–55, 118, 123, 128, 131–133, 135, 138, 141, 143, 144, 149–160, 162, 164–171, 177, 178, 192, 203, 208, 211–216, 219, 220, 222

BBD Box-Behnken Design. 26

BO Bayesian Optimisation. i, iv, 2–5, 30–32, 39–41, 45, 48–53, 67, 68, 70, 71, 93–95, 118, 120, 122, 124, 126, 128–131, 145, 147, 148, 178, 181, 184, 185, 189, 190, 213–216

CCD Central Composite Design. 26, 27

D-MOEA Decomposition Based Multi-Objective Evolutionary Algorithm. ix, 59, 64–67, 180, 187, 189–192

DACE Design and Analysis of Compute Experiments. 50, 180

DAS Dendritic Arm Spacing. i, xiv, xv, xviii, 6, 160–163, 165, 166, 168–171, 173–176, 214, 222

DAS-D Dendritic Arm Spacing Deviation. 163

DED Directed Energy Deposition. i, xv, xvii, xviii, 6, 131, 160–162, 164–167, 170, 171, 177, 214, 218, 222

DMAIC Design Measure Analyse Improve and Control. 14

DoE Design of Experiments or Experimental Design. i, iii, vii, xii, 1–11, 13–16, 19–24, 26, 28, 30, 31, 52, 53, 58, 67–71, 75, 83, 85, 93–96, 101–103, 109, 111, 113, 114, 116–118, 126–132, 136, 137, 160, 162, 163, 165–167, 178, 179, 189, 207, 212–219, 221

EA Evolutionary Algorithm. 59, 61, 62

EGO Efficient Global Optimisation. 50

EI Expected Improvement. x, xi, xvii, 43, 44, 50–52, 67, 68, 93, 95, 104–109, 115–123, 125, 129, 130, 147, 178, 180, 184, 185, 187, 190, 199, 214

EP Expectation Propagation. 46, 47, 123, 124, 130

ES Entropy Search. 46, 123

FDM Fused Deposition Modelling. 25

GA Genetic Algorithm. xii, 5, 48, 49, 52, 59, 73–75, 79–86, 91, 93–95, 131, 141, 144, 151, 163, 164, 180, 189, 193–195, 198, 214, 217, 219, 220

GD Generational Distance. xvi, 204, 205, 207, 208, 211

GP Gaussian Process. vii, viii, 32–34, 36, 39, 40, 42, 44, 49, 85, 86, 93, 95, 103, 106, 111–113, 115, 117, 131, 135, 142, 147, 148, 151, 162, 163, 165, 168–170, 180, 181, 185, 186, 188, 192, 193, 199, 203, 211, 220, 221

GP-BUCB Gaussian Process Batch Upper Confidence Bound. xii–xiv, 135, 150–160, 170, 171, 214

GP-CB Gaussian Process Confidence Bound. x, xi, 68, 95, 104–109, 115–118, 125, 126, 128, 130–134, 136, 150, 154, 178, 181, 190, 192, 214

GP-LCB Gaussian Process Lower Confidence Bound. 45, 125, 199

GP-UCB Gaussian Process Upper Confidence Bound. 45, 125, 199

GPR Gaussian Process Regression. iii, 30, 33, 69, 72, 85, 87–91, 94, 104, 135, 138

HV Hyper-Volume. 181, 182, 186, 188, 190, 191

HVI Hyper-Volume Improvement. 187, 188

IBC Intra-Batch Constraint. 140, 141, 164

KNN K Nearest Neighbour. 196, 204

LHD Latin Hypercube Design. vii, 16, 18, 218

MAE Mean Absolute Error. 72, 73

MaOP Many-Objective Optimisation Problem. 61, 63, 64, 66, 67, 191

MCS Monte Carlo Sampling. 15

MES Max Value Entropy Search. x, xi, 47, 48, 51, 95, 102, 104–109, 115–118, 123, 124, 129, 130, 183, 184, 191, 214

MESMO Max-Value Entropy Multi-Objective Search. 183

MOBB-DoE Multi-Objective Batch Bayesian Experimental Design Optimisation. i, xv, xvi, xix, 3, 4, 6, 178, 187–192, 197, 199, 200, 202–217, 219, 220, 222

MOBBO Multi-Objective Batch Bayesian Optimisation. 187, 189, 190, 195

MOBO Multi-Objective Bayesian Optimisation. 178, 180–185, 187–191, 195, 196, 198, 199

MOEA Multi-Objective Evolutionary Algorithm. 52, 59, 61–64

MOEA/D Multi-Objective Evolutionary Algorithm via Decomposition. viii, 63–65

MOGA Multi-Objective Genetic Algorithm. 195

MOO Multi-Objective Optimisation. 56–59, 112, 184, 186, 189, 191, 192, 198, 204, 217, 218

MOP Multi-Objective Problems. viii, 4, 56, 58, 59, 61–67, 112, 180–182, 184, 186–188, 190, 191, 193, 195, 211, 212, 214

MSE Mean Squared Error. 72

NGB-GP-CB Non-Greedy Batch GP-CB. xii–xiv, xvii, 133–137, 140–142, 149–160, 163, 170, 171, 178, 179, 189, 191–193, 199, 202, 210, 214

NRMSD Normalised Root Mean Square Deviation. 73, 104, 130, 167, 171

ODC Observed Data Constraint. 140, 141, 164

OVAT One-Variable-at-a-time. i, xvii, 1, 13, 15, 20, 21

P_I Performance Indicator. 62, 63

PD-MOEA Pareto-Dominance Multi-Objective Evolutionary Algorithm. 59–61, 63, 65, 67, 178, 184, 186, 189, 192, 193, 198, 199, 202, 210, 214, 217, 221

PES Predictive Entropy Search. 46, 47, 51, 123, 183, 191

PESMO Predictive Entropy Multi-Objective Search. 183

PF Pareto Front. xvi, 4, 7, 57, 59, 62, 65, 112, 180, 181, 184, 186, 189–192, 194, 198, 202, 204–212, 215–217, 221

PI Probability of Improvement. 41–43, 51, 119, 147, 182

PI-MOEA Performance Indicator Multi-Objective Evolutionary Algorithm. 59, 62, 63, 65, 67, 189–192

POS Pareto Optimal Set. 4, 7, 56–62, 64, 178, 179, 181–186, 188, 189, 191, 192, 198, 199, 202–205, 207, 210, 211, 215

PPES Parallel Predictive Entropy Search. 150, 151, 153, 155, 170, 214

PPO Process Parameter Optimisation. 25

QMCS Quasi-Monte Carlo Sampling. 15

RD Robust Design. 11, 19

RMSE Root Mean Squared Error. 72, 73

RSM Response Surface Methodology. 11, 13, 19, 26–29

SE Squared Exponential. 34, 35, 38, 87, 88

SE-ARD Squared Exponential Automatic Relevance Determination. 88

SLM Selective Laser Melting. xi, 109–112, 114–117, 128–130

SMCS Stratified Monte Carlo Sampling. 15

SOBO Single-Objective Bayesian Optimisation. 180, 182–184, 186, 191, 192

SOO Single-Objective Optimisation. 58

SOP Single-Objective Problems. viii, ix, 58, 63–67, 112, 113, 180, 181, 187, 190, 191, 207

TS Thompson Sampling. 122, 180

Chapter 1

Introduction

Within current and developing manufacturing industries the development of new manufacturing processes, products, and implementation of new machinery require an incredibly large investment. These Research and Development phases require large investments of time, labour, and resources prior to consumers ever seeing or purchasing the final products.

In order to effectively design new products/processes for use in manufacturing industries the underlying interactions between controllable manufacturing variables and the desired outcomes in product properties needs to be understood. In order to gain the understanding of the process interactions, experiments are required to access the process at varying levels of controllable variables (machine settings, feed materials, operational constraints). The techniques used to perform these experiments to understand these interactions are Design of Experiments (DoE) or Experimental Design methods which are used to analyse and inform users on which experiments to perform to gather as much information as possible.

1.1 Motivation

The DoE techniques used in current and developing industries is typically of a very simplistic nature. The techniques that are often used are simple One-Variable-at-a-time (OVAT) methods which are implemented due to their ease of use [213]. Other techniques do see application but at a more infrequent rate such as Taguchi Orthogonal Arrays, Factorial or Fractional Factorial Designs [109] which serve as an introductory approach for statistical methods of DoE. The DoE approaches introduced so far are static approaches in which every experiment that is selected must be performed prior to analysis [164].

One disadvantage of such approaches is the non-adaptability of experiments which can lead to excessive experimentation in areas of low importance. Coupled with the lack of data analysis until the completion of experimentation will inhibit these approaches use in Industry 4.0. Wherein, Industry 4.0 the increasingly complex processes accompanied alongside manufactures' lack of willingness to adopt statistical methods for selecting experiments, will lead to excessive experimentation being performed needlessly [177].

The problem plaguing the introduction and application of statistical methods for selecting experiments for use in manufacturing can be broken down into two main areas [213]:

- Theoretical ignorance of DoE for real applications.
- Absence of a clear methodology to simplify implementation.

The first key issue in manufacturing DoE is the availability of suitable techniques for the types of problems faced in manufacturing industries, whereby in industry the types of problems that manufacturers can face are varied depending on the industry to which they belong to. For example, in new up and coming industries such as Additive Manufacturing (AM): (3-D printing, Metal Powder Manufacturing, Polymers), these industries are capable of producing detailed and intricately designed products using new materials. For these developments they need to analyse the machining parameters and feedstock quality of many types of materials to optimise their processes, this leads to requiring a method with these requirements:

- Capable of handling many input variables. (High Dimensional Problems)
- Capable of optimising many output variables. (Multi-Objective)
- Capable of utilising multiple manufacturing units simultaneously. (Batch)
- Minimise experimentation time/resource use. (Expensive Materials)
- Help with post-experimentation needs. (Model Development and Data post-processing)

One method that has been gaining traction in literature that tackles some of these issues outlined is Bayesian Optimisation (BO). BO is a statistical experimental design optimisation framework that uses statistical modelling with a Bayesian approach to select new experiments in an iterative manner, such that each experiment selected provides maximal information gain to the user in order to guide towards an optimum. This methodology was originally introduced back by Mockus et al. [161] but has been gaining renewed interest in the most recent decade as an efficient iterative approach for DoE in fields such as: robotics [144], environmental monitoring [153], automatic machine learning [25] [210], sensor networks [77], and experimental design [13].

BO uses an optimisation framework to iteratively select the next best experiment to perform by building a surrogate statistical model of the process to cheaply evaluate each experiments potential value. Once an experiment is chosen and evaluated its results are then used to update the surrogate model, this stage allows for each experiment selected to improve knowledge in the region of interest. Thus, with each experiment selected the model updates to better encapsulate the information obtained so far and guide the selection of the next best experiment to perform. By using a BO approach, the surrogate model is built and updated throughout, the data post-processing is handled during operation. Also, the iterative selection allows for reduction in excessive over-experimentation from selection prior to experimentation.

1.2 Research Aims and Objectives

Whilst BO has provided a solid framework for iterative DoE for use in a variety of industries such as: hyper-parameter tuning [25], robotics [49], neuroscience [135], materials discovery [218], and Additive Manufacturing (AM) [246]. It is still lacking in some its use-cases for desirable properties that manufacturers require, as well as lacking in uptake of use by manufacturers. These properties include a generalised and simple to understand method that the consumers with less technical knowledge in the fields of statistics can apply.

Also more recently developments in the BO literature have begun the development of Batch Bayesian Experimental Design Optimisation (BB-DoE) approaches [59] [192] and Multi-Objective Batch Bayesian Experimental Design Optimisation (MOBB-DoE) approaches [51] [225] using the basis of B-DoE to tackle some of the lack of availability of these techniques. However, the approaches utilised are often designed for use by people in the same field and utilise overly complex statistical techniques preventing their adoption for use in manufacturing [213]. Therefore this work aims to produce suitable techniques which utilise more generalised approaches and easier to follow methodologies to encourage the implementation in manufacturing DoE problems as well as demonstrate its capabilities on additive manufacturing problems. The aims of this study are:

- Reproduce and benchmark existing Bayesian Optimisation (BO) algorithms for Design of Experiments or Experimental Design (DoE) problems. Assess and analyse their performances with regard to exploration and exploitation of the design spaces to assess suitability of acquisition functions for later development within the thesis.

- Investigate the theoretical properties of the available BO acquisition functions to assist in the determination of which acquisition function is most suitable to act as the foundation for further development throughout this thesis.

- Investigate whether the suggested BO algorithms can provide a suitable basis for expensive DoE on Additive Manufacturing (AM) problems to improve efficient data driven experiment selection.

- Extend and benchmark an existing BO acquisition function with suitable properties by designing a novel Non-Greedy Batch Bayesian Experimental Design Optimisation (BB-DoE) that will both generate accurate surrogate model (exploration) and locate the global optimum (exploitation) using a batch selection scheme.

- Compare the non-greedy BB-DoE algorithms performance against both types of current state of the art algorithms: a greedy BB-DoE algorithm and a non-greedy BB-DoE algorithm.

- Investigate the performance of any developed non-greedy BB-DoE algorithm on a blind AM manufacturing case to assess performance in a real-world scenario in which no experimentation has been previously applied.

- Extend and benchmark the BB-DoE algorithm for use onto Multi-Objective Problems (MOP) by designing a novel Multi-Objective Batch Bayesian Experimental Design Optimisation (MOBB-DoE) algorithm suited to producing an accurate, diverse and spread Pareto Optimal Set (POS) that represents the true Pareto Front (PF) by efficiently selecting batches of experiments.

- Develop and assess a potential stopping criterion methodology to accompany the algorithms designed throughout this thesis, with the purpose of early termination of experimentation once accurate surrogate models are determined.

1.3 Thesis Structure

The structure of this thesis is built upon the initial exploration of Bayesian Optimisation (BO) techniques as an alternative DoE methodology for expensive AM problems. This is to discern if literature Bayesian Experimental Design Optimisation (B-DoE) techniques are a suitable framework for AM problem characteristics discussed in Section 1.1, and if so develop new and novel methods to tackle novel AM problem characteristics.

- Chapter 2 reviews and introduces the background knowledge and history of the literature regarding DoE and the optimisation frameworks used in correspondence with DoE. This chapter is meant to serve as an introduction to the basis of knowledge and background history to support the research that is carried out in later chapters.

- Chapter 3 provides the theoretical knowledge of the optimisation tools used throughout the thesis. This will include an introduction of the metrics used to assess performance of developed methods as well as their comparison against methods published in literature. The set-up for the global optimiser used throughout the thesis: Genetic Algorithm (GA) and the methodology used when tuning the surrogate models parameters.

- Chapter 4 introduces the sequential B-DoE framework onto a series of representative synthetic benchmarks as well as an AM problem to assess the methods available in literature and their suitability to the problems goals. The synthetic simulations, AM problem application and theoretical analysis will be analysed to discern the most suitable set-up for further extensions of B-DoE for application in AM in Chapters 5/6.

- Chapter 5 using the insights from Chapter 4 aims to extend B-DoE into a non-greedy batch selection (BB-DoE) that is not as computationally complex as literature non-greedy methods but is of comparable performance to both greedy and non-greedy literature methods. This developed framework is then assessed and compared by implementation on a set of benchmark functions and a manufacturing case study.

- Chapter 6 seeks to further extend upon the work in Chapter 5 by introducing a second beneficial extension for AM, Multi-Objective handling capabilities. This extension develops the MOBB-DoE framework which draws inspiration from MOGA a rank-based multi-objective evolutionary algorithm to solve batch multi-objective problems, which is then applied onto a benchmark function to assess performance.

- Chapter 7 concludes the thesis. The development of the BB-DoE and MOBB-DoE frameworks are summarised and their application to AM problems. The limitations encountered throughout the thesis as well as the potential future avenues for research are also presented.

1.4 Research Contributions

The primary research contributions of this thesis are as follows:

- The identification of a suitable data efficient sequential DoE methodology for application onto expensive Additive Manufacturing (AM) problems which can discern which experiments are of greater potential value in improving process models and optimisation of process parameter settings in an iterative manner to fully exploit experimental information as it is obtained.

- Extending and designing a novel Non-Greedy Batch Bayesian Experimental Design Optimisation (BB-DoE) methodology capable of selecting multiple experiments simultaneously that weights the contributions of each experiment against other members of the batch set, based upon the data efficient sequential DoE methodology.

- The implementation of a novel Non-Greedy Batch Bayesian Experimental Design Optimisation (BB-DoE) methodology onto a blind AM DoE problem: Optimisation of micro structural property Dendritic Arm Spacing (DAS) in nickel super-alloy on a Directed Energy Deposition (DED) process. Investigating the process interactions between the Laser Power, Nozzle Velocity, and Hatch spacing with the DAS.

- Extending and designing a novel Multi-Objective Batch Bayesian Experimental Design Optimisation (MOBB-DoE) methodology which selects batches of experiments in a data efficient manner to provide an accurate, spread and diverse Pareto Optimal Set (POS) close to the true Pareto Front (PF) based upon the novel Non-Greedy BB-DoE methodology.

- Designing and assessing a novel DoE stopping criterion to determine when to terminate the DoE methodology in the event of accurate surrogate model(s) being found.

A paper is currently being prepared for review prior to publication at the time of submission.

Chapter 2

Literature Review

This chapter aims to introduce the main background concepts and an overview of literature relevant to the research conducted in this thesis which will be covered in the coming chapters. The main topics that will be discussed in Chapter 2 will be in the field of Design of Experiments or Experimental Design (DoE) and its use in manufacturing industries, Bayesian Experimental Design Optimisation (B-DoE) and Extensions to B-DoE suitable for manufacturing problem characteristics including: batch selection and multi-objective optimisation.

2.1 Manufacturing DOE

2.1.1 Design of Experiments

Manufacturing industries are entering the 4th industrial revolution or industry 4.0, which aims to integrate automation into the manufacturing process leading to rapid advancements in productivity, customised production and increased control over the entire value chain of products [222]. One of the nine pillars of Industry 4.0 is *Big Data* which encapsulates the five v's of data: Volume, Variety, Veracity, Velocity and Value of data [222]. One of the challenges facing Industry 4.0 is the analysis of this Big Data, both historical and new manufacturing process data analysis is needed to achieve improvements in efficiency [222].

Manufacturing industries are constantly innovating new products, processes and even new branches of manufacturing fields to deliver new products and services that have previously never existed. Even when innovation is not occurring manufacturing companies seek to continually improve upon products and processes that have been previously developed to provide higher quality, lower costs and reduce variability [123]. The fundamental key to both innovation and improvement of products/processes is knowledge and understanding of underlying interactions and mechanisms of processes being utilised in order to harness their full potential [123].

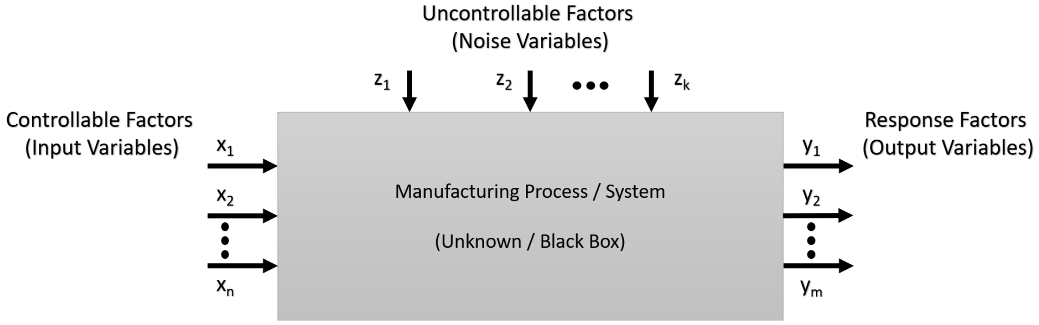


Figure 2.1: Block Diagram of manufacturing process to visualise DoE method.

An illustration of a manufacturing process is shown in Figure 2.1. Figure 2.1 represents a manufacturing process as an unknown mechanism (old or new), where there is no knowledge on the inner mechanisms with controllable input variables and observable response variables [10]. Therefore, in order to understand the processes underlying mechanisms, experimentation is required to vary the controllable input variables, to discern the underlying relationships with the desired output response variables, whilst under uncertainty caused by uncontrollable variables [10].

The methodological approach that forms the basis for how experimentation is performed in manufacturing as shown in Figure 2.2. Figure 2.2 details the thought process when tackling a new manufacturing problem; determining goals, selection of responses, factors, and levels, designing experiments, analysis of results and testing of the experiments hypothesis [94]. In this thesis we focus on the 4th step in Figure 2.2, in which Engineers and Scientists utilise a statistical methodology known as DoE defined in Definition 2.1.1. Whereby the goal of DoE in manufacturing is to select experiments systematically in order to:

- Maximise information gain of underlying processes.
- Minimise the total number of experiments, and subsequently cost.
- Develop accurate and representative models.
- Optimise process parameters of desired manufacturing processes.

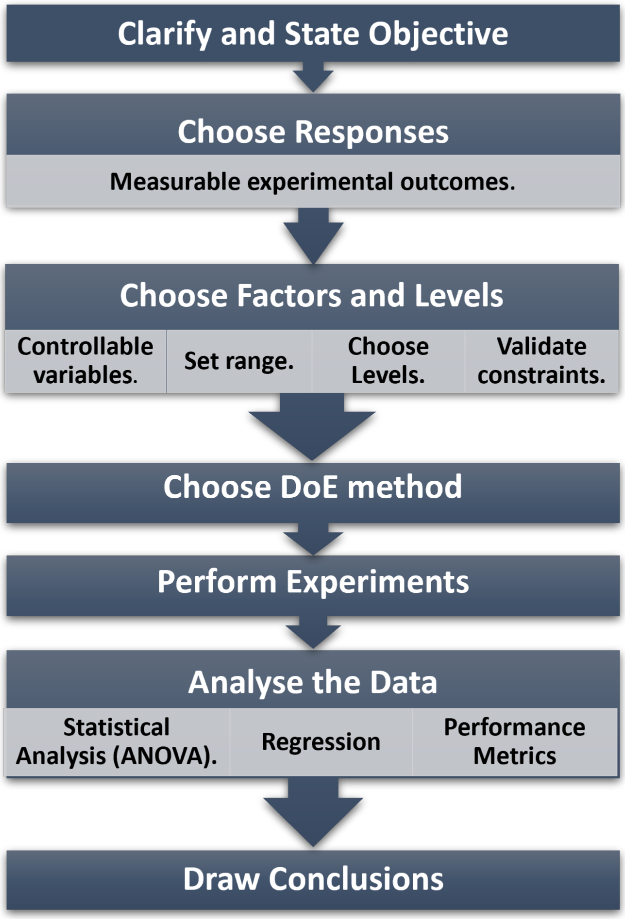


Figure 2.2: Flowchart of steps taken in order to define the DoE.

Definition 2.1.1 (Design of Experiments).
A systematic statistical method that seeks to determine the relationship between the controllable (input) factors affecting a process and their influence on key (output) responses. A structured approach for the collection and analysis of data typically in the form of experiments [164].

DoE is a statistical method unlike mathematical modelling, as the key difference is their representation of underlying mechanisms. Mathematical modelling seeks to produce a model that replicate the physical relationship of the complex mechanisms, whereas DoE produces a statistical correlation between factors (controllable variables) and responses (output variables) [8]. DoE can also be further categorised by the purpose of the experiments into 5 fields of study [94]:

1. Comparison Studies:

If a choice between multiple options is required, you can design an experiment to choose the best option to run. An example of such a comparison could be several nozzle designs for a AM process, will each design provide a similar production lifetime? If they're different, how do they differ and which is the best choice.

2. Variable Screening

   The Pareto Principle in DoE [164] states that most of the performance is controlled by a few important controllable variables. Thus, a variable screening DoE can be used to reduce the complexity of the DoE by selecting/removing controllable variables to a reduced set. This ensures that the variables chosen are most likely to have the largest contribution in effecting the performance of the response variables.

3. Transfer Function Exploration

   The relationship between the controllable variables and the response variables is referred to as the *Transfer Function* [94]. Therefore, once a set of controllable variables has been chosen, DoE can be used to efficiently select experiments to explore the design space to study the underlying relationship.

4. System Optimisation

   Once a transfer function and/or model has been identified DoE can be used to efficiently select experiments sequentially in order to identify the optimal settings of the controllable variables. The DoE achieves this by selecting experiments that move towards the region of interest, for improving the processes product quality and reliability in an efficient manner.

5. System Robustness

   After identifying the underlying process interaction through transfer function exploration and identifying the optimal controllable system parameter settings through system optimisation. Another important task that DoE can handle is to reduce the uncertainty in the system by making it robust to uncontrollable variables such as environment conditions and noise variables via Robust Design (RD).

Two of these 5 categories (3 and 4) have been grouped together for study in literature under the name Response Surface Methodology (RSM), for which this will be the sub-category of approaches that will be built upon throughout

Table 2.1: Commonly used terminology in Design of Experiments and their descriptions [164].

Experimental Design Terminology	
DoE Term	Description
Response	A response is the dependant (output) variable(s) that is to be evaluated by varying the controllable factors of interest. There can be multiple responses studied in one experiment.
Factor	A factor is the independent (input) variable(s) that are varied to assess their effects on the response. There can be multiple factors studies in one experiment.
Level	Each factor is varied at pre-specified discrete intervals (levels) during experiments on a continuous domain. The number of levels considered will affect the level of precision obtained from experiments.
Orthogonality	Orthogonality can be expressed as factors independence such that it guarantees that the effect of one factor or interaction can be estimated separately from the effect of any other factor or interaction in the experiment.
Blocking	Blocking is a method for increasing precision by removing the effect of known uncontrollable factors (e.g. batch-to-batch variability).
Randomisation	Randomisation protects against unknown bias distorting experiments (e.g. measurement drift with each successive experiment).
Replication	Replication is the the repeat of an experiment to increase the precision in the response from uncontrollable factors.

this thesis. Throughout Section 2.1.1 and the remainder of the thesis some key terms will be used repeatedly and so shall be defined further in Table 2.1: Response, Factor, Level, Orthogonality, Blocking, Randomisation and Replication.

2.1.2 Brief History of Design of Experiments

The most common DoE method proposed in early work focused on a One-Variable-at-a-time (OVAT) approach which often could be thought of a trial-and-error approach to experimentation. This method was brought to its peak through the work of Thomas Edison's *Trial and error* methods [10]. Whereby all controllable variables that are chosen for the experiment are set to a constant value except for a single variable, for which this variable is varied with each experiment at a variety of levels. This is then repeated for each controllable variable involved in the experiment.

However, despite OVAT popularity due to its simplicity it came with a variety of problems and limitations including over-experimentation, lack of variable interaction analysis, and luck-based results. As the industrial revolution began to pick up steam, industrial research could no longer utilise/implement the generic OVAT approach with its obvious limits. The next spark that began the field of research we now know of today as *Design of Experiments* was due to the work of Fisher's in Agriculture in the 1930's. In R.A. Fishers influential work [70], they found carrying out experiments improperly (trial and error approach) had a negative impact of the analysis of the data. This coupled with the nature of the field of work, agriculture requires large areas of land, long times between set-up to completion and a wide variety of both controllable and uncontrollable factors to consider. Therefore, implementing OVAT approaches would negatively impact experimentation, which lead to the development of early DoE methods [4].

The next big push in DoE literature began in the 1950's with Box and Wilson [36], where after the 2nd World War they were attempting to solve problems within the chemical industries at Chemical Imperial Industries [234]. Their inspiration arose from the dissimilarities of chemical process industries in comparison to agriculture in two main ways. Firstly, the response output variable can be observed almost immediately (referred to as, immediacy) and secondly, the experimenter can learn crucial information from a small number of runs in order to set which experiments to run next (referred to as, sequentiality) [164]. Another difference was in the experimentation cost namely the cost per run for process industries was much greater than in agriculture [234]. Thus Box and Wilson [36] began developing statistical modelling (transfer function exploration) and process optimisation methodologies which were later grouped together under Response Surface Methodology (RSM) as mentioned in Section 2.1.1.

In the 1950's the term 'Made in Japan' was often regarded to as a product of low quality but, in the late 1950's W Edward Deming taught the importance of statistical quality control including Design of Experiments to Japanese Scientists and Engineers [10]. One such scientist that became the pioneer of this era/field was Genichi Taguchi who advocated for using DoE for what termed *Robust Parameter Design* [164]. This ushered in the Second Industrial Era or Quality Revolution which primary focuses were on [164]:

1. Make processes insensitive to uncontrollable factors.
2. Make products insensitive to variation from components.
3. Finding levels of controllable factors that force the response factors to a desired value whilst simultaneously reducing variability around this value.

Taguchi [211] suggested a variety of statistical methods to achieve these goals to which he referred to as *Orthogonal Arrays* but were more often referred to as *Fractional Factorial Designs* in the west. One of the first companies to apply his methods were Toyota. His work generated much discussion and controversy due to its lack of peer review but by the late 1980's it had been peer reviewed and more widely accepted. The benefit of this controversy led to an increased interest in the field of DoE not only in manufacturing industries but a variety of industries [164].

Finally, the most recent push in the field of DoE is regarding the *Six Sigma* quality initiative which was launched by Motorola in the 1980 – 1990's [4]. Whilst Taguchi simplified DoE for managers to minimise costs, Six Sigma packaged DoE into a framework for adoption by companies on a larger scale [87]. Where DoE slotted into its improvement phase of its Design Measure Analyse Improve and Control (DMAIC) methodology [8]. Goh [87] argued that Six Sigma took DoE further in a variety of ways:

- Extending its application to transactional processes as well as physical processes.
- Focusing on its impact to attract the attention of CEO and management.
- Attaching certification system with its training using a martial arts belt system (yellow, green and black belts).

Whereas in the history of DoE there have been many developments into opening new fields of study and increased benefits associated with each field, the Six Sigma methodology is seeking to package DoE as part of a greater method for the wider adoption of larger companies [8].

There have been many developments in DoE in recent history when seeking to develop understanding between controllable inputs and measurable outputs typically when analysing a physical process. Alongside the advent of computer technology and its rapid advancement has also opened other alternative branches of modern DoE such as non-deterministic DoE methods which sought to select experiments that filled the design space as uniformly as possible [79].

2.1.3 Non-Deterministic DoE

The early development of non-deterministic DoE methods borrowed inspiration from traditional DoE methods which didn't account for knowledge of the underlying systems such as OVAT or Factorial Designs. Therefore, the early non-deterministic DoE methods sought to fill the design space of interest as uniformly as possible [79]. This was primarily achieved through the random placement of experiments within the design space of interest.

One of the first pseudo-random non-deterministic DoE methods was Monte Carlo Sampling (MCS) [158] which utilised pseudo-random numbers to generate a pre-specified number of random sample point within the design space of interest. The intention behind these initial approaches were that by perform random actions, eventually the design space would be filled. However, depending upon the sample size the random space filling could result in clustering of samples in section of the design space and sparsity in others resulting in under-representation [79].

In order to alleviate the issues caused by MCS, researchers proposed Stratified Monte Carlo Sampling (SMCS) which divided the design space into non-random strata (sub-sections) and applying MCS to each strata [79]. Another alternative proposed by researchers are the Quasi-Monte Carlo Sampling (QMCS) which use a quasi-random low discrepancy (QRLD) sequences to generate samples [79]. In these cases a low discrepancy sequence are sequences that generate random points within a design space that minimise large areas of low and high density of random points resulting in a more spread distribution. There have been a variety of QRLD sequences used in QMCS including: Halton [97], Hammersley [98], and Sobol [205] which provide a more evenly distributed random sequence than pseudo-random sequences such as MCS.

Space-filling designs are another non-deterministic DoE methodology which seek to construct a systematic uniform grid of sample point within the design space [79]. Johnson et al. [115] proposed two designs: minimax and maximin designs that placed sample points with a pre-defined design space based upon the distance between each sampled point. The maximin is a distance based design which places samples that maximise the minimum distance between every other point in the design space. Whereby samples are placed as far as possible from the other as shown in Figure 2.3b. Whilst on the other hand, the minimax is also a distance-based design which places samples as close together as possible whilst maintaining maximum coverage of the design space, as shown in Figure 2.3a.

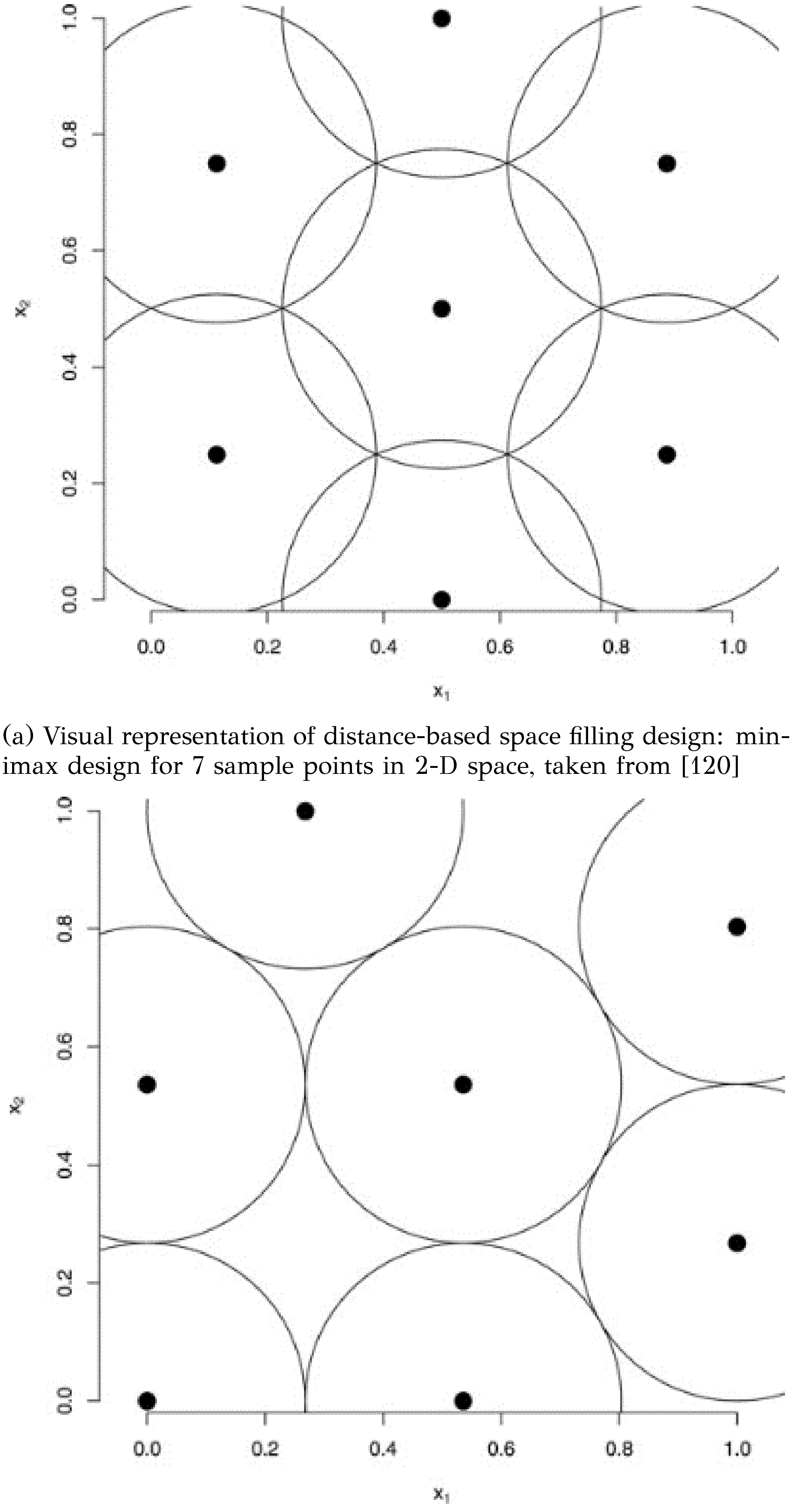
One of the more popular space-filling design developed and researched within literature is the Latin Hypercube Design (LHD) [155]. A LHD works by dividing the design space into bins of equal size based upon the number of samples n_s for which their will be an equal number of bins along each dimension of the design space as illustrated in Figure 2.4.

Once the design space has been partitioned into the equally sized bins there will be an equal number of bins along each dimension of the design space as shown in Figure 2.4 with $n_s = 7$ in a 7x7 grid on the 2-D design space. Samples are then placed within the LHD grid in the design space such that for each column and row (in 2-D) of bins, no two samples are placed which is known as the non-collapsing design condition [79]. Due to its formulation as well as the non-collapsing design condition, LHD performs well over a range of dimensions but it is not without its limitations. One limitation is the sampling of design points using LHD is still random which in some cases can lead to poor space-filling designs shown in Figure 2.5 which has $n_s = 10$ all placed along the diagonal of the 2-D design space.

Therefore, since LHD publication a variety of researchers have been improving the core LHD with various features such as: maximising the minimum distance between sample points [166] and orthogonal array-based LHD [240].

2.1.4 Use of DoE in Manufacturing

As shown in Section 2.1.2 despite DoEs origins over the years of its initial development of its five sub-classifications it has been applied to a variety of fields involving both manufacturing and non-manufacturing fields. In [61] the authors analysed 20 years of DoE research and showed that there had been steady growth in the application of DoE in literature that has been increasing with time. With medicine (18%), engineering (10%), biochemistry (10%), physics (7%) and computer science (6%) equating to 50% of the total literature. The primary focus of DoE regarding this thesis is in the field



(a) Visual representation of distance-based space filling design: minimax design for 7 sample points in 2-D space, taken from [120]

(b) Visual representation of distance-based space filling design: maximin design for 7 sample points in 2-D space, taken from [120]

Figure 2.3: Distance-based space filling designs: minimax and maximin for 7 sample points in 2-D space, taken from [120].

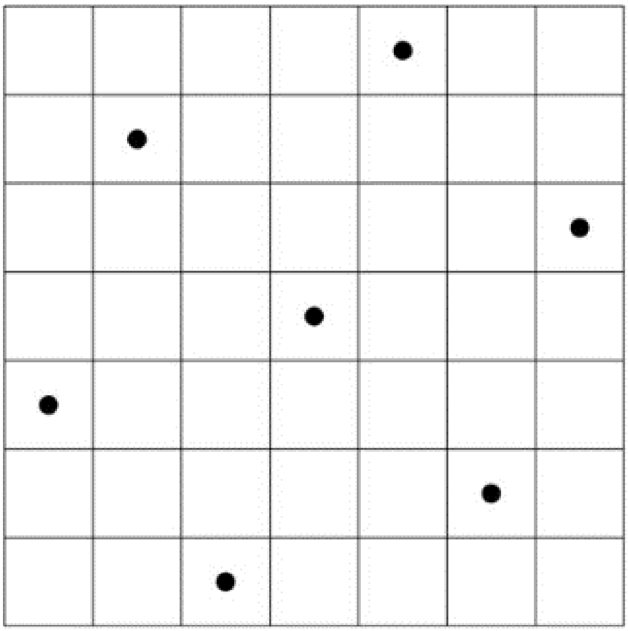


Figure 2.4: A Latin Hypercube Design (LHD) in 2-D space using $n_s = 7$ samples uniformly distributed, taken from [120].

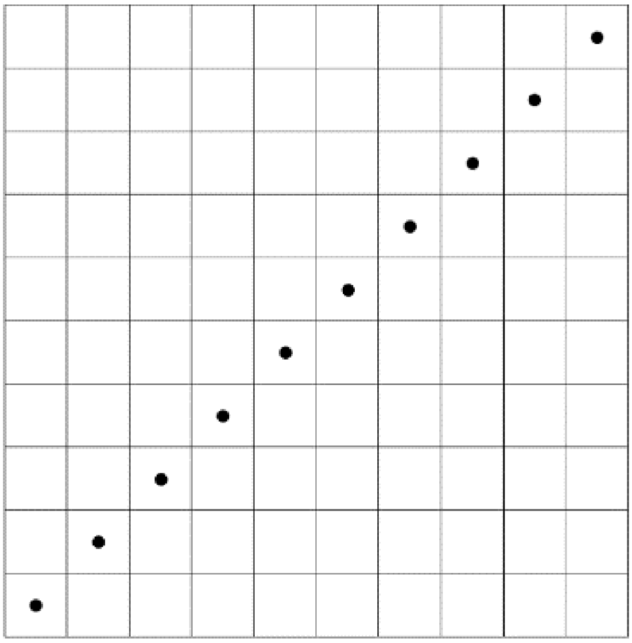


Figure 2.5: A Latin Hypercube Design (LHD) in 2-D space using $n_s = 7$ samples distributed only the diagonal, taken from [79].

of manufacturing industries/engineering with a particular focus on Additive Manufacturing (AM).

Ilzarbe et al. [109] provides a snapshot of the application of DoE in literature in the field of engineering over the period of 2001-2005, which offers good overall insight into how researchers are utilising the statistical tools that have been developed thus far. In this study they looked at practical application papers in a variety of engineering fields totalling 77 application papers looking at a selection of metrics of how DoE was used.

The first metric to consider is the types of DoE utilised previously introduced in Section 2.1.1, in which there are five categories of DoE which where grouped into three in this paper: Traditional (Comparison + Variable Screening), RSM (Modelling and Optimisation) and Robust Design (RD). The first insight gained is that traditional methods were implemented in most applications studies at 58% with 33% for RSM and 9% for RD [109] a decade ago. This suggests that despite the availability of powerful statistical methodologies researchers are often opting for a more simplistic approach.

Another important characteristic to be considered is the number of controllable variables (or factors) to be assessed, which in a manufacturing setting can initially a large variety of factors. This may be due to the need to account for process operation parameters and materials characteristics as well as their interactions which can lead to complex multi-factor problems. Yet Ilzarbe et al. [109] found that on average the number of factors considered were 5 factors and 71% of the studies considered 2-5 factors. Manufacturing and engineering problems typically would be presumed to have a large number of factors to consider during DoE due to multiple controllable inputs and uncontrollable factors. However, in [109] results were contrary to this hypothesis suggesting researchers were screening factors prior to publication by either using screening designs or prior knowledge. This screening could have been implemented using expert knowledge or DoE methods as “most of the performance is controlled by a few important controllable variables” as stated by the Pareto principle of DoE [164].

The final characteristic to be considered from [109] was the types of DoE methods implemented. Of which there were four main designs considered: Taguchi orthogonal arrays (31%), Full Factorial Designs (16%), Fractional Factorial Designs (14%) and Central Composite Designs (9%). This is further corroborated in AM in which Durão et al. [62] noted the large majority of DoE methods implemented used either a Taguchi orthogonal array or a factorial design, whilst RSM were utilised less frequently.

Both Durakovic [61] and llzarbe [109] have shown the breakdown of the increasing use of DoE in literature and characteristics involved in the application of DoE in engineering. However Tanco et al. [213] presents a survey of 760 manufacturing companies in Basque Country of which 18% responded showing a disconnect between literature developments and industry application. It showed that 80% of the companies implemented an OVAT strategy, Moreover, only 20% carry out experimentation using a pre-established DoE method [213]. There was also a dependency of industry size where 18% of small-medium sized companies applied DoE and 29% at large industries. Taguchi methods were applied at the same rate regardless of company size, likely due to simplicity in application. Finally, Tanco et al. [213] asked its respondents what they believed the biggest barriers to uptake of DoE to which the following split of responses were received:

- Theoretical ignorance of DoE for real applications (43%).
- Absence of a clear methodology to simplify implementation (37%).

2.1.5 Common DoE used in Manufacturing

Despite the many advances in the field of DoE, manufacturers often rely on the simpler statistical implementation schemes as pointed out by Ilzarbe et al. [109] which are: OVAT, Full Factorial, Fractional Factorial and Taguchi Orthogonal Arrays in a variety of industries.

One-Variable-at-a-time (OVAT)

One variable at a time method has a simple implementation methodology, where the experimenter has chosen an array of factors and several levels to vary each factor by. The experiments are then chosen by setting all factors to a constant level while varying one factor for the levels chosen. OVAT methods main advantage is their simplicity in implementation in comparison to alternative statistical methods. However, this is usually the only advantage and it should only be chosen in scenarios where experimentation costs are cheap.

In contrast the disadvantages occur when experimentation costs are higher and factor interactions are of greater importance. As in most scenarios it is more important to vary factors simultaneously in order to access the factors interactions and their effect on the response. Another disadvantage is the large number of runs required to achieve the same level of precision as more complex DoE methods, which can often miss optimal settings of factors.

Table 2.2: Example of OVAT approach for selecting experiments for 2 factors at 2 levels.

One-Variable-at-a-time (OVAT) DoE for 2^2 Design		
Experiment Index	Temperature (°C)	Pressure (Pa)
1	50 (T_1)	250 (P_1)
2	100 (T_2)	250 (P_1)
3	50 (T_1)	500 (P_2)

Table 2.2 illustrates a simple OVAT example in which two factors: Pressure (P) and Temperature (T) are varied for two levels each, in order to search for an optimum. In Table 2.2 the temperature and pressure are held constant at levels: T_1 and P_1, resulting in 3 experiments but, as only 1 factor is varied at a time not all interactions are considered.

Despite the disadvantages of OVAT listed above, the methodology is one of the more popular choices in engineering and manufacturing applications [213] and has been applied in a variety of industries: Materials Science [190], Food Sciences [66], Chemical [156], and Mechanical [141].

Full Factorial Design

Whilst a OVAT approach chooses to vary a single factor at a time, a full factorial design consists of performing all possible combinations of the factors and levels involved. This difference can be illustrated by looking at the same example shown in Table 2.2 in Section 2.1.5, looking at Pressure (P) and Temperature (T) at two levels each.

Table 2.3: Example of Full Factorial approach for selecting experiments for 2 factors at 2 levels.

Full Factorial DoE for 2^2 Design		
Experiment Index	Temperature (°C)	Pressure (Pa)
1	50 (T_1)	250 (P_1)
2	100 (T_2)	250 (P_1)
3	50 (T_1)	500 (P_2)
4	100 (T_2)	500 (P_2)

Comparatively a full factorial design is an improvement on OVAT methods as it also accounts for the analysis of factor interaction effects, which is the primary advantage. As long as the number of factors and levels to be used have been chosen, the method of implementation is simple to use. Although, by accounting for factor interaction effects the total number experiments required can increase dramatically based on the combination of factors and levels considered which is the methods primary disadvantage. For example, using Pressure and Temperature, $2^2 = 4$ experiments however, as the number of factors and levels increase the number of experiments grows drastically such as:

- Increasing factors at low levels e.g.,

  5 factors at 2 levels is $2^5 = 32$ experiments.
- Increasing levels at low factors e.g.,

  2 factors at 5 levels is $5^2 = 25$ experiments.
- Or increasing both levels and factors e.g.,

  5 factors at 5 levels is $5^5 = 3125$ experiments.

These issues are then amplified further when repetitions are required in some manufacturing cases due to variability. Even in the event of requiring many runs the Full Factorial design is the 2^{nd} most utilised DoE in engineering literature [109] and has been applied in a variety of industries: Additive Manufacturing [105], Logistics [75], Material Science [189], Chemical [224], Automotive [69], and Food Industry [241].

Fractional Factorial Design

A fractional factorial design is a reduced size factorial design that acts as a screening design to determine which factors and their interactions have the most influence on the response. This allows the experimenter to reduce the total number of factors that then need to be considered in a later full factorial design. In order to do this it follows the sparsity-of-effects principle [164], which states that a system is usually dominated by main factors or two factor interactions and so higher-order interactions can be ignored. For example in Table 2.4 a 3 factor 2 level fractional factorial design is set-up, by using the sparsity of effects principle it can be assumed that the effect of the ABC interaction is negligible and thus can be ignored. Therefore, as ABC is investigated at two levels initially one of the levels can be removed and in this example level 2 is removed as shown in Table 2.4a to Table 2.4b.

As can be seen in Table B as the 3^{rd} order interaction was removed it only needs to stay at a single level. Also notice factors AB and C have identical

Table 2.4: Example of Fractional Factorial approach for selecting experiments for 3 factors at 2 levels (1 and 2).

Full Factorial DoE for 2^3 Design							
Experiment Index	A	B	AB	C	AC	BC	ABC
1	2	2	1	2	1	1	2
2	1	2	2	2	2	1	1
3	2	1	2	2	1	2	1
4	1	1	1	2	2	2	2
5	2	2	1	1	2	2	1
6	1	2	2	1	1	2	2
7	2	1	2	1	2	1	2
8	1	1	1	1	1	1	1

(a) Full Factorial design before reduction

Fractional Factorial DoE for $2^{3\text{-}1}$ Design							
Experiment Index	A	B	AB	C	AC	BC	ABC
2	1	2	2	2	2	1	1
3	2	1	2	2	1	2	1
5	2	2	1	1	2	2	1
8	1	1	1	1	1	1	1

(b) Fractional Factorial design after removing 3^{rd} order interactions

levels, this is known a confounded or aliased effect. By using the results of the experiments with Analysis of Variance (ANOVA) the statistically significant factors can be identified for which a full factorial design can then be used.

The main advantage of using a fractional design is to reduce the total number of experiments required to determine which factors are the most significant. Whereas the main disadvantage is this is often used in conjunction with other DoE methods as a screening method and so is not a stand-alone technique. Just as with full factorial designs, fractional designs have been applied in a variety of industries: Chemical Industry [67], Food Industry [148], Mechanical [200], and Manufacturing [200].

Orthogonal Array's

The Taguchi Orthogonal Arrays are a type of highly fractionalised factorial design proposed by Genichi Taguchi [211], which are similar to fractional factorial design but incorporate the orthogonality property as described in

Table 2.1. After factors and levels have been chosen the DoE is chosen from a pre-defined list of designs using an array selector table [122] see Table 2.5.

Table 2.5: Taguchi Orthogonal Array Design Selection Array

		Number of Parameters								
		2	3	4	5	6	7	8	9	10
Number of Levels	2	L4	L4	L8	L8	L8	L8	L12	L12	L12
	3	L9	L9	L9	L18	L18	L18	L18	L27	L27
	4	L16	L16	L16	L16	L32	L32	L32	L32	L32
	5	L25	L25	L25	L25	L25	L50	L50	L50	L50

As with fractional factorial designs the advantages of this method is its easy application and large reduction in the total number of experiments to be performed. However, in contrast its disadvantages it shares some of the same disadvantages of being a screening method, it also does not account for every factor relationship. By not accounting for every relationship up until at least second order interactions, it leads to difficulty in determining if some factor interactions have a significant effect on the response. In this scenario it should not be applied where all process interactions are required. Despite these drawbacks the Taguchi method is one of the most frequently applied designs (31%) in manufacturing literature and has been used in: Additive Manufacturing [245], Logistics [194], Manufacturing [201], Mechanical [43], and Chemical [252].

2.1.6 Additive Manufacturing

The future of manufacturing is encapsulated by the concept of *Industry 4.0*, which combines the advancements in machine learning and automation to switch from a mass production model to a customised production model [222]. Industry 4.0 can be broken down into nine core principles: Augmented Reality, System Integration, Cloud Computing, Big Data, Internet of Things (IOT), Additive Manufacturing, Cyber Security, Autonomous Robots, and Simulation [60].

For the smart factories of Industry 4.0 consumers will have constantly changing demands increasing the challenge of producing high quality and high-performance products for individualised use rather than mass production of generalised products [222]. Additive Manufacturing provides a beneficial framework for tackling these issues through its capability in producing products with complex geometries with unique micro structures. Additive Manufacturing fabricates products layer by layer using a computer aided design

(CAD) file, allowing the fabrication of complex parts over a shorter production cycle [162]. Therefore, Additive Manufacturing is currently being used to supply these specialised products to a variety of industries including: Biomedical [130], Aerospace [220], Construction [235], and Automotive industries [221].

Despite its vast manufacturing potential for application, its adoption into manufacturing industry is being hindered by a limited materials library, formation of various types of defects during processing, and inconsistent product quality [226]. These issues stem from a lack of understanding of the underlying mechanisms which can be explored in process parameter optimisation using design of experiments [226].

Process Parameter Optimisation

Process Parameter Optimisation (PPO) in additive manufacturing is a crucial step for guaranteeing consistently well-developed products by minimising variation during the processing leading to defect formation at both mesoscale and macroscale [226]. Mesoscale properties are attributes at the microscale that may not be easily constructed from properties at the atomic scale [186]. Mesoscale properties in additive manufacturing that are of common concern include: porosity (void fraction of material), lack of fusion (lack of binding between melted layers), and cracks (fractures caused by thermal strain in rapid heating and cooling of materials) [39]. Whilst macroscale properties refer to a materials mechanical properties such as: tensile strength, yield strength, and hardness [226], which are dependent on the mesoscale properties.

Traditionally, PPO literature implements experimental design methodologies mentioned in Section 2.1.5 [226]. This was illustrated in the work of Mohamed et al. [162], which reviewed the PPO literature in a popular AM process, Fused Deposition Modelling (FDM) [170]. Mohamed et al. [162] found Taguchi Designs with an ANOVA analysis to be a dominant choice among researchers in determining which factors were significant with a substantial reduction in the total number of experiments. This reduction is achieved when factor interactions are confounded with each other which leads to issues in the determination of the optimal process parameter settings, as shown in a variety of FDM studies [7] [169] [245] [132].

Whilst Taguchi methods provide reductions in both factors and total experiments required, they do not satisfy other goals and requirements of AM when performing PPO. Other goals when performing PPO include the generation of high quality fitted models of the process being explored [162]. Whilst there are a variety of requirements for AM problems to tackle [162]:

- Expensive experimentation.
- Large number of factors.
- Batch processes.
- Constrained processes.
- Multiple conflicting response criteria.

Therefore, the use of traditional Taguchi methods in these scenarios is not a suitable approach to tackle these issues. Rather another sub-category of DoE offers more beneficial properties for addressing some of these requirements: Response Surface Methodology (RSM).

2.1.7 Response Surface Methodology (RSM)

Response Surface Methodology (RSM) in essence is a combination of design and analysis of experiments, process modelling, and optimisation [125]. Figure 2.6 depicts the generalised approach of RSM methods in which an initial screening design is used, typically a fractional factorial or Taguchi design to explore as many factors in as few runs as possible [125]. Once these screening experiments are performed an ANOVA analysis is used to identify the factors which have a significant effect on the response factor [125]. Upon reducing the total factors to evaluate a RSM experimental design is chosen. The most commonly applied design is: Central Composite Design (CCD) and Box-Behnken Design (BBD) [168]. Experiments are then performed according to these designs in the next step to generate a data set of responses from which surrogate models can be fit to the experimental data.

Surrogate modelling is used to provide a low-cost approximation of underlying processes. This offers a cheaper means of approximating responses to various experiment settings and thus cheaper optimisation of the underlying process [92]. However, the performance of the optimisation is dependent on the choice of the surrogate model according to the assumptions of the underlying process and the subsequent precision of the surrogate model implemented [33]. RSM typically approximates using a low-order polynomial. If the underlying process is well-modelled as a linear function a first-order polynomial model should be used [149]:

$$f(x) = \eta_0 + \sum_{i=1}^{n} \eta_i x_i + \sum_{i<j} \sum \eta_{ij} x_i x_j + \epsilon \tag{2.1}$$

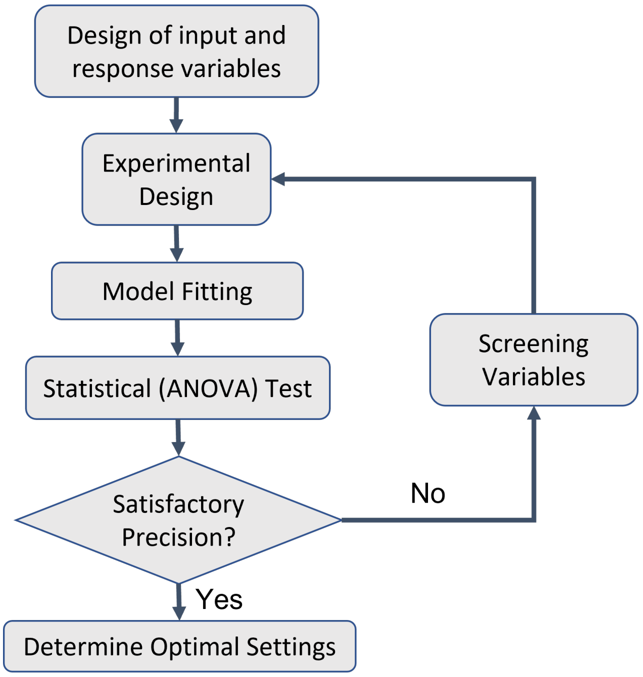


Figure 2.6: Response Surface Methodology flowchart.

Where, η_0 is the unknown constant parameter, η_i is the unknown 1^{st} order parameter for input x_i, and η_{ij} is the unknown 1^{st} order parameter for the interaction effect between input x_i and input x_j. ϵ is the random error. On the other hand, if the underlying process has a non-linear function a first-order model will result in poor model precision. In such cases a RSM design such as CCD which utilise central points can investigate non-linear functions to be approximated using a second order-polynomial model [149]:

$$f(x) = \eta_0 + \sum_{i=1}^{n} \eta_i x_i + \sum_{i=1}^{n} \eta_{ii} x_i^2 + \sum_{i<j} \sum \eta_{ij} x_i x_j + \epsilon \tag{2.2}$$

where, η_{ii} is the unknown 2^{nd} order input parameter.

Whilst a low-order polynomial is unlikely to reproduce the true underlying relationships of the process over a small region of experimentation with a small number of factors, it is a good approximation [149]. Once the surrogate models have been fitted to the experimental data, their predictive precision is assessed using an ANOVA test to quantify whether they have reached a satisfactory level of precision. If a satisfactory level of precision is not achieved, Steps 2 to 6 in Figure 2.6 are repeated to until new models of greater fit are developed [125]. The final step is to implement an optimisation approach to determine the optimal process parameter settings of the process, which are then further verified via experimentation [125].

The development and implementation of RSM in manufacturing did tackle some of the initial requirements of AM process DoE problems through the integration of modelling and optimisation techniques [167]. Yet as mentioned in Section 2.1.6, in manufacturing industries such as AM each experiment can be expensive due to use of excess resources [5]. Also, when coupled with increasing numbers of factors the total experiments required will scale accordingly as will the importance of model precision [5].

Hence traditional RSM methods do not provide the tools necessary for current manufacturing issue in regard to manufacturing DoE. However, alternatives have been introduced in a variety of literature as early as 1933 [214]/1935 [215], as well as by Mahalanobis in 1940 [151]), and finally in regard to experimentation literature in 1952 by Robbins [184] which suggested utilising a sequential selection design scheme.

2.1.8 Adaptive Response Surface Methodology (ARSM)

Sequential designs choose the next experiment settings based upon information provided from previous experiments. The primary benefit of implementing a sequential scheme is their greater adaptability to features that appear during experimentation [92].

These capabilities were integrated into Adaptive Response Surface Methodology (ARSM) after model precision checks are performed as shown in Figure 2.7, in which instead of re-specifying DoE factors as in Step 6 of Figure 2.6 the next experiment is chosen using a defined infill criterion [5]. This cycle iterates through performing experiments, re-fitting the model, assessing precision, and choice of the next optimal experiment until a pre-specified model precision is reached [5]. The efficiency and performance of an ARSM is strongly dependent upon the choice of surrogate model, infill criterion, and optimisation methodology employed [74].

Previously, the primary choice of surrogate models in RSM and ARSM were low-order polynomial a type of parametric model [167]. Parametric models are typically used for the simplistic implementation, training speed and low data requirements [167]. The nature of parametric models defines that the underlying process can be represented by a finite set of parameters, which makes parametric models not very flexible [167].

An alternative choice are non-parametric models which assume that the data can be represented by an infinite number of functions. This in turn

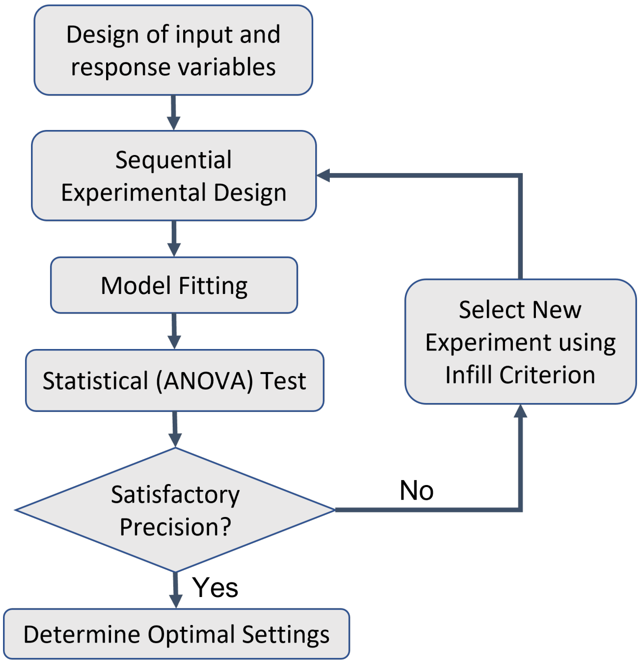

Design of input and response variables
Sequential Experimental Design
Model Fitting
Statistical (ANOVA) Test
Satisfactory Precision?
No
Select New Experiment using Infill Criterion
Yes
Determine Optimal Settings

Figure 2.7: Adaptive Response Surface Methodology flowchart.

allows for greater flexibility in scenarios where knowledge of the properties of underlying functions is unknown [182]. Care is required when tuning non-parametric surrogate model parameters as there can be an infinite number of functions, if handled improperly over fitting the model to the data can occur [182]. Non-parametric surrogate models used in RSM and ARSM include: Gaussian Processes, Thin-plate splines, and neural networks [167] [33] [92].

The next decision in the design of a suitable ARSM approach is the choice of the infill criterion which decides which experiment is optimal to perform by balancing two features; exploration and exploitation [33]. Firstly, exploration ensures that experiments are chosen in areas/regions of factor space that have previously not been explored [92]. Secondly, exploiting in areas of high potential in the factor space where a local or potentially global optimum could be located [92]. The infill criterion chosen should ensure that a reasonable equilibrium between exploration and exploitation can be reached. This ensures a well fitted surrogate model can be produced as well as locating the global optimum for process parameter settings in the problem investigated [92].

2.1.9 Summary of Manufacturing DoE

In summary, AM and manufacturing DoE problems are constantly seeking to improve their understanding upon current and new manufacturing processes. This improvement in understanding is achieved through systematic and efficient experimentation following structured approaches known as Design of Experiments or Experimental Design (DoE), which aim to:

- Maximise information gain of underlying processes.
- Minimise the total number of experiments, and subsequently cost.
- Develop accurate and representative models.
- Optimise process parameters of desired manufacturing processes.

Whilst choosing a suitable DoE approach that is capable of tackling the AM problems characteristics:

- Expensive experimentation.
- Large number of factors.
- Batch processes.
- Constrained processes.
- Multiple conflicting response criteria.

Therefore, the most suitable class of DoE that satisfies the DoE aims is an ARSM. The model choice should be a non-parametric model for increased flexibility against varied unknown problem types. An appropriate infill criterion that balances between exploring regions of interest and exploiting potential globally optimal regions. Finally, a global optimiser methodology of the infill criterion to ensure the entire factor space is explored to locate the global optimum. One such approach that has garnered attention from the literature that suits these requirements is Bayesian Optimisation (BO) using Gaussian Process Regression (GPR) surrogate models [92].

2.2 Bayesian Optimisation

Bayesian Optimisation (BO) is a sequential model-based approach for finding the global optimiser for an unknown response function that is expensive to evaluate [40]. BO can be applied to a variety of optimisation problems where there is no closed-form response representation, but where the response can be observed (possibly noisy) in the event of providing a sample or experiment settings [40]. The aim of BO is shown through Equation 2.3, where each experiment selected aims to maximise its expected utility towards locating the global optima and building a representative surrogate model:

$$x_{t+1} = \arg\max_{x \in R} u\left(x_t | D_{1:t}\right) \qquad D_{1:t} = \left[\mathbf{x}_{1:t}, \mathbf{y}_{1:t}\right] \tag{2.3}$$

Where x_{t+1} is the next experiment to be selected. R is the design space of interest (feature space). $u(\ldots|\ldots)$ is the expected utility or acquisition function to be optimised. x_t is an experiment comprising of d inputs, where d is the total number of input variables. $D_{1:t}$ is the currently observed data set which contains pairs of experiments and their observed responses for all experiments run up until the interval t.

BO is known as a data-efficient approach for the number of function evaluations required [119]. This efficiency stems from the use of prior beliefs in DoE approach. A prior belief is the understanding or expert knowledge of a particular problem that can be encoded into the optimisation procedure to help guide the search of the feature space [40]. This prior belief is then sequentially refined with each experiment performed as the surrogate model is updated via Bayesian posterior updating [195].

Bayesian posterior updating is the core of BO, which is based upon Bayes Theorem developed by Thomas Bayes in the 18th century [19]. In this context, Bayes Theorem states that "The posterior probability of a model given data is proportional to the likelihood of the data given the model multiplied by the prior probability of the model over the probability of the data" [19].

$$P(M|E) = \frac{P(M) * P(E|M)}{P(E)} \tag{2.4}$$

Where $P(\ldots|\ldots)$ is the conditional probability of an event given that another event has occurred. $P(M|E)$ is the posterior probability of a model M given evidence E. $P(M)$ is the prior probability of a model M and $P(E)$ is the prior probability of evidence E. $P(E|M)$ is the likelihood of evidence E given model M. However, in the case of BO the probability of the data is fixed and the Bayes theorem is reduced to its simplified form as shown in Equation 2.5:

$$P(M|E) \propto P(M) * P(E|M) \tag{2.5}$$

In this case, the prior represents the beliefs about the behaviour/properties of the feature space for possible response functions [40]. Manufacturers can utilise this property to incorporate their expert knowledge into the modelling and optimisation process to improve its efficiency [195].

There exists a variety of priors that could be used in BO to incorporate prior knowledge however, a set of assumptions were determined to restrict which probabilistic priors were most suitable for use [40]. Initially, priors had to satisfy two assumptions whereby a BO method would guarantee convergence to the optimum if [160]:

- The acquisition function (expected utility) is:
    - Continuous.
    - Minimises the expected deviation from the global optimum at a fixed point.
- Conditional variance converges to zero (or noise) if and only if the nearest observation distance is also zero.

Mockus [160] would later incorporate 3 more assumptions that were deemed natural conditions for selecting a suitable prior: the response is continuous, the prior is homogeneous, and Optimisation is independent of the N-th differences.

Following these assumptions, a Gaussian Process (GP) prior is the most suitable choice [160]. A GP prior is frequently implemented choice of prior [131] [161] [119] [208] [100] where a GP is an extension of the multi-variate Gaussian distribution to an infinite-dimension stochastic process [40].

To summarize, BO can be subdivided into two key components required. Firstly, a probabilistic surrogate model which encodes the data received through experimentation as well as the prior beliefs of the underlying response function [195]. Secondly, an acquisition function which quantifies the expected utility or loss of potential experiments which trade-off between exploring the feature space and exploiting to locate the desired optimum [195]. See Algorithm 2.1 for a pseudo-code representation of the generic BO framework.

Algorithm 2.1 Generic Bayesian Optimisation Framework

```
Inputs:
        $D_t$ : Dataset, $D_t = \left[x_i, y_i\right]_{i=1}^{t}$
        $\lambda$ : Model Hyper-Parameter Priors
        $T$ : Experimentation Budget

GP(m, k) ← $\theta_t\left(\lambda, D_t\right)$                         // Fit GP Hyper-parameters, $\theta_t$
for $t = 1, 2, \ldots, T$ do
    $x_{t+1} = \arg\max_{x \in R \subset R^N} u\left(x|D_{1:t}\right)$       // Optimise Acquisition function
    $y_{t+1} = f\left(x_{t+1}\right) + \epsilon$                          // Evaluate Proposed Experiment
    $D_{t+1} = \left[D_{1:t}, \left(\mathbf{x}_{t+1}, y_{t+1}\right)\right]$               // Augment Data set
    GP(m, k) ← $\theta_{t+1}\left(\lambda, D_{t+1}\right)$               // Update Surrogate Model, $\theta_{t+1}$
end for
```

2.2.1 Gaussian Process Regression (GPR)

Definition 2.2.1 (Gaussian Process Regression).
GPR is a non-parameteric regression to model the predictive distribution $P\left(f_{t+1}|D_{1:t}, \mathbf{x}_{1:t}\right)$ using a GP prior and condition it on training data to model the joint distribution between the observed training data $f_{1:t}$ and the inferred test point f_{t+1} as a GP [182].

Definition 2.2.2 (Gaussian Process).
A GP is a potentially infinite collection of random variables (functions) such that the joint distribution of a finite set of random variables have a consistent joint multivariate Gaussian Distribution [182].

A GP can be fully specified by its mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$ [182].

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{2.6}$$

Where f(.) is the predicted response output for inputs $\mathbf{x}$ using the surrogate model. $m(\mathbf{x})$ is the mean function and $k(\mathbf{x}, \mathbf{x}')$ is the covariance function (kernel) of a GP.

A standard $f(\ldots)$ returns a scalar output at any given input. Whereas, a GP surrogate model will return a mean and variance value of a normal distribution at any given input [40]. The mean and variance represent the GP surrogate models prediction and uncertainty respectively for an experiment with factors $\mathbf{x}_i$ [40]. Therefore, a GP's possible response functions are dictated by the selection and tuning of its mean and variance functions [40].

Gaussian Process: Mean Function

A GP mean function determines the possible off-set in the mean of the response [195]. The off-set can either be set to a constant off-set or can be a variable off-set function [195]. Despite the ability to incorporate expert knowledge into the GP prior through choosing a mean function, in literature for convenience the mean function is often set to a constant $m(\mathbf{x}) = 0$ [40].

Gaussian Process: Variance Function

As the GP mean function is often set to a zero constant, this results in the choice and tuning of the GP variance (kernel) function determining the generalisation properties of the GP surrogate model [63]. A kernel is a positive-definite function between two input vectors $\mathbf{x}$. The covariance is the similarity of outputs based upon the belief of if two inputs are located in proximity of each other so will their outputs, this functionality is embedded in the covariance function [63].

The kernel functions available exist in a variety of forms, each of which encode different properties which affect how the response function is modelled. Each kernel function contains a set of free parameters that can be tuned to specify the shape of covariance functions [182]. These free parameters are referred to as hyper-parameters as they do not specify parameters that affect the response function directly [182]. Rather, they affect the distribution over the response function parameters [182]. Some of the more commonly implemented kernels include [63]:

- Squared Exponential (SE) Kernel

$$k_{SE}\left(x_i, x_j\right) = \sigma_f^2 \exp\left(-\frac{1}{2L^2}\left(x_i - x_j\right)^2\right) \qquad \mathbf{x} = \left[x_1, x_2, \ldots, x_d\right] \tag{2.7}$$

  where, σ_f^2 is the output variance hyper-parameter. L is the length scale hyper-parameter. $\mathbf{x}$ is a vector of d inputs for an experiment.

  The SE kernel is a stationary kernel that is infinitely differentiable and has its response limited to the local neighbourhood at experiment x. The length scale determines the distance in the feature space from which a new experiment can be extrapolated from observed data points. The output variance determines the average distance the response function is from its mean, although simply this is just a scaling factor [63].

- Periodic Kernel

$$k_{\mathrm{Per}}\left(x_i, x_j\right) = \sigma_f^2 \exp\left(-\frac{2\sin^2\left(\pi||x_i - x_j||/p\right)}{L^2}\right) \qquad \mathbf{x} = \left[x_1, x_2, \ldots, x_d\right] \quad (2.8)$$

where, ||.|| is the absolute function. p is the period hyper-parameter.

The periodic kernel is also a stationary kernel similarly to the SE kernel but is more suited to model functions that have a repeating pattern such as seasonal effects or an oscillatory nature. The period factor determines the distance between each repetition of the function [63].

- Linear Kernel

$$k_{\mathrm{Lin}}\left(x_i, x_j\right) = \sigma_b^2 + \sigma_f^2 \left(x_i - c\right)^T \left(x_j - c\right) \qquad \mathbf{x} = \left[x_1, x_2, \ldots, x_d\right] \qquad (2.9)$$

where, c is the off-set hyper-parameter and σ_b^2 is the uncertainty off-set for the model.

The linear kernel on the other hand is a non-stationary kernel which varies its responses in a linear structure. The off-set c determines the input coordinate from which all functions in the posterior must pass through. Whilst the constant variance is an uncertain offset that determines the height of the response function that will be observed at zero [63].

The stationary property of a kernel means that the covariance is only dependent upon the difference between two sets of experiments, or $x_i - x_j$. As such, if each experiment were to be shifted by Δx, the observed covariance would remain unchanged. In contrast to this is in the case for non-stationary kernels such as linear kernels where if shifted by Δx, the observed covariance would change [63]. Figure 2.8 illustrates how different kernel functions represent different covariance structures which in turn when sampled from the GP prior results in varied response structures.

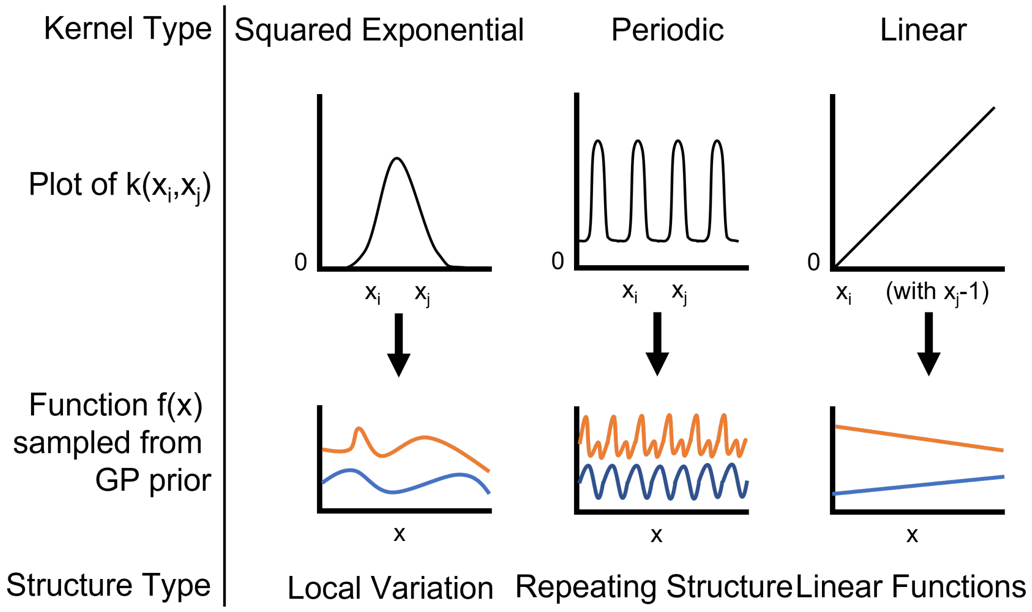


Figure 2.8: Underlying structure of basic kernels and example draws from the GP prior of each example kernel, adapted from [63].

Gaussian Process: Squared Exponential Kernel

Once an appropriate kernel is selected for the GP prior surrogate model its hyper-parameters need to be tuned [182]. One property of GP models that is beneficial for this problem is that there is an analytical expression for the marginal likelihood of the data [195]. Where the marginal likelihood of the data represents the likelihood of a particular set of model hyper-parameters estimating the true response at previously observed experiments [195]. The log marginal likelihood is simply given by Equation 2.10:

$$\log \mathrm{P}\,(\mathbf{y}|\mathrm{D},\theta) = -\frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathrm{K}_{\mathrm{y}}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathrm{K}_{\mathrm{y}}| - \frac{\mathrm{N}}{2}\log 2\pi \tag{2.10}$$

Where $\mathrm{K}_{\mathrm{y}} = \mathrm{K} + \sigma_{\mathrm{n}}^2 * \mathrm{I}$ is the covariance matrix for noisy target responses $\mathbf{y}$, with K, σ_{n}^2, and I representing the noise free covariance matrix, noise hyper-parameter and identity matrix respectively. n is the number of data points in the D or data set used for training the model. θ is a vector of the hyper-parameters used to tune the GP model.

The log marginal likelihood in Equation 2.10 can be broken down into three interpretable terms. The first term quantifies the surrogate model fit of response estimates to the true responses [195]. The second term is a quantification of the model complexity as smoother covariance functions will have a smaller determinant and subsequently lower model complexity [195]. The final term is a penalty term for the data set size, where as the size of the data set increases the log marginal likelihood decreases [195].

The optimisation of the hyper-parameters using the log marginal likelihood can be performed using an off the shelf optimiser as long as the kernel is differentiable, or for non-differentiable kernels using cross-validation [195]. In this thesis a k-fold cross validation approach will be predominately used with k = 10, except in the instances where the data set size is smaller than 10 experiments [182]. In these cases the k-fold cross-validation approach will be replaced with a leave-one out cross-validation approach [182].

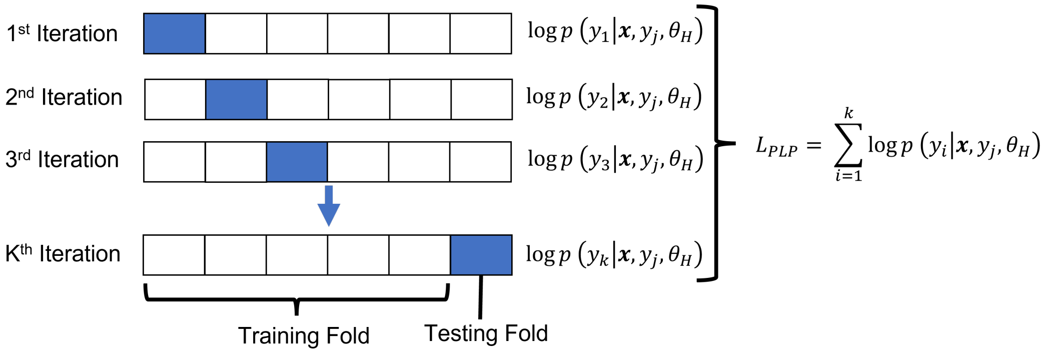

1st Iteration
2nd Iteration
3rd Iteration
Kth Iteration
$\log p\,(y_1|\boldsymbol{x}, y_j, \theta_H)$
$\log p\,(y_2|\boldsymbol{x}, y_j, \theta_H)$
$\log p\,(y_3|\boldsymbol{x}, y_j, \theta_H)$
$\log p\,(y_k|\boldsymbol{x}, y_j, \theta_H)$
$L_{PLP} = \sum_{i=1}^{k} \log p\,(y_i|\boldsymbol{x}, y_j, \theta_H)$
Training Fold
Testing Fold

Figure 2.9: A diagram to represent the data partitioning into training and testing data sets for both a K-Fold cross-validation approach and a leave-one-out cross-validation approach when the size of the data set n < k.

Figure 2.9 illustrates the procedure of both a k-fold cross-validation approach or a leave-one out cross-validation approach for one set of hyper-parameters being assessed θ_{h}. The primary difference between the two cross-validation approaches is the partitioning of the data set into a training and testing data set. For example in a 10 fold cross-validation on 20 experiments each fold contains 2 experiments, which when split into the two sub-sets: training (white in Figure 2.9) and testing (blue in Figure 2.9), would result in 18 training experiments and 2 testing experiments. On the other hand for leave-one-out cross-validation with 5 experiments there would only be 5 iterations with one experiment set aside to perform testing with the remainder used for training.

After partitioning the initial data set, the training data is used to train the surrogate model for a particular set of hyper-parameter settings and generated the covariance matrix.

$$K_{train} = k_{SE}\left(\mathbf{x}_{train}, \mathbf{x}_{train}, \theta_h\right) \tag{2.11}$$

Where $\mathbf{x}_{train}$ is the vector of training input data and θ_h is a vector set of hyper-parameters that are to be assessed, in a noise-free SE kernel $\theta_h = \left[\sigma_f^2, L\right]$. Once the surrogate model is trained the responses are inferred at the testing sets experiment locations to get their estimated mean (See Equation 2.21a) and uncertainty (See Equation 2.21b).

The estimated response (mean) and their uncertainty (standard deviation) are then used to calculate the predictive log probability which is referred to as a pseudo-log likelihood [182].

$$\log p\left(y_i|\mathbf{x}, y_j, \theta_i\right) = -\frac{1}{2}\log\sigma_i^2 - \frac{\left(y_i - \mu_i\right)^2}{2\sigma_i^2} - \frac{1}{2}\log 2\pi \tag{2.12}$$

Where i and j represent the testing and training sets respectively which are partitioned from the observed data set, the data contained within each i and j sets will alternate after each cross-validation fold is completed. $\log p\left(y_i|\mathbf{x}, y_j, \theta_h\right)$ is the pseudo-log likelihood of the testing data given the model trained on the training data using θ_h hyper-parameters. μ_{test} is the predicted mean of the test set and σ_{test} is the predicted standard deviation of the test set.

These steps are repeated for each fold k of the cross-validation as shown in Figure 2.9, whereby the training data and testing data are recombined and redistributed accordingly for each iteration after the predictive log probability has been calculated. Once the predictive log probability for all folds is determined the cross-validation predictive log probability is calculated [182] using Equation 2.13.

$$L_{PLP}\left(\mathbf{x}, \mathbf{y}, \theta_h\right) = \sum_{i=1}^{k} \log p\left(y_i|\mathbf{x}, y_j, \theta_h\right) \tag{2.13}$$

Where L_{PLP} is the predictive log probability which is the summed pseudo-log likelihood $\log p\left(y_i|\mathbf{x}, y_j, \theta_h\right)$ for each fold distribution for hyper-parameter set θ_h. In order to select an appropriate hyper-parameter setting, the cross-validation predictive log probability is maximised for a variety of hyper-parameter settings.

$$\theta_{best} = \arg\max_{\theta} L_{PLP}\left(\mathbf{x}, \mathbf{y}, \theta_h\right) \qquad h = 1, 2, \ldots, H \tag{2.14}$$

Where θ_{best} is the optimal hyper-parameter set according to maximising the $L_{PLP}\left(\mathbf{x}, \mathbf{y}, \theta_h\right)$ k-fold cross-validation pseudo log likelihood. H is the total number of hyper-parameter set variations being assessed. This is achieved through the use of a suitable global optimiser that can efficiently test a variety of hyper-parameter settings [182].

Gaussian Process: Inference

As shown in Section 2.2.1 in Equation 2.12 an analytical inference expression is required for the prediction of the posterior mean and variance functions of a GP. This allows for the prediction at any point **x** to determine their estimated mean and variance [182]. This can be used in model selection detailed in Section 2.2.1 but also will be utilised to cheaply optimise an acquisition function detailed in Section 2.2.2 to select the next optimal experiment in the BO framework [182].

When sampling from the GP prior we would use the observed data **x** and sample the responses **y** at these indices to produce data pairs that make up the initial data set, D [40].

$$D_{1:t} = \left[\mathbf{x}_{1:t}, \mathbf{y}_{1:t}\right]$$

Where **x** is a vector set of experiment inputs and **y** is a vector set of the true responses at the experiment locations. t is the total number of experiments run so far. The response y are drawn according to a multivariate normal distribution [40].

$$f_{1:t}(\mathbf{x}) \sim \mathcal{N}(m_{1:t}, \mathbf{K}_{1:t})$$

$$m_{1:t}(\mathbf{x}) = 0 \qquad \mathbf{K}_{1:t} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_t) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_t, \mathbf{x}_1) & \dots & k(\mathbf{x}_t, \mathbf{x}_t) \end{pmatrix} \tag{2.15}$$

Where $m_{1:t}(\mathbf{x})$ is the mean function of the Gaussian process of the observed data set, $\mathbf{K}_{1:t}$ is the covariance matrix function of the Gaussian process of the observed data set. By using the properties of a GP the next experiment $\mathbf{x}_{t+1}$ and response $f(\mathbf{x}_{t+1})$ are jointly Gaussian with the GP prior responses $f(\mathbf{x}_{1:t})$.

$$\begin{bmatrix} f_{t+1} \\ f_{1:t} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k\left(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}\right) & k^T \\ k & \mathbf{K}_{1:t} \end{bmatrix}\right) \tag{2.16}$$

Where,

$$k = \left[k\left(x_{t+1}, x_1\right), k\left(x_{t+1}, x_2\right), \ldots, k\left(x_{t+1}, x_t\right)\right]^T \tag{2.17}$$

Where k is the covariance between the sampling point and the already observed data set. In this form the predictive distribution can be formulated as a conditional distribution of $P\left(f_{t+1} \middle| f_{1:t}\right)$ by using the Sherman-Morrison-Woodbury inversion lemma formula [40].

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} A & C \\ C^T & B \end{bmatrix}\right) \tag{2.18}$$

$$x|y \sim \mathcal{N}(\mu_x + CB^{-1}(y - \mu_y), A - CB^{-1}C^T) \tag{2.19}$$

Using the Sherman-Morrison-Woodbury inversion lemma can easily arrive at an expression for the predictive posterior distribution. This can be used to derive analytical expression for the predictive mean and variance at the next experiments experiment location x_{t+1}.

$$P\left(f_{t+1}|D_{1:t}, \mathbf{x}_{1:t}\right) = \mathcal{N}\left(\mu_t\left(x_{t+1}\right), \sigma_n^2\left(x_{t+1}\right)\right) \tag{2.20}$$

Where,

$$\mu_t\left(x_{t+1}\right) = k^T K^{-1} f_{1:t} \tag{2.21a}$$

$$\sigma_t^2(x_{t+1}) = k\left(x_{t+1}, x_{t+1}\right) - k^T K^{-1} k \tag{2.21b}$$

Where $\mu_t\left(x_{t+1}\right)$ is the predictive mean and $\sigma_n^2\left(x_{t+1}\right)$ is the predictive variance at sampling point x_{t+1}. The posterior mean and variance represent the posterior GP models prediction and uncertainty in the response at any experiment evaluated [195].

2.2.2 Bayesian Optimisation: Acquisition Functions

As mentioned in summary of Section 2.2, BO was sub-divided into two key stages. In the previous Section 2.2.1 a surrogate model was chosen which encodes expert knowledge and sequentially observed data that will be selected through the optimisation of a cheap acquisition function [40]. The focus of this section is the secondary stage of BO where the focus lies on the optimisation of the cheaper-to-evaluate acquisition function, which trades off between exploring the feature space and exploiting potentially optimal experiments

[40].

The acquisition function or infill criterion is the function which represents the expected utility of potential experiments. Depending on the acquisition function that is utilised the expected utility focus may change between locating the global optimiser (exploitation), looking for new promising areas of the feature space (exploration), or a balance of both.

$$x_{t+1} = \arg\max_{x} u\left(x \middle| D_{1:t}\right) \tag{2.22}$$

where x_{t+1} is the next experiment to be chosen by optimisation an acquisition function u. There is a vast literature detailing the strategies and acquisition functions available to achieve the goals of BO, but they can be broadly categorised into 3 main types [195]:

- Improvement-Based acquisition functions.
- Optimistic-Based acquisition functions.
- Informative-Based acquisition functions.

Bayesian Optimisation: Improvement-Based Acquisition Functions

Improvement-based acquisition functions seek to choose experiments that improve over the current best solution located, also known as the incumbent solution $f\left(x^{+}\right)$ [40]. There have been a few different improvement-based acquisition functions published and reviewed in literature with one of the earliest being Probability of Improvement (PI) [131].

PI seeks to identify potential experiment settings that have a high probability to improve over the current target solution which is comprised of a pre-specified improvement over the incumbent solution [131].

$$\tau = f\left(x^{+}\right) - TL * f\left(x^{+}\right) \tag{2.23}$$

Where τ is the target improvement TL over the incumbent x^{+}, the posterior probability distribution when using a Gaussian prior is also Gaussian, this can be evaluated analytically using Equation 2.24:

$$\begin{aligned} u_{PI}(x) &= P\left(f(x) \geq f\left(x^{+}\right)\right) \\ u_{PI}(x) &= \Phi\left(\frac{\mu(x) - \tau}{\sigma(x)}\right) \end{aligned} \tag{2.24}$$

Where P(A ≥ B) is the probability that random variable A is greater than random variable B. PI is a simple and easy to implement acquisition function shown in Figure 2.10, which seeks to maximise the improvement over the current best solution. Although this can be a disadvantageous trait as the PI value will be highly dependent upon the tuning of τ. Whereby, if τ is set too small the PI will tend to over-exploit locally in the neighbourhood of the current best solution aggressively, thus leading to a lack of exploration [118]. Whereas, if τ set too large PI will tend to over-explore. Although in contrast if the target improvement is selected well then, the experiments selected using PI are encouraged to explore more globally [118].

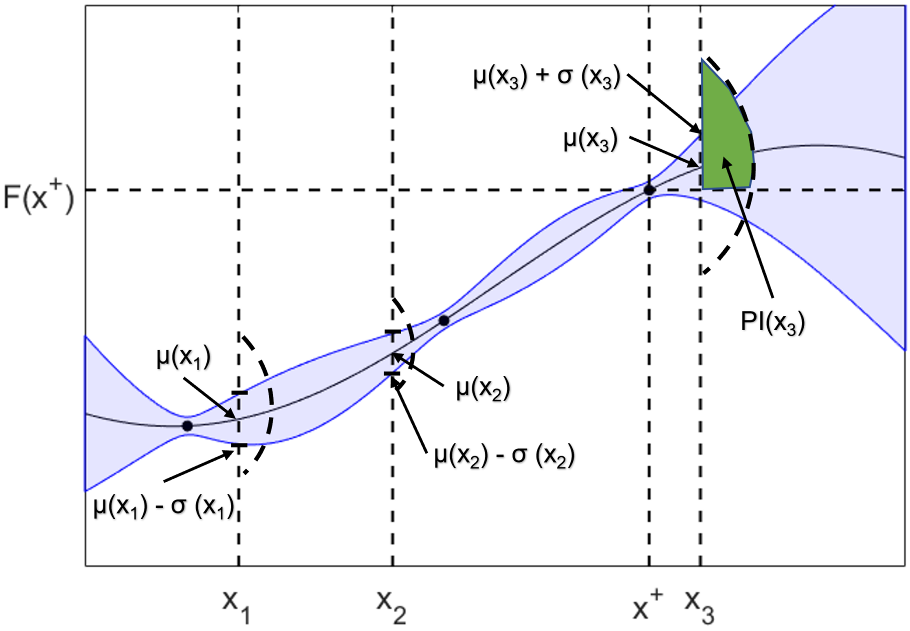


Figure 2.10: A single input/output function of a stretched sine wave (black line) in which three data points have been observed (black dots), whereby the shaded region denotes twice the standard deviation (uncertainty) at each input value. A GP model has been fitted to the observed data using a SE kernel. x_1, x_2 and x_3 represent three potential experiments to be assessed and x^+ being the currently best observed data point with an output value of $F(x^+)$. Adapted from [40].

In Figure 2.10 the experiments x_1, x_2 and x_3 are being evaluated to see if they can improve or maximise the 1-D function. The PI of x_1 and x_2 show no possibility of improvement as they do not improve over the current best observed data point x^+. On the other hand, x_3 does improve over the current best solution x^+ with the dark-green shaded region representing the measure of improvement.

Another improvement-based acquisition function was then introduced into literature to tackle these issues stemming from aggressive exploitation in PI, Expected Improvement (EI) [195]. EI sought to improve upon PI by introducing the concept of assessing the magnitude of improvement that a potential experiment can provide over an incumbent as well as its probability to improve [40]. Mockus et al. [161] achieved this by introducing the improvement function shown in Equation 2.25.

$$I(x) = \max\left[0, f_{t+1}(x) - f_{1:t}\left(x^+\right)\right] \tag{2.25}$$

Where I(x) is the improvement function of experiment x. $f_{t+1}(x)$ is the predicted response using the surrogate model for the next experiment x. $f_{1:t}\left(x^+\right)$ is the incumbent's response which is current best solution up until interval t.

Equation 2.25 shows is that the improvement of the next experiment x_{t+1} is only positive when the predicted response is greater than the incumbent, otherwise it is set to zero [40]. By maximising the expectation of the improvement function we can select the next optimal experiment.

$$x_{t+1} = \arg\max_{x} \mathbb{E}\left[\max\left[0, f_{t+1}(x) - f_{1:t}\left(x^+\right)\right] \,\middle|\, D_{1:t}\right] \tag{2.26}$$

Where $\mathbb{E}[A]$ is the expectation of event A. However just as with PI as the posterior probability distribution of the predicted response is Gaussian, an analytical form of the expectation of the improvement function can be derived [40].

$$u_{EI}(x) = \begin{cases} (\mu(x) - f(x^+))\Phi(Z) + \sigma(x)\varphi(Z) & \text{if} \quad \sigma(x) > 0 \\ 0 & \text{if} \quad \sigma(x) = 0 \end{cases} \tag{2.27}$$

Where,

$$Z = \begin{cases} \frac{(\mu(x) - f(x^+))}{\sigma(x)} & \text{if} \quad \sigma(x) > 0 \\ 0 & \text{if} \quad \sigma(x) = 0 \end{cases} \tag{2.28}$$

Where Φ is the cumulative distribution function, φ is the probability density function. $\mu(x)$ is the posterior mean at the experiment point, $f(x^+)$ is the observed value of the incumbent point, $\sigma(x)$ is the posterior standard deviation at the experiment point.

In Equation 2.27 the incumbent solution is equivalent to the τ in Equation 2.24 whereby it would be thought to aggressively search the surrounding feature space, however in EI this is not the case. This choice subsequently reduces the complexity of EI with requiring no hyper-parameters to tune.

Bayesian Optimisation: Optimistic-Based Acquisition Functions

Optimistic-based acquisition functions seek to choose experiments that are optimistically the best selection even in the face of uncertainty according to the surrogate model [195]. Thus, an optimistic selection policy can be thought of as selecting an optimal experiment by comparing their respective upper (or lower) confidence bound as shown in Figure 2.11 [195].

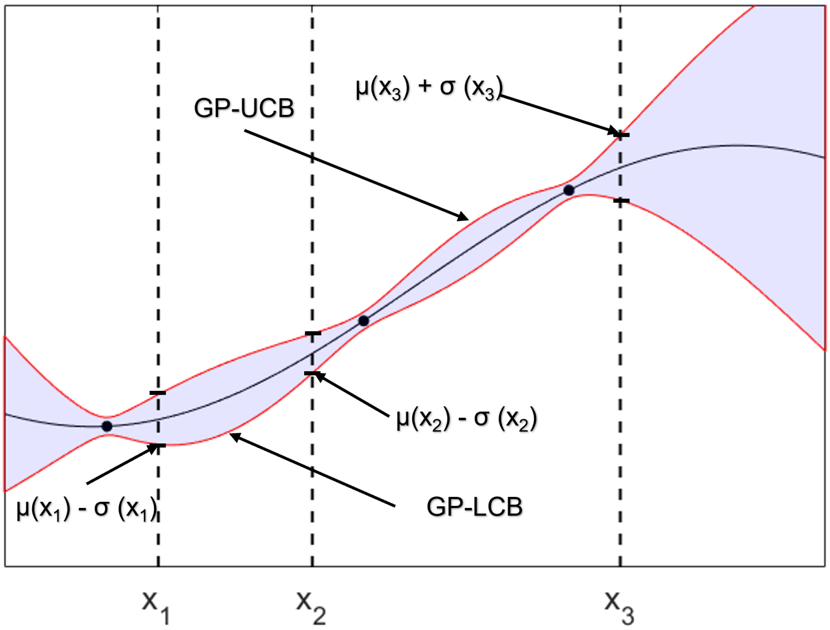


Figure 2.11: A single input/output function of a stretched sine wave (black line) in which three data points have been observed (black dots), whereby the shaded region denotes twice the standard deviation (uncertainty) at each input value. A GP model has been fitted to the observed data using a SE kernel. x_1, x_2 and x_3 represent three potential experiments to be assessed and the red border represents the upper and lower bounds. Adapted from [40].

In Figure 2.11 the experiments x_1, x_2 and x_3 are being evaluated to which experiment is the next best selection. Depending on if Equation 2.29 or Equation 2.30 is used the selected experiment would change accordingly. Since, as can be seen if Equation 2.29 were maximised x_3 would be selected due, whereas if Equation 2.30 were minimised x_1 would be selected.

This selection policy was first introduced as a means of trading-off between exploration and exploitation in the multi-armed bandit literature with provable cumulative regret bounds [133]. Srinivas et al. [208] adapted the upper confidence bound criterion from multi-armed bandit literature into a suitable criterion for BO literature with provable regret bounds in Equation 2.29:

$$u_{GP-UCB}(x) = \mu(x) + \beta_t \sigma(x) \tag{2.29}$$

where, $u_{GP-UCB}(x)$ is the Gaussian Process Upper Confidence Bound (GP-UCB) acquisition function, β_t a domain-specific time-varying trade-off parameter, $\mu(x)$ is the posterior predictive mean of experiment x, and $\sigma(x)$ is the posterior predictive standard deviation of experiment x. Equation 2.29 can also be modified to its lower bound variant by subtracting the second term instead of adding as shown in Equation 2.30:

$$u_{GP-LCB}(x) = \mu(x) - \beta_t \sigma(x) \tag{2.30}$$

where, $u_{GP-LCB}(x)$ is the Gaussian Process Lower Confidence Bound (GP-LCB) acquisition function. The selection and tuning of the β_t is the determining factor in the theoretical convergence of their regret bounds [208]. Whereby the β_t parameter adaptively determines the trade-off between exploration and exploitation in optimistic BO policies [40].

Bayesian Optimisation: Information-Based Acquisition Functions

Information-based acquisition functions are the most recent class of selection criterion introduced in BO literature which uses information theory to approximate the information gain of potential experiments towards the unknown optimum x^* [100] [101] [230]. The basis for this approach was built upon the foundation of expected change in knowledge from experiments, where exploring smaller high-density information regions (local optimums) may lead to smaller increases in global knowledge. Whereas exploring a broader region of low-density information could lead to a greater expected increase in knowledge which was not captured by improvement or optimistic techniques [100].

This expected change in knowledge through information gain is quantified via information theories entropy [197], which is the average level of information for a random variable for all possible outcomes. However, entropy is used to describe discrete random variables and so differential entropy will be used henceforth for continuous random variables. In information-based acquisition functions sought to gain the most knowledge of the global optimiser x^* which can be quantified using its distribution $P\left(x^* \middle| D\right)$ and its corresponding entropy:

$$H\left(P\left(x^* \middle| D\right)\right) = -\int_X f\left(P\left(x^* \middle| D\right)\right) \log P\left(x^* \middle| D\right) dx \tag{2.31}$$

Where, $P\left(x^* \middle| D\right)$ is the posterior distribution of the global optimiser, $H(.)$ is the differential entropy of its argument. Entropy Search (ES) was the first implementation of an information-based approach that sought to maximise the information gain with each experiment, this is equivalent to maximising the reduction in entropy of $P\left(x^* \middle| D\right)$ [100] as shown in Equation 2.32:

$$u_{ES}(x) = H\left(P\left(x^* \middle| D\right)\right) - \mathbb{E}_{P(y|D,x)} H\left[P\left(x^* \middle| D \cup \{x, y\}\right)\right] \tag{2.32}$$

Where, u_{ES} is the Entropy Search (ES) Acquisition Function, $\mathbb{E}_{P(.)}H(.)$ is the expectation of the differential entropy of its argument $H(.)$ over the posterior distribution $P(.)$. $H\left[P\left(x^* \middle| D \cup \{x, y\}\right)\right]$ is the differential entropy of the posterior distribution of the global optimum with respect to a particular experiment $\{x, y\}$ and $P(y|D, x)$ is the posterior predictive distribution of y given x.

Whilst an information-theoretic approach provides a more probabilistic approach in comparison to the heuristic methods of improvement and optimistic approaches, they are more computationally expensive [195]. The computational expense in ES is due to two difficulties:

- $P\left(x^* \middle| D \cup \{x, y\}\right)$ must be computed for various combinations of x and y during the optimisation.
- The entropies involved in Equation 2.32 are analytically intractable and so must be approximated using Expectation Propagation (EP) on a discretised grid.

In order to alleviate some of the issues raised due to the computational complexity of the ES framework Henrández-Lobato et al. [101] developed Predictive Entropy Search (PES). PES is able to reduce some of the computational complexity by utilising the mutual information between x^* and y given a set of experiments (N) have been performed [101]. Since the mutual information is a symmetric function Equation 2.32 can be re-written in the form Equation 2.33:

$$u_{PES}(x) = H\left(P(y|D, x)\right) - \mathbb{E}_{P(x^*|D)} H\left[P\left(y \middle| D, x, x^*\right)\right] \tag{2.33}$$

where, u_{PES} is the Predictive Entropy Search (PES) Acquisition Function, $H\left(P\left(y \mid D, x\right)\right)$ is the differential entropy of the posterior predictive distribution, $P\left(x^{*} \mid D\right)$ is the posterior distribution of the global optimum, and $H\left[P\left(y \mid D, x, x^{*}\right)\right]$ is the differential entropy of the posterior predictive distribution of y given the observed data D and global optimum x^{*} are known.

The benefit of PES is that the first term is now an entropy of a predictive distribution, for which there is an analytical form. Although the second term is still a distribution dependent upon the global optimiser x^{*} its entropy still requires approximation using EP [101].

The final information-based approach in literature is the Max Value Entropy Search (MES) [230]. MES is a similar approach to PES in that it utilises the symmetric property of mutual information to reformulate Equation 2.32. Instead MES focuses on the mutual information between y^{*} and the next experiment x. Therefore, Equation 2.33 can then be reformulated into Equation 2.34.

$$u_{MES}(x) = H\left(P\left(y \mid D, x\right)\right) - \mathbb{E}_{P(y^{*} \mid D)} H\left[P\left(y \mid D, x, y^{*}\right)\right] \tag{2.34}$$

where, u_{MES} is the Max Value Entropy Search (MES) Acquisition Function, $H\left(P\left(y \mid D, x\right)\right)$ is the differential entropy of the posterior predictive distribution, $P\left(y^{*} \mid D\right)$ is the posterior distribution of the global optimum output, and $H\left[P\left(y \mid D, x, y^{*}\right)\right]$ is the differential entropy of the posterior predictive distribution of y given the observed data D and global optimum output y^{*} are known.

MES shares advantages with those previously mentioned in regard to PES whereby the first term is an entropy of a predictive distribution and is thus analytically tractable. However, the key advantage arises in regard to the second entropy term where it was previously analytically intractable due to the dependency on the global optimiser x^{*} [101]. Instead in MES the second entropy is a truncated Gaussian distribution of y^{*}, which is analytically tractable given y^{*} is known [230].

Only the expectation over the distribution $P\left(y^{*} \mid D\right)$ requires approximation which is achieved using Monte Carlo estimation using a set of sampled y^{*} [230]. Wang and Jegelka [230] suggested two methods to sample the y^{*}:

- Sampling from a Gumbel distribution.
- Sampling from a posterior Gaussian distribution and optimising the samples.

By utilising the analytical forms of the differential entropies and the approximation of the expectation using Monte-Carlo estimation an analytical form for MES is shown in Equation 2.35:

$$u_{\mathrm{MES}}(\mathbf{x}) = \frac{1}{\omega} \sum_{y^* \in Y^*} \left[\frac{\gamma_{y^*}(\mathbf{x})\, \varphi\left(\gamma_{y^*}\right)}{2 * \Phi\left(\gamma_{y^*}\right)} - \log \Phi\left(\gamma_{y^*}\right) \right] \tag{2.35}$$

where,

$$\gamma_{y^*} = \frac{y^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})} \tag{2.36}$$

ω is the total number of Monte Carlo estimations of the global optimum output y^*, φ is the probability density function, and Φ is the cumulative density function.

2.2.3 Bayesian Optimisation: Global Optimiser

The final decision in BO is the choice of a suitable optimiser for cheap acquisition functions which is central to the operation of the BO framework for selecting new experiments [195]. Hence the choice of an optimiser is dependent upon the properties of the acquisition function for which traditional optimisers may not be suitable.

As the optimisation landscape for acquisition functions is multi-modal, this leads to increased optimisation complexity with many local optima as well as problems which require constraints and/or multiple response criteria [195]. As a result, the literature has implemented a variety of optimisation approaches including: discretization [203], adaptive grids [16], divided rectangles [116], gradient-based optimisers [145], and other global optimisation heuristics [35]. Although gradient-based optimisers require multiple restarts to prevent being trapped in local optima and can only be implemented when gradients can be cheaply evaluated [195].

Therefore, a global optimisation technique with capabilities in handling multi-modal functions, large input parameter spaces, constraints, and multiple response criteria all of which increase the problem complexity would be suitable. A global optimiser well-suited to multi-modal problems are population-based search meta-heuristics in particular Genetic Algorithm (GA) [233]. A GA is a population-based metaheuristics which imitates Darwin's *Theory of Natural Selection* to evolve or iterate a population of *Individuals* " (experiments) towards a global optimum [35].

The benefit of implementing a GA is the evolution of a population of individuals which provides the ability to perform parallel searches in the factor space and locate various potentially optimal solutions [33]. This in contrast to gradient-based optimisers in which if the initial points are chosen poorly it will result in unfavourable local optima, thus requiring multiple restarts which is not the case for GA's [149]. GA's are also incredibly flexible for the incorporation of constraints into the feature space, as well as handling multiple response criteria simultaneously [149]. On the other hand, a GA is inhibited by the requirement to tune a large number of evolutionary parameters that are used in both primary as well as additional evolutionary functions which are used to obtain good GA performance [149].

Therefore, a population-based metaheuristic in this thesis Genetic Algorithm (GA) was chosen as the optimiser of choice for use within the Bayesian optimisation framework. Section 3.3 will detail the choices of primary and secondary evolutionary functions incorporated into the GA optimiser as well as the tuning or choice of their subsequent parameters.

2.2.4 Bayesian Optimisation: Related Literature

Whilst BO is most commonly applied using a GP surrogate model in combination with any of the acquisition functions mentioned in Section 2.2.2, these are not the only surrogate models utilised. Other alternative surrogate models are implemented throughout BO literature including student-t process priors [193], random forests [108], deep neural networks [204], Bayesian neural networks [206], Mondrian trees [231], and Kriging [128].

Of these alternative surrogate models, a popular alternative in BO literature is Kriging which was developed in Geo-statistics at the University of Witwatersrand in 1951 [128]. Although the method was further popularised by the works of Matheron and colleagues in 1970s [154]. In Kriging the core assumption lies in the fact that errors are not independent and are in fact spatially correlated [40]. For example, if an experiment has a large error, then an experiment close in the feature space is also likely to have a large error [40].

Kriging is a combination of two models: a linear regression model and a stochastic (GP) model of the errors of the linear model [40]. Therefore, a kriging model interpolates a random field via a linear predictor [40]. Thus, a Kriging model will fit a model to locally restricted regions of the feature space in contrast to a GP that utilises all data in the feature space to produce a global model [40].

Kriging was implemented into experimental design as part of the Design and Analysis of Compute Experiments (DACE) framework [188]. The DACE DACE framework was then integrated with the sequential EI criterion shown in Equation 2.27 by Jones et al. [119], to produce the Efficient Global Optimisation (EGO) algorithm which has become a significant body of work with various extensions.

2.2.5 Constrained Bayesian Optimisation

Expensive Constrained Optimisation Problems (ECOP) exist both within the literature and manufacturing industries whereby either the process or objective functions are expensive to evaluate and subject to constraints [229], represented in Equation 2.37:

$$\min \quad f(x) \quad x = \left(x_1, x_2, \ldots, x_d\right) \tag{2.37}$$

$$\text{s.t.} \quad g_i(x) \leq 0 \quad i = 1, 2, \ldots, I \tag{2.38}$$

Where f(x) is the response variable, x is the input variables from 1 to d, $g_i(x)$ is the i^{th} inequality and equality constraints. In this classification a solution is determined to be feasible if it is contained within the feasible space defined by the constraints be held valid. The constrained Bayesian optimisation literature can be roughly categorised into two sections [229]:

1. Constrained Bayesian Optimisation (CBO).
   - Probability of Feasibility
   - Expected Volume Reduction
   - Multi-Step Look-ahead
2. Surrogate-Assisted Constraint-Handling methods.

CBO: Probability of Feasibility

CBO literature can be further broken down into three main sub-categories, of which the first is probability of feasibility based methods. Probability of feasibility based methods are combinations of BO acquisition functions with feasibility indicators to provide a combined acquisition function for the constrained BO problems. The most commonly modified BO acquisition for use with feasibility indicators is the EI such as: weighted EI (wEI) [80] [76] and constrained EI (cEI) [93] [124] which is typically defined as in Equation 2.39

$$\mathrm{cEI}(x) = \mathrm{EI}(x) \prod_{i=1}^{I} \Pr\left(g_i(x) \leq 0\right) \tag{2.39}$$

Although EI is the more frequently adapted acquisition function for use with feasibility indicators there have been implementations with alternative acquisitions such the Knowledge gradient [44] [219].

CBO: Expected Volume Reduction

The expected volume reduction branch of CBO is a class of acquisition functions based upon the principles of Stepwise Uncertainty Reduction (SUR) [46]. The SUR strategy seeks to construct a sequence of evaluation points so as to quickly reduce the residual uncertainty about a quantity of interest given the information provided by the evaluation points [46]. Therefore, many acquisition functions can be derived to accommodate constraints by reducing a specific type of uncertainty measure [229].

In Picheny [175] an uncertainty measure was based upon PI combined with a feasibility indicator (see Section 2.2.5) in order to accommodate for the constraints. Following similar principles using the EI acquisition function the Integrated expected conditional improvement (IECI) was defined [28]. Alternatively other uncertainty measures have been implemented using entropy in information-theory based acquisition functions using both PES [103] [78] and MES [174].

CBO: Multi-Step Look-Ahead

Multi-step Look-Ahead is one of the smaller sub-branches of CBO which implements constraint handling on non-myopic acquisition functions. Whereby, typically in BO acquisition functions are myopic in which they seek to select an experiment one step ahead into the future, whilst ignoring the potential impact this selection may have on future selections [229]. In contrast a non-myopic acquisition function aims to select samples or experiments by evaluating and optimising the long-term reward multiple steps into the future.

In Lam and Willcox [134] formulated the BO look-ahead as a dynamic problem using a variety of acquisition functions including: PI, EI, and UCB. Constraints were factored into [134] by redefining the stage-reward to the reduction of the objective function that satisfied the constraints, however the large computational burden is also large. Recently in order to rectify the computational burden produced from the previous non-myopic acquisition function in [134], Zhang et al. [251] introduced a constrained two step acquisition function known as 2-OPT-C which uses likelihood ratios to perform the optimisation.

Surrogate-assisted constraint-handling

In order to circumvent the computational and theoretical complexity involved in the derivation and development encountered in ECOP's, instead of developing new acquisition functions the BO can simply be incorporated into existing constraint-handling methods.

One branch of constraint-handling methodology of great interest is the evolutionary algorithm such as genetic algorithms or MOEA's [229]. For example in [249], two acquisition functions (EI and probability of feasibility) are used as output objectives for the MOEA to optimise simultaneously.

2.3 Manufacturing Extensions: Bayesian Optimisation

As mentioned in Section 2.1.9, manufacturing industries in particular AM have various problem characteristics that require a suitable DoE approach. In Section 2.2, BO was introduced as a DoE approach for tackling problems with expensive experimentation, that produces representative surrogate models and locates global optimums in as few experiments as possible. Also, by using a GA optimiser for the acquisition function optimisation the inclusion of factor or process constraints are easily implemented.

Equally important AM DoE problems are DoE methods capable of selecting experiments to be performed in batches as well as selecting experiments in the presence of multiple conflicting response variables. In this section an introduction into the DoE issues related to selecting experiments in batch and optimisation in the presence of multiple conflicting response variable will be introduced.

2.3.1 Batch Design of Experiments

Initially BO performed a sequential design in which experiments are chosen sequentially to perform to generate new information. Although, in recent years the manufacturing industries such as AM have seen large investment due to their ability to provide customized high-quality products with increased production speed [60]. These capabilities are leveraged through the use of

AM production processes which are often smaller in scale in comparison to conventional manufacturing processes [60].

As AM uses intensified manufacturing processes, multiple processes can be run in parallel as well as using *Nesting* in which multiple parts can be built in parallel in the same processing step [45]. Therefore, when performing DoE on AM problems selecting multiple experiments at each iteration instead sequential selection will result in a reduction of the total experimentation time required [91]. Despite the potential benefits performing multiple experiments in parallel can potentially bring there are also subsequent disadvantages that need to be addressed. The primary concern of a Batch Bayesian Experimental Design Optimisation (BB-DoE) approach is the type of selection method: Greedy and Non-Greedy selection.

Batch Optimisation: Greedy vs Non-Greedy Selection

Initially in BO the goal of each iteration is to select a single experiment which is used to both explore the feature space as well as search the feature space for a globally optimal parameter settings for the DoE problem. However, in batch selection the optimisation goal in Equation 2.3 is modified in Equation 2.40 for the selection of multiple experiments in the same iteration step:

$$\mathbf{B}_{t+Q} = \arg\max_{B \in R} u\left(\mathbf{B}|D_{1:t\text{-}1}\right) \qquad \mathbf{B} = \left[x_{t+1}, x_{t+1}, \ldots, x_{t+Q}\right] \tag{2.40}$$

where, $\mathbf{B}_{t+Q}$ is the next batch set of Q experiments to be selected, R is the design space of interest, and **B** is a batch set that is evaluated.

A greedy batch selection scheme selects a locally optimal experiment to be added to the batch set one at a time, which is repeated until the batch set is filled. Although as each experiment added to the batch set is not yet evaluated, an alternative heuristic should be used to provide an approximation of their outputs prior to evaluation [53]. The approximated outputs are referred to as Hallucinated Outputs as defined in Definition 2.3.1:

Definition 2.3.1 (Hallucinated Outputs).
Approximation of the output for pending experiments, using a variety of heuristics such as: Kriging Believer [85], Constant Liar [85] and Predictive mean Desautels et al. [59]. These hallucinated outputs act as placeholders in the data set $D_{1:t+Q}$ until the batch sets are filled and the experiments evaluated.

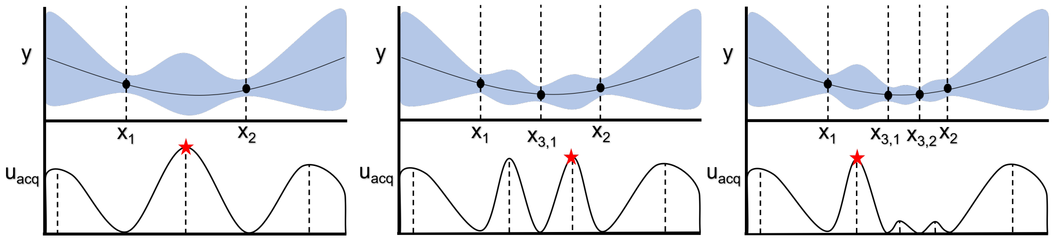
$$D_{1:t+Q} = \begin{bmatrix} x_1, & y_1 \\ x_2, & y_2 \\ & \vdots \\ x_{t+1}, & \bar{y}_{t+1} \\ x_{t+2}, & \bar{y}_{t+2} \\ & \vdots \\ x_{t+Q}, & \bar{y}_{t+Q} \end{bmatrix} \tag{2.41}$$

where, $\bar{y}$ are pseudo or hallucinated response values. The purpose of the approximated responses or hallucinated outputs is that each experiment chosen will influence the selection of the next experiment. Therefore, the hallucinated output acts as an approximate response value to be used to update the data set temporarily. This informs the sequential optimisation of the next batch experiment and thus the method used to update pseudo-response impacts the performance of greedy BB-DoE approaches. This series of sequential optimisation and pseudo-response updating is iterated until the batch set has been filled as shown in Figure 2.12a. Once filled the batch of experiments are performed to retrieve the actual response values, then the data set and subsequently the surrogate model are updated.

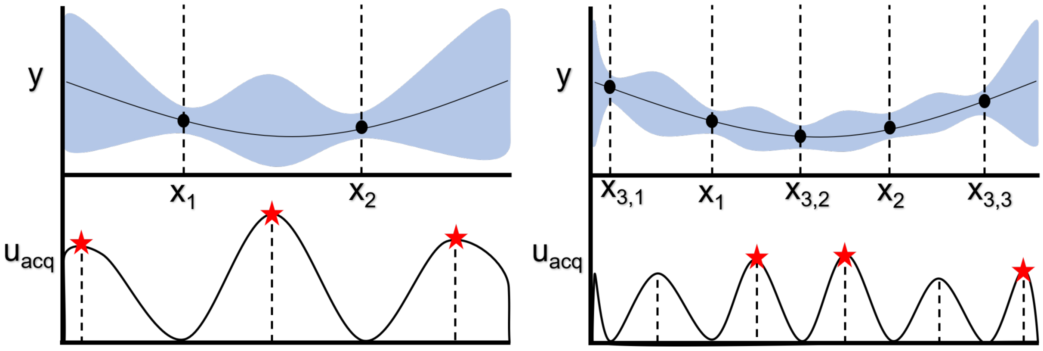
The advantage of greedy BB-DoE scheme is the capability of locating locally optimal parameter settings within a small number of steps which in some cases can lead to a good approximation of a globally optimal solution [192]. This is an attractive property when tackling complex feature spaces with a large number of factors as these can be complex to optimise. Secondly, is their fast implementation as the scheme is strictly a sequential scheme with adaptations to allow for multiple experiments to be selected without observation of their true responses [195]. Whereby, Greedy BB-DoE literature approaches are primarily varied based upon the pseudo-response estimation method implemented [86] [84] [59].

On the other hand, the disadvantage of a greedy BB-DoE scheme is that they frequently do not locate or approximate a globally optimal parameter setting, but rather a lower performing local approximation [192]. Thus, greedy BB-DoE schemes can lead to the globally worst solution as was shown in Gutin et al. [95] for the travelling salesman problem.

This is in contrast to non-greedy BB-DoE selection schemes which seek to select each experiment within the batch set simultaneously as shown in



(a) Greedy Batch Selection of three points over three iteration in order from left to right of $x_{3,1}$, $x_{3,2}$, and $x_{3,3}$ respectively. The figures along the top represent the GP model with the figures underneath their corresponding acquisition function value plots.



(b) Non-Greedy Batch Selection of three points over 1 iteration selecting all $x_{3,1}$, $x_{3,2}$, and $x_{3,3}$ simultaneously. The figures along the top represent the GP model with the figures underneath their corresponding acquisition function value plots.

Figure 2.12: Comparison between the selection schemes of Greedy BB-DOE shown in Figure 2.12a and Non-Greedy BB-DOE shown in Figure 2.12b for a batch set of B = 3. The problem is a minimisation problem on a 1-D function (black line) with two previously observed data points (black dots) at x_1 and x_2 and the shaded region denotes twice the standard deviation (uncertainty) at each input value.

Figure 2.12b. Thus, the main advantage of a non-greedy BB-DoE selection scheme is that the joint acquisition cost for a batch set can be determined whilst accounting for experiment interactions with each other to locate an optimal combination of experiments.

Consequently, this also links to a new combinatorial optimisation component for non-greedy BB-DoE approaches which becomes their primary disadvantage. Whereby, in order to select the next batch set, a large variety of combinations of experiments for which each experiment will also contain their own combinations of factors (parameter settings) must have their joint acquisition cost determined. Therefore, non-greedy BB-DoE strategies require methods to minimise increased optimisation complexity in order to leverage the improved performance achieved through accounting for inter-batch set interactions and their effects on the batch set acquisition cost.

2.3.2 Multi-Objective Optimisation

Similarly, in Section 2.3.1 wherein B-DoE was initially focused upon sequential selection instead of batch selection, in the same way traditional methods were focused upon optimising a single response variable or objective. In particular for manufacturing and AM problems this is often not the case but rather a manufacturing process can have multiple response variable to optimise in conflict with one another. Therefore, an alternative branch of optimisation is used to seek solutions to Multi-Objective Problems (MOP) that contain multiple conflicting response variables, known as Multi-Objective Optimisation (MOO) represented in Equation 2.42:

$$\min \quad f_m(x) \quad m = 1, 2, \ldots, M \tag{2.42}$$

$$\text{s.t.} \quad g_i(x) \geq 0 \quad i = 1, 2, \ldots, I \tag{2.43}$$

$$h_j(x) = 0 \quad j = 1, 2, \ldots, J \tag{2.44}$$

where, $f_m(x)$ is the m^{th} response variable to be optimised, $g_i(x)$ are the inequality constraints, and $h_j(x)$ are the equality constraints.

When optimising MOP which have multiple response variables it often the case that in order to improve the performance in regard to one response, this will lead to degradation in the other responses [237]. Therefore, when optimising MOP's the goal isn't to seek a globally optimal solution but rather a set of potentially optimal solutions that trade-off between all response variables known as the Pareto Optimal Set (POS) [54]. Each experiment in the POS is optimal in the sense that no improvement can be gained in one response variable without the degradation of another [73] [71] [72].

To determine whether a potential experiment belongs to the POS a set of definitions are required to differentiate which experiments are: Inferior, superior and non-inferior with respect to other experiments in the feature space [73].

Definition 2.3.2 (Inferiority).
An experiment $\mathbf{u} = \left(u_1, \ldots, u_n\right)$ is said to be inferior to experiment $\mathbf{v} = \left(v_1, \ldots, v_n\right)$, if experiment $\mathbf{v}$ is partially less than experiment $\mathbf{u}$ ($\mathbf{v} < \mathbf{u}$) at

all inputs i:

$$v_i \leq u_i \qquad \forall i = 1, \ldots, n$$

Definition 2.3.3 (Superiority).
An experiment $\mathbf{u} = \left(u_1, \ldots, u_n\right)$ is said to be superior to experiment $\mathbf{v} = \left(v_1, \ldots, v_n\right)$, if $\mathbf{v}$ is inferior (See Definition 2.3.2) to $\mathbf{u}$.

Definition 2.3.4 (Non-Inferiority).
A set of experiments (**u** and **v**) are said to be non-inferior to each other if they are neither Inferior (See Definition 2.3.2) or Superior (See Definition 2.3.3) to each other.

For an experiment to be a member of the POS they must satisfy Definition 2.3.3 with respect to experiments outside the POS and Definition 2.3.4 to members within the POS. The POS is collectively known as the Pareto Front (PF) which represents the current best solutions in the response space as shown in Figure 2.13 in a two-response variable example.

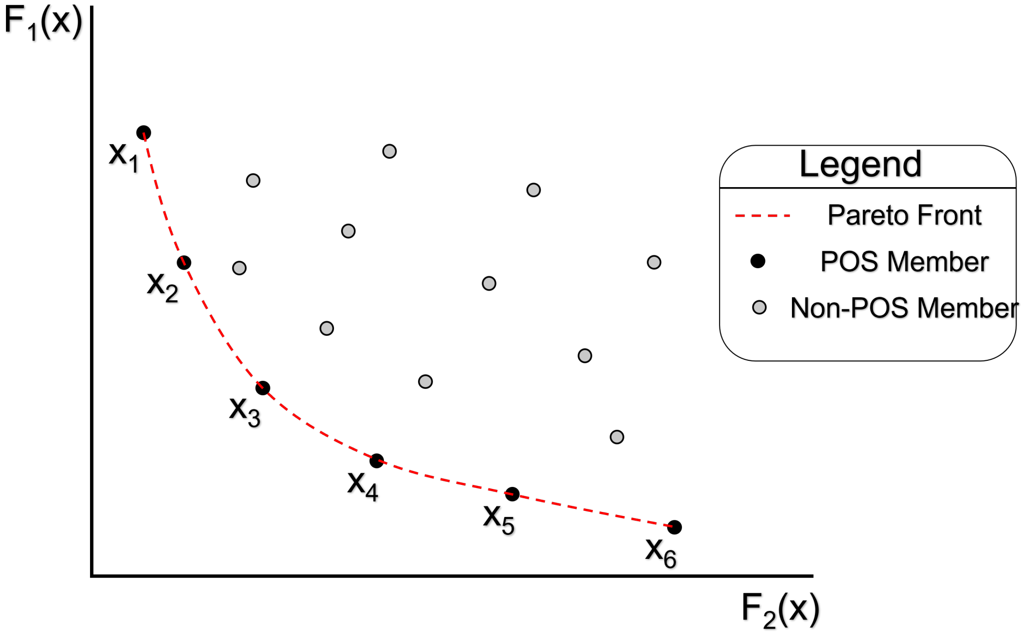


Figure 2.13: Representation of a Pareto Front for a 2-Dimensional (Response) Minimisation Multi-Objective Problem in the response space.

MOO strategies can be classified into 3 main types: a priori methods, a posteriori methods and interactive/preference methods [54]. Firstly, a priori methods utilise high level information (provided by an expert or the decision maker) in order to convert a MOP into a Single-Objective Problems (SOP). This is accomplished through the use of a weight vector scalarisation approach such as Weighted Sum approach shown in Equation 2.45:

$$\sum_{i=1}^{M} w_i * f_i(x), \quad i = 1, 2, \ldots, M \quad \text{and} \quad \sum_{i=1}^{M} w_i = 1 \tag{2.45}$$

where, w_i is the i^{th} weight and M is the total number of output objectives. Through conversion of the MOP into a SOP, a simpler Single-Objective Optimisation (SOO) approach can be implemented to retrieve a single optimal solution. However, this methodology is highly dependent on the choice of the weight vector set which can be disadvantageous as this does not guarantee the global optimal solution is located [54].

Secondly, a posteriori method in contrast to a priori methods seek to generate a full POS for the MOP using a MOO. Once a non-dominated set of solutions has been generated and converged a decision maker or experimented can utilise high-level information to select the optimal solution. Whilst this method generates a POS to reduce the risk of missing a potentially globally optimal solution, its primary disadvantage is the increase in computational complexity [54].

Finally, a hybrid of both a priori and posteriori methods are available also known as interactive or preference methods. An interactive approach initially follows a posteriori method using a MOO on the MOP to generate a POS. Upon the generation of a POS the optimisation process is paused for the decision maker to utilise high-level information to focus on particular regions of interest in order to guide the search. These steps are repeated systematically until a generated POS or optimal solution is located that matches the decision makers preference [54].

Multi-Objective Optimisation: Optimisation Methodologies

Of the 3 categories of MOO previously mentioned the most suitable choice for inclusion with a DoE approach is a non-interactive approach that provides a population of solutions to be selected from in each iteration. Therefore, a posteriori approach is the most suitable choice but within literature there are 3 main design goals that should be satisfied when selecting a MOO to ensure the production of an optimal and representative POS [217]:

Design Goal 1: Convergence

Locate a POS as close to the true PF as possible.

Design Goal 2: Diversity

Locate a uniformly distributed and diverse POS along the PF as possible.

Design Goal 3: Coverage

Locate a POS that covers the majority of the PF.

Within the literature in previous decades the most popular choice of a posteriori MOO used is Evolutionary Algorithm (EA) as shown in 2002 [117], whereby 70 % of MOO used EA's. Also, the primary optimiser for the B-DoE approach will be a GA (a type of EA)) this will provide ease of integration of a MOO approach when extending the B-DoE DOE methodology for tackling MOP's. For further introductions into the core GA methodology implemented see Section 3.3 for further details.

Multi-Objective Evolutionary Algorithm (MOEA) can be further classified into 3 main optimisation approaches based upon their approach to optimising the MOP, which will be briefly defined before a decision on the type of approach to be utilised is made:

1. Pareto-Dominance Multi-Objective Evolutionary Algorithm (PD-MOEA)
2. Performance Indicator Multi-Objective Evolutionary Algorithm (PI-MOEA)
3. Decomposition Based Multi-Objective Evolutionary Algorithm (D-MOEA)

Pareto-Dominance Multi-Objective Evolutionary Algorithms

A PD-MOEA is an MOEA which sorts solutions within the response space in relation to how they dominate other solutions into ranks. This is expressed as non-dominated solutions that lie within the POS as having the greatest potential fitness and thus the best ranking. Whereby, each rank lower will have a correspondingly lower fitness assigned and as such have a lower likelihood to be selected for evolution in the GA.

This approach promotes convergence (**Design Goal 1**) by increasing the potential of non-dominated solution to *mate* in the GA. However, this mechanism does not encourage or maintain the diversity of solutions within the population and so these properties are maintained by alternative evolutionary functions [54]. A variety of diversity management methods have been developed for PD-MOEA's including crowding distance, fitness sharing, elitism and non-dominated sorting [244]. The combination of PD-MOEA methodology with select diversity management techniques mentioned previously have led to the development of frequently implemented state-of-the-art methods: MOGA [73] [71] [72], SPEA/SPEA-II [253] [255], and NSGA/NSGA-II [207] [55].

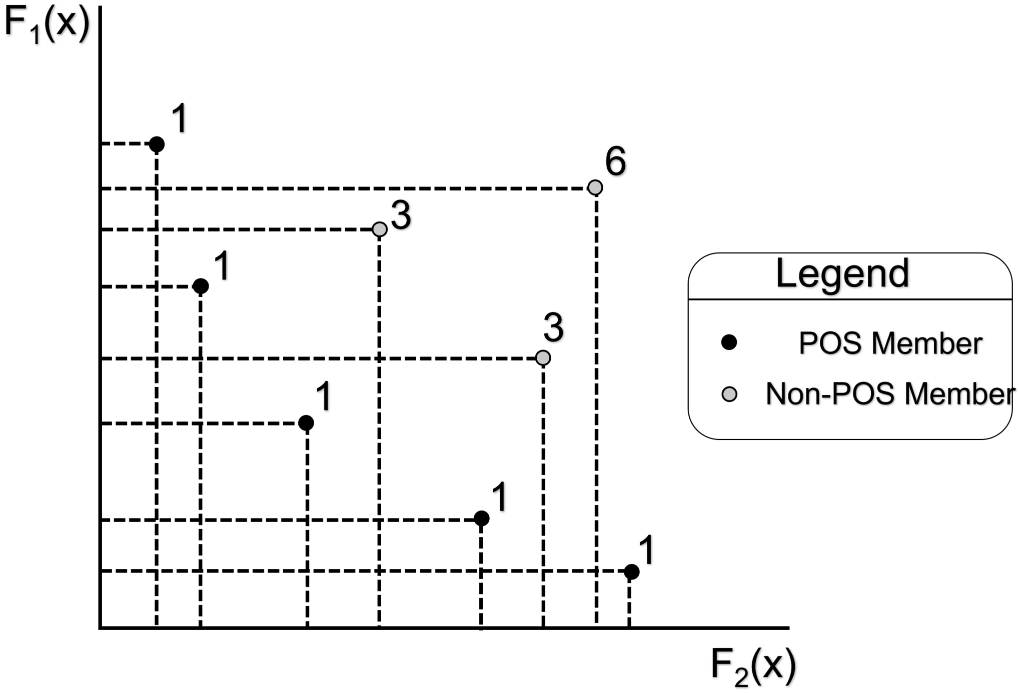


Figure 2.14: Ranking a Pareto front in 2-Dimensional objective (response) space using pareto dominance from Fonseca and Fleming [73] on a minimisation problem. As stated in Definition 2.3.2 a solution is dominated if for all outputs of interest are partially lower than another. This can be seen visually on a 2-D problem with rank 1 solutions as their dashed line zones do not contain any other solutions and are thus non-dominated solutions. Whereas as can be seen with the rank 3 solutions within their dashed box zones there are 2 other solutions resulting in a rank of 3 and are thus dominated solutions.

Fonseca and Fleming [73] initially developed MOGA which comprised of a Pareto ranking strategy, where a chosen solution rank is determined by how many other solutions it dominates with non-dominated solution in POS holding a rank of 1 as shown in Figure 2.14. In order to maintain diversity MOGA implemented a fitness sharing measure based upon *niching* suggested by Goldberg [88]. A niching criterion determined dense locations within the response space where solutions were located within a pre-specified distance to each other (σ_{share}) to have their fitness reduced. This encourages the EA to search the entire feature space for alternative non-dominated solution promoting diversity within the population.

Srinivas and Deb [207] developed NSGA and updated to NSGA-II which is a PD-MOEA which uses: non-dominating sorting method, a crowding parameter and elitism. The non-dominating sorting method is similar in the sense that solutions are incrementally ranked based upon how many solutions they dominate and are sorted into sets of corresponding ranks (S_p). The variation occurs in the diversity management step whereby a niching parameter isn't used but the distance between solutions within the same rank is calculated. The distance within a (S_p) is used to partially order solutions, so solutions in higher ranks and less dense spaces within the response space are more likely chosen during selection. The final evolutionary function is an elitism parameter which is used to maintain good solutions through generations of the EA. Elitism operators work by maintaining an external set of solutions that are updated with each generation of the EA but are not involved during the population update procedure. By maintaining an external set, a potentially good solution may not be lost from one generation to the next preventing generational divergence, explained in more detail in Section 3.3.5.

The final state-of-the-art PD-MOEA is SPEA/SPEA-II developed by Zitzler and Thiele [253] which utilise another alternative ranking measure called *strength value* based upon how many solutions it dominates and it is dominated by. They also implement an elitist set which is filled after each generation with non-dominating solutions. However, when the elitist set is overfilled, it is truncated using a nearest neighbours' algorithm to remove solutions until a fixed size is reached.

PD-MOEA's is a thoroughly research field of MOEA literature with a breadth of knowledge and application onto MOP's. However, they are limiting in their applicability to MOP's, with greater than 4 objectives known as Many-Objective Optimisation Problem (MaOP). Since as the number of response variables increase there will also be an increase in the total number on non-dominated solutions within the POS and thus require increasing EA population sizes. By increasing population sizes in accordance with these issues will result in greater computational complexities during optimisation [110] [81].

Performance Indicator Multi-Objective Evolutionary Algorithms

A Performance Indicator Multi-Objective Evolutionary Algorithm (PI-MOEA) is an MOEA which uses a Performance Indicator (P_I) measure of a solutions contribution towards PF as part of the POS as a combination of **Design Goal 1 and 2** [217]. By utilising a P_I, a quantitative measure of the quality of solutions in the POS can be determined and subsequently optimised. Within PI-MOEA literature there are a variety of P_I available: Hyper-volume [136], S-Metric [29], and Lesbesgue measure [138]. These are then incorporated into EA optimisers to produce state-of-the-art PI-MOEA's which include: IBEA [254], SIBEA [41], and SMS-EMOA [29].

$$\mathrm{HV}\left(\mathrm{f}^{\mathrm{ref}}, \mathbf{x}_{\mathrm{POS}}\right) = \Lambda\left(\bigcup_{\mathrm{x}\in\mathbf{x}_{\mathrm{POS}}} \left[\mathrm{f}_1(\mathbf{x}), \mathrm{f}_1^{\mathrm{ref}}\right] \times \ldots \times \left[\mathrm{f}_{\mathrm{N}}(\mathbf{x}), \mathrm{f}_{\mathrm{N}}^{\mathrm{ref}}\right]\right) \tag{2.46}$$

where, $\mathrm{HV}\left(\mathrm{f}^{\mathrm{ref}}, \mathbf{x}_{\mathrm{POS}}\right)$ is the hyper-volume approximation dependent upon $\mathrm{f}^{\mathrm{ref}}$ the response reference point and $\mathbf{x}_{\mathrm{POS}}$ the experiment members of the POS. Λ is the Lebesgue measure.

Firstly, the Hyper-volume measure is the most widely used P_I measure due to its known theoretical properties and thorough application as shown in Equation 2.46 [12]. The Hyper-volume is the volume that a non-dominated solution occupies within the response space in correspondence with a reference point, illustrated in Figure 2.15

As shown in Figure 2.15 the Hyper-volume indicator is maximised as a solution converges towards the true PF and is in less dense sections of the POS in accordance with **Design Goal 1 and 2**. This is supported in the work of Zavala et al. [242], which validated the hypothesis that the Hyper-volume indicator is maximised if and only if the PF contains only Pareto optimal solutions.

In Beume et al. [29], the S-Metric is used in SMS-EMOA which is formulated as a combination of the Hyper-volume indicator and the non-dominated sorting from NSGA-II [55]. SMS-EMOA generates a single solution per population in each generation to be added to an external archive set that is sorted using non-dominated sorting. The archive set is truncated iteratively to remove the lowest contributing Hyper-volume in the worst ranked front in the archive set. Knowles and Corne [126] demonstrated that the Hyper-volume indicator is highly dependent on the correct reference point which will also impact the non-dominated sorting and handling of boundary solutions.

The main advantages of using a PI-MOEA is their quantitative selection measure and pressure during the optimisation of MOP's which assesses both

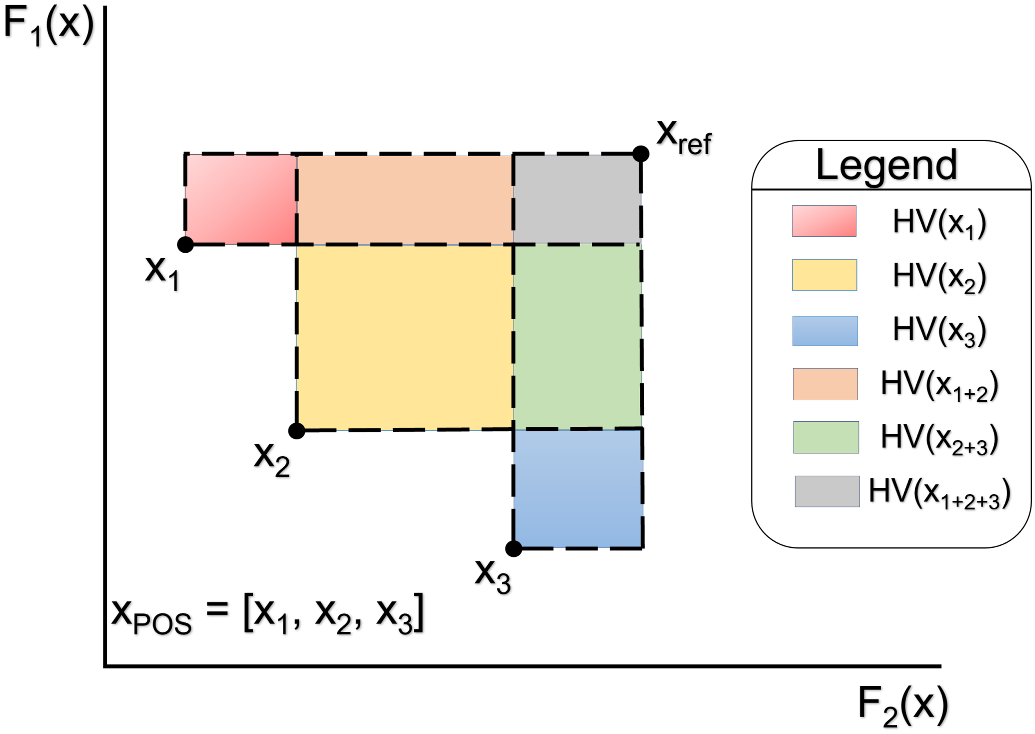


Figure 2.15: Graphical representation of how hyper-volume is quantified along the Pareto Front (PF) in a simple 2-Dimensional minimisation problem.

the convergence and diversity of solutions within the population [217]. Also, unlike PD-MOEA's which have scalability issues, PI-MOEA's are suitable for application onto MaOP's [244]. Whereas for disadvantages firstly, the diversity of P_I's available mainly stem from the Hyper-volume indicator [29]. Secondly, is there computational complexity corresponding to a large computational cost that is polynomial with population size and exponential with the number of response variables [29].

Although, there are some works within the literature to tackle computational complexity by producing more efficient optimisation procedures using Monte Carlo simulations in HypE [14], there is also research into more efficient Hyper-volume based indicators in R2 [42] and δ_p [191] [217].

Multi-Objective Evolutionary Algorithms via Decomposition

A Multi-Objective Evolutionary Algorithm via Decomposition (MOEA/D) is an MOEA which converts an MOP into a set of SOP's which are then optimised simultaneously using a scalarisation function and weight vectors. The neighbourhood collaboration occurs during the mating step (See Section 3.3.2) to encourage good solutions with similar weight vectors to mate prior to the implementation of genetic operators, as shown in Figure 2.16

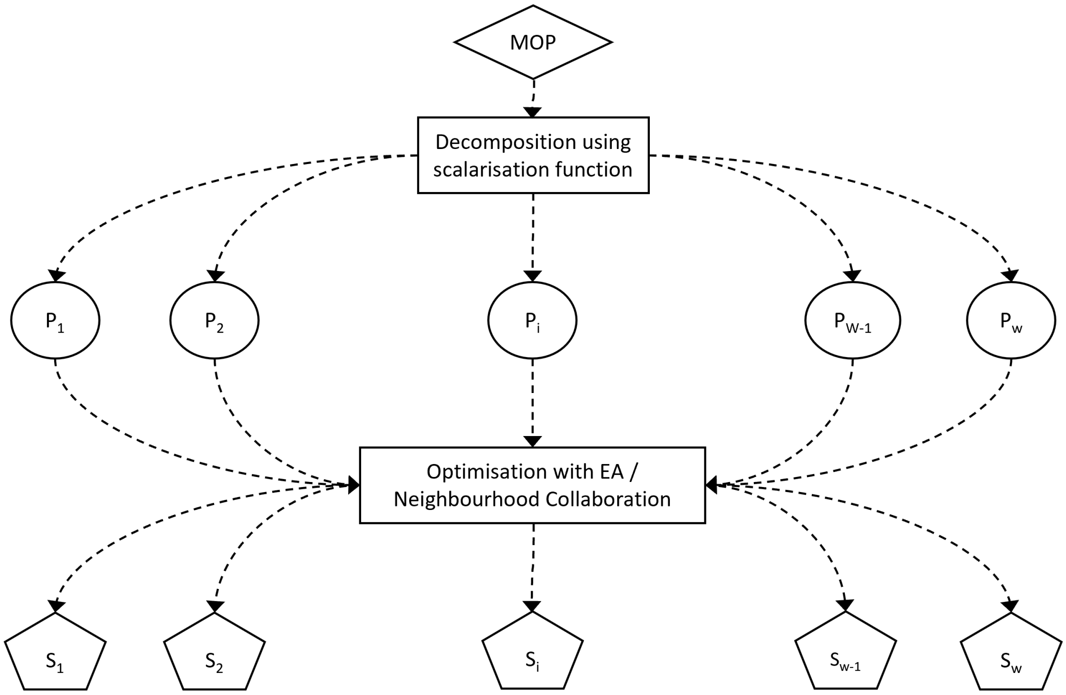

MOP
Decomposition using scalarisation function
P_1
P_2
P_i
P_{W-1}
P_w
Optimisation with EA / Neighbourhood Collaboration
S_1
S_2
S_i
S_{w-1}
S_w

Figure 2.16: Flowchart detailing the logic process for an Multi-Objective Evolutionary Algorithm via Decomposition (MOEA/D) splitting an MOP into several SOP

The field of D-MOEA was inspired by the publication of [247] on their MOEA/D algorithm, after which many approaches have been published to expand and improve upon their initial approach which is thoroughly reviewed in Trivedi et al. [217]. D-MOEA are comprised of two main variations upon other MOEA's: the method of scalarisation of MOP into SOP's and the method of neighbourhood collaboration.

In order to decompose a MOP into several SOP's a scalarisation function is required and the diversity of the POS is highly dependent upon the weight vector set used [217]. However, an evenly or uniformly distributed weight vector set does not guarantee an evenly or uniformly distributed POS [247], especially for MaOP's [82] [83]. Thus, some authors have sought to develop methods to generate weight vector sets that would satisfy **Design Goal 2** such as the work by Giagkiozis et al. [82]. In Giagkiozis et al. [82], Generalised

Decomposition is used to select an optimal weight vector based upon the perceived shape of the PF. However, the disadvantage of this method is that the shape of PF is often unknown and thus a linear approximation of the PF is used. The approximation was able to achieve good performance albeit not optimal due to lack of prior PF shape knowledge.

Three main scalarisation functions were assessed in [247]: Weighted Sum (WS, See Equation 2.45), Tchebycheff (TBF, See Equation 2.47), and Penalty Boundary Intersection (PBI, See Equation 2.48).

$$S_{TCH} = \max_{1\leq i\leq N}\left[w_i \left| f_i\left(\mathbf{X}\right) - z_i^*\right]\qquad z^* = \left[z_1^*, \dots, z_M^*\right] \tag{2.47}$$

where, w_i is the i^{th} weight parameter, z_i^* is the i^{th} reference point, and i refers to its corresponding response variable.

$$S_{PBI} = d_1 + \theta d_2 \tag{2.48}$$

where,

$$d1 = \frac{||a||}{||w||}$$

The effect of various scalarisation functions affect how they search the response space during an optimisation which can be shown illustratively for a 2-Dimensional MOP in Figure 2.17 [227]. As seen in Figure 2.17 region A determines which new solution can be used to improve upon previous solutions and its shape is determined by its contour. The contour property is unique to each scalarisation function utilised which was investigated in [58]. Derbel et al. [58] demonstrated that depending on the MOP's specific PF geometry different scalarisation functions would be more suitable than others such as WS performing better on convex PF's vs non-convex PF's [217].

The neighbourhood mating scheme was the 2^{nd} core principle which drove D-MOEA's unique formulation to differentiate from PD-MOEA's and PI-MOEA's. The neighbourhood mating scheme defined a neighbourhood set of size T around each sub-SOP to generate new offspring solutions from [247]. However, an issue in Zhang and Li [247] MOEA/D was during mating replacement, a single optimal solution could replace multiple solutions in multiple sub-SOP's simultaneously [217]. If left unchanged this would lead to potential loss of population diversity with each successive generation.

This led to the development of an additional constraints to be used with scalarisation functions to extend D-MOEA [217] to MOEA/D-CD and MOEA/D-

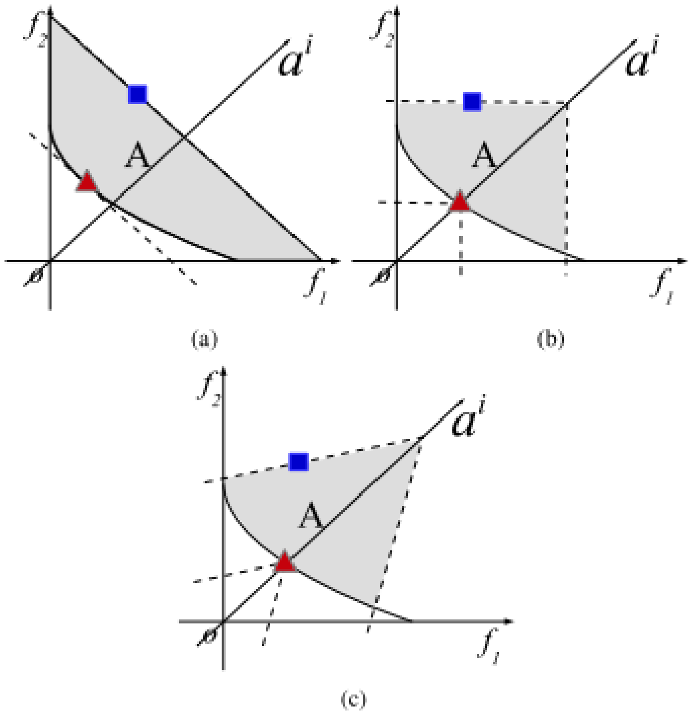


Figure 2.17: Illustration of the effect of most popular scalarisation functions in D-MOEA literature on a 2-Dimensional response space. A represents the improvement region of each response space: (a) Weighted Sum, (b) Weighted Tchebycheff, (c) Penalty Boundary Intersection. The square point is the current best solution for each sub-SOP with the triangle point indicating the next optimal solution in the neighbourhood along direction vector a^i, taken from [227].

ACD. MOEA/D-CD and MOEA/D-ACD used constant and adapting constraints respectively, to alter the contour property to minimise excessive replacement. Thus, increasing diversity and improve optimisation performance if the constraints were tuned correctly.

Whilst Zhang and Li [247], utilised constraints to minimise excessive replacement Giagkiozis et al. [83] suggested changing how the neighbourhood mating pools were defined. As originally the neighbourhood mating pools were defined by proximity of weight vectors to one another, this did not guarantee proximity of sub-SOP solutions in the response space. Giagkiozis et al. [83] also suggested adaptively updating the neighbourhood sets as the population of solutions would also evolve through the optimisation.

The main advantages of using D-MOEA's is their flexible and scalable applicability to MOP's, constrained MOP's, and high dimensional MaOP's

[244]. By leveraging the scalarisation functions to decompose complex MOP's and MaOP's into a set of sub-SOP's and solve each in parallel. On the other hand, the main disadvantage of D-MOEA's is limited application and research in comparison to PD-MOEA and PI-MOEA due to interest being sparked from Zhang and Li [247] in 2007. Thereby lacking equivalency in theoretical property analysis and implementation in a variety of fields.

2.4 Literature Analysis

In BO there has been a wide uptake of use for the methodology in a large variety of fields including Design of Experiments or Experimental Design (DoE) [92]. Although Bayesian Optimisation (BO) is continuously being developed and updated for an ever-increasing variety of use-cases such as: high-dimensional optimisation, multi-objective optimisation, multi-fidelity optimisation, constrained optimisation, etc [92]. these approaches are not always assessed on practical real-world application studies. Thus, it is difficult to assess whether the developed techniques are applicable for manufacturers to implement on a larger scale especially with the increased drive towards Industry 4.0.

Therefore, a systematic search of the DoE literature for implementations of BO was conducted through a search of Google Scholar using the following keywords: Bayesian optimisation, sequential, adaptive, acquisition or utility function, and Design of Experiments or DOE. The findings suggest the use of Bayesian Optimisation (BO) as a DoE on application cases is a relatively limited field with 100 papers and the majority of application papers being published within the last five years. The most prominent fields of application which utilised BO were Aerospace/Automotive [9] [180] [142], Pharmaceuticals [223], Chemistry [199] [129], Biology [187], Electronics [56] [52] [196] [172], Engineering [137] and Additive Manufacturing (AM) [246].

Of the DoE applications implemented the BO methodology most frequently implemented was the sequential BO using mainly the Expected Improvement (EI) acquisition function [56] [52] [187] [137], with some methods utilising GP-CB acquisition function [196] [137] as well as ensemble approaches [172]. After which the next most frequently implemented DoE approach was Multi-Objective Bayesian optimisation [142] [129] albeit with half as many publications using some variation of the EI acquisition function. The remaining DoE implementations using BO were for more specialised problem types including: constrained [180], batch [199] and multi-fidelity [9].

One field of DoE application that is beginning to see increasing interest is AM with 8 papers within the last five years [246]. Of these 8 papers the vast majority also implemented sequential BO using primarily the EI acquisition function [104] [238] [163] [90]. Whilst there were also one paper each for sequential BO using GP-CB [57], constrained optimisation using an cEI (see Section 2.2.5) [198], and multi-fidelity optimisation [236].

Chapter 3

Methods and Manufacturing Case Studies

In the previous chapter DoE's history and progression as a field of study was introduced including its background concepts to provide a basis of knowledge for the research conducted in this thesis. In this chapter, further detail will be provided on key methods, performance measures, and processes to be used throughout the thesis to support the research.

3.1 Data Processing

In traditional regression analysis or modelling, the processing of both input and output data can play an important role in ensuring well-tuned models with good inference performance. Input variables are processed to minimise the impact of differing input scales/ranges. Whilst output variables are processed to centre on a zero mean to simplify GPR model assumptions (See Section 2.2.1).

Processing of input data is particularly important in application where models are based upon distance/similarity between data points [182] such as GPR models. As GPR models determine the covariance between two data points if the scale/range of one variable far exceeds the remaining variables it will likely dominate any subsequent covariance calculations such as predictive mean and variance in Equation 2.21a and 2.21b respectively.

The method of processing the regression data for modelling using GPR in this work is known as Standardisation. In this thesis, the standardisation approach implemented is referred to as Standard Score or Z-scores which is calculated using Equation 3.1.

$$v^{score}\left(x_i\right) = \frac{x_i - \mu\left(x_i\right)}{\sigma\left(x_i\right)} \tag{3.1}$$

Where, each experiment x can be broken down into each of its i^{th} input variables the $v^{score}\left(x_i\right)$ standard score of the i^{th} input variable can be determined using: the mean of the i^{th} input variable $\mu\left(x_i\right)$ and the standard deviation of the i^{th} input variable $\sigma\left(x_i\right)$.

The mean and standard deviation of a variable to be standardised can be found from the data set accurately if a large volume of data is available. However, the application cases in B-DoE studies and Additive Manufacturing (AM) are expensive to evaluate and thus have limited experiments/experimental budgets. ITherefore, an assumption surrounding the distribution of input variables is made to allow for the approximation of the mean and standard deviation of input variables. As the selection of any single input variable setting is equally likely the variables are assumed to be distributed uniformly in a continuous space. By using the properties of a continuous uniform distribution Equation 3.2a and Equation 3.2b are used to estimate the mean and standard deviations of each variable respectively.

$$\mu\left(x_i\right) = \frac{1}{2}\left(x_i^L + x_i^U\right) \tag{3.2a}$$

$$\sigma\left(x_i\right) = \sqrt{\frac{1}{12}\left(x_i^U - x_i^L\right)^2} \tag{3.2b}$$

Often the upper and lower bounds required for standardisation are known apriori for input variables but unknown for output variables. In order to account for the unknown bounds of the output variables they must be re-estimated after each iteration of the DoE to provide new upper and lower bounds. These updated bounds can then be utilised with Equation 3.2a and Equation 3.2b to re-standardise the output data after each iteration of DoE.

3.2 Algorithm Performance Metrics

In traditional DoE methodologies the experiments are all selected *a priori*, after which experiments are performed with an ANOVA statistical test being conducted last. Whereas, when implementing adaptive DoE or BO in this thesis, analysis of performance can occur at each iteration of experimentation. The DoE aims to satisfy the goals of maximising information gain, minimising experiments performed, develop accurate surrogate models and to optimise the process parameter settings. The performance metrics can be used to assess which of these goals the DoE methods can satisfy as well as compare against alternative methods. For this work two performance measures are required to illustrate these DoE goals:

- Global Optimisation Performance:

  Ability of an algorithm to locate globally optimal design parameters within a design space.

- Statistical Model accuracy/precision:

  Ability of a statistical model to encapsulate underlying patterns and interactions to allow for accurate inference of unexplored locations.

3.2.1 Optimisation Metric: Regret

The metric for assessing an algorithm's capability in locating the globally optimal parameter setting that has been used throughout adaptive DoE or BO literature is, *Regret* [208] [230] [100] [101]. When making decisions under uncertainty an individual's choice of their perceived best course of action does not always lead to the best possible decision in that instance, the response to this disparity is often referred to as Regret. Regret defines the disparity between the choice made under uncertainty and the best possible decision, in decision theory. [24].

In decision theory there are multiple variations of types of Regret that can be calculated, however, for use in DoE or BO literature the most suitable type is the *Immediate Regret*.

$$R_t = \left| f\left(x^+\right) - f\left(x^*\right) \right| \tag{3.3}$$

Where, $|...|$ is the absolute function, R_t is the regret of the decision made at iteration t, x^+ is the incumbent which is the current best experiment performed and x^* is the global optimum. R_t represents the difference between the best choice made at interval t and the globally best choice (global optima) that could have been made. This metric is a simple calculation to implement but also accurately represents the progress of searching the decision space for the global optima. This is often the only metric which is considered when comparing algorithm frameworks in literature. Their goals are to improve upon the current state-of-the-art methods in locating globally optimal solutions within a minimum number of intervals t.

3.2.2 Statistical Model Error Metric: NRMSD

The goal of this work is not to simply improve upon the state-of-the-art method but also to provide a suitable framework for implementation in industry. As previously specified in Section 3.2, the goals of B-DoE are to produce accurate/precise surrogate models of the underlying process and their variable interactions. The prediction or inference accuracy of statistical models is often overlooked when assessing algorithmic performance as it is assumed that the models produced are accurate.

As shown in Section 2.2 the selection of each new experiment is built upon the inference of GPR models, the accuracy of models directly affects the optimisation performance. By monitoring the accuracy improvements of the surrogate models with each experiment selected this can provide insight into the algorithms performance in exploring the decision space.

A variety of performance metrics are implemented in regression/machine learning literature with changing preference year by year [34]:

1. Mean Squared Error (MSE)

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{3.4}$$

2. Root Mean Squared Error (RMSE)

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} \tag{3.5}$$

3. Mean Absolute Error (MAE)

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \tag{3.6}$$

where, N is the total number of test predictions, $\hat{y}_i$ is the predicted output of test i and y_i is the true output of test i.

MSE is perhaps one of the more frequently applied regression performance metrics due to its simple calculation method as shown in Equation 3.4. Despite its simple calculation method which leads to its primary advantage, it also pertains to one of its primary disadvantages as well since the MSE is sensitive to outliers in its residual $(y_i - \hat{y}_i)$. Another consideration to be made is that the unit of MSE due to squaring the residual is not directly relatable to the units of the data. This led to the adoption of RMSE whereby it takes the square root of the residuals as shown in Equation 3.5. Another frequently used choice is MAE which instead of taking the square of the residual to remove negative values it instead takes the absolute of the residual as shown in Equation 3.6. Also, just as with RMSE, MAE units are the same as the data's initial units, allowing for improved interpretability.

Given the properties of the modelling performance metrics the most suitable metric would be either RMSE or MAE, however another consideration should be accounted for when choosing a suitable metric and that is comparative ability. As with both metrics the quantities are sensitive to the scale of the response variable being analysed [182]. Therefore, in order to compare modelling performance on different types of problems, to make the models errors independent of response scale, it is suitable to use a normalised metric. Consequently Normalised Root Mean Square Deviation (NRMSD) would be a suitable modelling metric to use for this work:

$$\mathrm{NRSMD} = \frac{\mathrm{RMSE}}{\left(y^{U} - y^{L}\right)} \tag{3.7}$$

Where, y^{U} is the maximum value or upper limit of the predicted output variable and y^{L} is the minimum or lower limit of the predicted output variable.

3.3 Global Optimisation Algorithm: Genetic Algorithm

As specified in Section 2.2 and Section 2.3, Genetic Algorithm (GA) are a suitable optimisation approach for global optimisation problems that seek to isolate a globally optimal solution using a population-based search built upon the principle of *survival of the fittest*.

This core theory behind this principle as an optimisation approach is to generate a set of random solutions or *Individuals* which constitute a combined *Population* of solutions which are augmented through iterative *Generations* using evolutionary-based functions to progressively evolve the population towards an optimal solution [54]. This procedural evolution uses the concept that with each successive generation the good/optimal controllable variables values *Genes* converge towards the optimal solution. Whilst the baseline GA is a simple algorithm shown in Figure 3.1, using a variety of modified or add-on evolutionary functions allows the developer to augment its features. To improve performance or adjust application onto various types of optimisation problems, aside from the initial sequential problems others include: batch, multi-objective, and multi-objective batch.

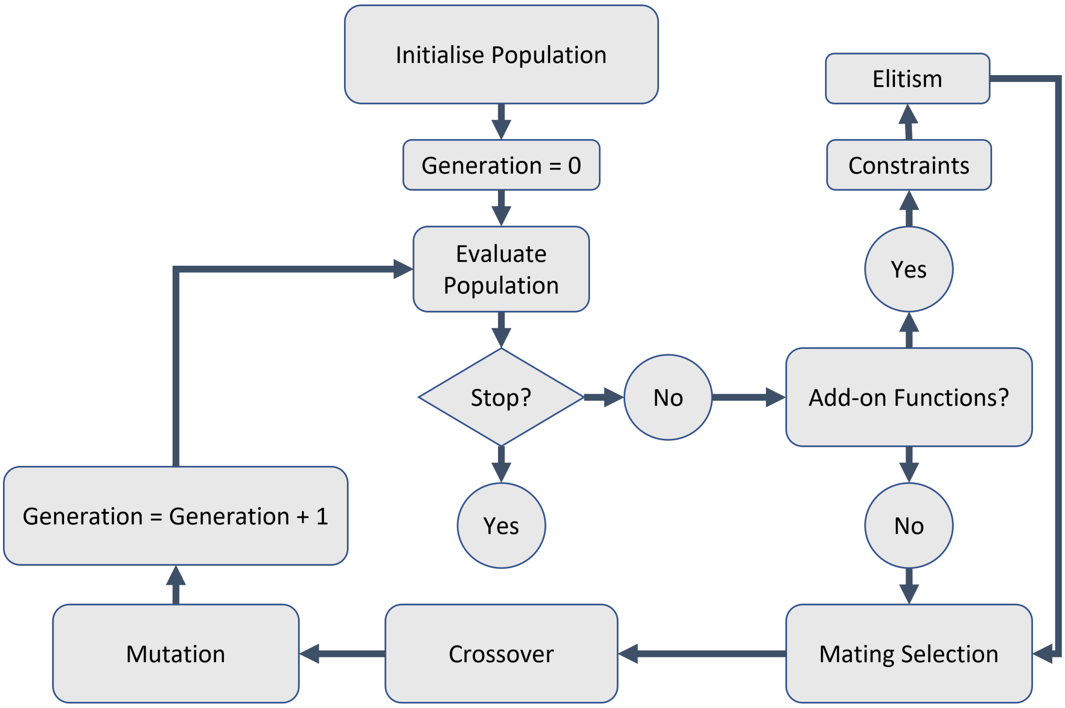


Figure 3.1: Flowchart illustrating basic structure of an Evolutionary Algorithm, Genetic Algorithm.

Figure 3.1 Illustrates the basic structure of a GA with its primary evolutionary functions: population generation, mating pool selection, crossover, and mutation. In addition to the primary evolutionary functions, there are a series of additional add-on functions that will be utilised to improve GA's performance throughout including: elitism, and population constraints.

3.3.1 Population

The first step in the GA algorithm is to generate an initial set of individuals which will make up the initial population. Each individual C is split further into components known as *Genes* however, these genes can encode information by two different methods: Binary and Real Parameter encoding [54].

The method of encoding is dependent on the type of problem that is being optimised, a binary encoding is suitable for discrete-space controllable inputs, and real parameter encoding is suitable for continuous-space controllable inputs [54]. The issue of using a binary encoding in a continuous space is a matter of attainable precision. The binary string length to encode discrete value into a binary string must be known *a priori*, a predetermined amount of information can be encoded into each gene, which limits controllable inputs precision in the continuous space. Therefore, in a DoE setting of manufacturing industries the majority of problems involve a continuous controllable inputs, so henceforth real parameter encoding and evolutionary functions shall be used:

$$C = \left[x_i, x_{i+1}, \ldots, x_d\right] \qquad \forall \quad i = 1, 2, 3, \ldots, d \tag{3.8}$$

where, C is the real-parameter encoded individual and x_i is the real-parameter value of gene i representing the i^{th} input variable.

In order to generate an initial set of individuals the boundaries of the controllable factors need to be defined. After which, drawing from a random distribution and scaling the draws to within the bounds of each gene will generate the initial population prior to implementation of the GA.

3.3.2 Mating Selection

Once the initial population has been generated the first generation of the GA begins. Prior to any evolutionary functions evolving a population the cost of each individual in a population is calculated. The cost assignment employed is dependent on the type of problem being solved, wherein we are assessing an experiments expected value if performed. Therefore, the individuals that produce a large expected value or acquisition cost should be selected, the specific cost applied will be dependent upon the B-DoE acquisition function being employed, with the acquisition functions being investigated detailed in Section 2.2.

Once the population has been evaluated and cost assigned to each individual in the population, a convergence criterion is assessed such as: wall-clock time [152], maximum number of generations [152], small improvements in optimum located [152], and small improvements in acquisition value [147] [171]. In order to implement a stopping criterion using incremental changes in either a global optimum or acquisition value, a tolerance must be set and tuned which can be difficult to determine [152]. On the other hand, using a maximum number of generations could lead to a waste in computational budget if set too large, but it is a much simpler scheme to utilise. If the criterion is met then the GA will end and export the optimal solution, whereas in the case the criterion is not met the population will be modified. Using three main evolutionary functions, a GA aims to duplicate good individuals and eliminating bad individuals [54].

The first evolutionary reproduction function is the mating pool selection, which determines which two or more individuals in a population will combine their encoded genes with the intention of generating improved solutions during the subsequent reproduction stages. There exist a large variety of selection schemes but can these be broadly classified into 3 types [54]:

- Proportionate Selection
- Ranking Selection
- Tournament Selection

A proportionate selection scheme sets the likelihood of selecting an individual to add to a mating pair as proportional to its value or cost in relation to the total available cost of the entire population:

$$P\left(C_i\right) = \frac{\mathrm{eval}\left(C_i\right)}{\sum_{j=1}^{P} \mathrm{eval}\left(C_j\right)} \tag{3.9}$$

where, $P(\ldots)$ is the probability of selection of an individual, $\mathrm{eval}(\ldots)$ is the evaluated value or cost of an individual, and $\sum_{j=1}^{P}$ indicates the sum of all evaluated values/costs of individuals in the population of size P.

The most common implementation of this scheme is a roulette wheel selection scheme where the proportion of each individual is scaled to fit onto a roulette wheel, as seen in Figure 3.2. After which the roulette wheel is spun each time to select a solution to add to the mating population up until the number of mating pairs is reached [54].

The primary issue with using a proportionate selection scheme is that it has a scaling problem, whereby if a single individual has a significantly larger cost than the remaining population its probability of selection will dominate the remaining individuals. As such this would lead to the repeated selection of a single individual into the mating pool, increasing the likelihood of the population converging and losing population diversity which would inhibit convergence to the optimum.

To prevent this, a ranking selection was introduced [88]. Ranking selection first sorts individuals according to their costs from the worst (rank 1) to the best (rank N) [54]. Individuals in each rank are then assigned a new cost equal to their ranking upon which, using the same underlying functions as proportionate selection schemes, they choose the mating pairs.

According to [89], the tournament selection scheme is the best in terms of convergence and computational time than other methods. Another advantage

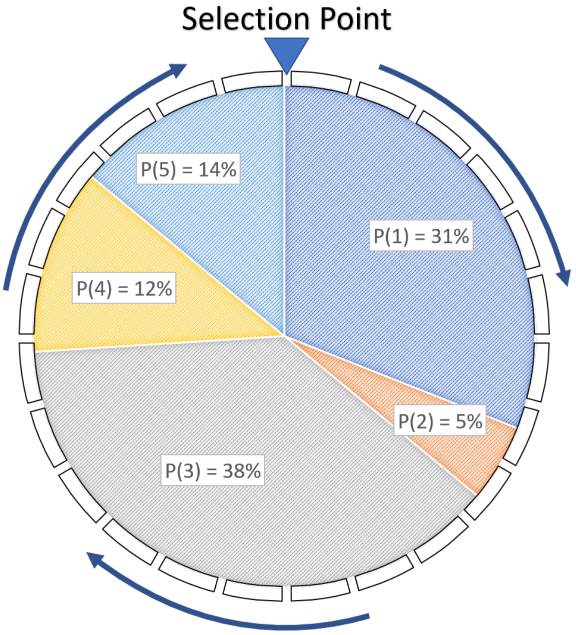


Figure 3.2: Example of Roulette Wheel Selection strategy for 5 possible individuals of varying fitness and subsequently probability of selection determined using Equation 3.9.

it is also unaffected by the scale of the cost function [15] unlike proportionate methods. In a tournament scheme, individuals are added to a *Tournament* in which the best individual is selected to be added to the mating pool as seen in Figure 3.3.

A tournament selection scheme can be varied by changing its tournament size (S) and whether individuals are replaced after each tournament. If S = 1 the tournament would be equivalent to random selection, on the other hand if S = N only the highest cost individual would be selected each tournament. Thus, a balance is required when selecting the tournament size wherein Blickle and Thiele [32] found when individuals are replaced after each tournament if S ≥ P the loss of population diversity with each generation would be over 50%. An important factor to consider is when using a lower S with replacement after each tournament can lead to some individuals not participating in any tournaments. On the other hand, using a tournament selection scheme without replacement, whilst guaranteeing that each individual participates at least once, it does not prevent loss of diversity as bad individuals will always be removed through selection. The main issue is to prevent loss of good solution diversity, which can be maintained through using other evolutionary

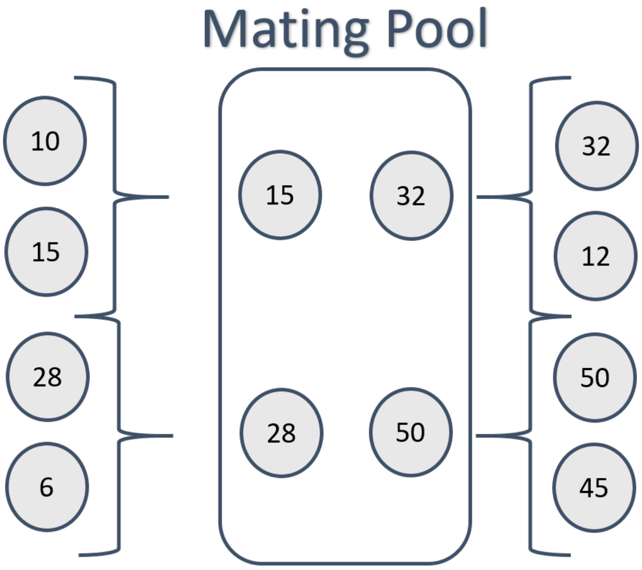


Figure 3.3: Diagram demonstrating an example of a binary tournament (S=2) selection scheme on a population of 8 individuals.

functions as described in Section 3.3.5.

3.3.3 Crossover Operator

After a parent sub-population has been chosen from the current generation total population, new individuals or *Offspring* are generated through crossover functions. The purpose of the crossover function is to expand upon the work done by the selection operator. Selection chooses individuals with good/desirable characteristics (genes) to carry on into future generations, the crossover function is to generate new potentially better solutions by recombining parents of superior genes [54]. Therefore, crossover guides the individuals to converge towards optimal regions of the design space.

This work implements a real-parameter crossover function known as *Blend Crossover* (BLX-a), which blends the two parents along every gene using a blending variable which is re-calculated for each mating pair.

$$\begin{aligned} O_{(1,t+1)} &= \left(1-\gamma_{(i)}\right)\times C_{(1,t)} + \gamma_{(i)}\times C_{(2,t)} \\ O_{(2,t+1)} &= \gamma_{(i)}\times C_{(1,t)} + \left(1-\gamma_{(i)}\right)\times C_{(2,t)} \end{aligned} \tag{3.10}$$

where, O_1 and O_2 are the offspring created via crossover, γ is the blending operator is re-drawn after each crossover according to Equation 3.11 which is re-calculated for each mating pair.

$$\gamma_{\mathrm{i}} = (1 + 2a)\,\mathrm{rnd}_{\mathrm{i}} - a \tag{3.11}$$

Where, a is the blending constant and $\mathrm{rnd}_{\mathrm{i}}$ is a random number drawn from a uniform distribution between (0,1). $\mathrm{rnd}_{\mathrm{i}}$ ensures a degree of randomness in the mixing between mating pairs from one pair to the next. Deb [54] states that a was found to be most optimal to be set to 0.5. Whereby the maximum variation achievable is 1.5 times over or under their parents gene values within the upper and lower bound limits as illustrated in Figure 3.4.

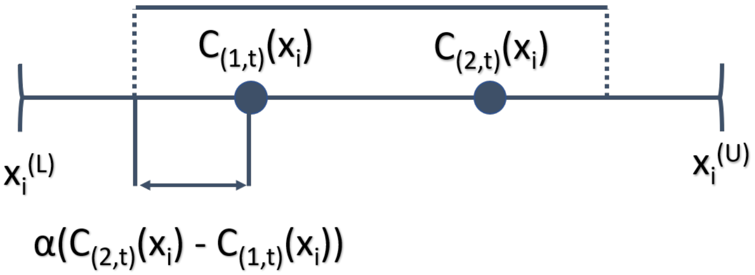


Figure 3.4: Total BLX-a crossover spread possible when using a blending constant (a) of 0.5.

3.3.4 Mutation Operator

The crossover operator guides the GA to convergence through repeated crossover of individuals with desired *Genes* over many generations, however there is no guarantee of convergence. This is due to the inherent diversity within the population of individuals, as the population may be trapped in a local optimum. Therefore, as the goal of a GA is to locate a globally optimal solution, another evolutionary function must be utilised to improve the diversity of the population. To ensure generational convergence to a global optimum rather than a local optimum, diversity of the population needs to be maintained through generations. GAs incorporate the *Mutation* operator to maintain population diversity [54].

This thesis uses a simple mutation function known as *Random Mutation*, which randomly selects an individual from the available offspring population onto which a random gene is specified on that individual to mutate. This gene is mutated by randomly drawing a new gene from within the bounds of the search space around its upper and lower bounds [159].

$$x_i^M = rnd_i \times \left(x_i^U - x_i^L\right) \qquad \forall \quad i = 1, 2, 3, \ldots, N \tag{3.12}$$

Where, x_i^M is the mutated form of gene i, rnd_i is a random number drawn from a uniform distribution between (0,1), x_i^U is the upper limit of gene i and x_i^L is the lower limit of gene i. This evolutionary function is equivalent to random initialisation of a new gene on an individual in order to create a new individual in the search space in order to maintain diversity.

After a small sample of the offspring population have been mutated, they are placed back into the offspring population. The offspring population and the prior generation of individuals are then combined into a single population and incremented into the next generation as shown in Figure 3.1.

3.3.5 Additional GA Functions

Apart from the core evolutionary functions required to implement a GA successively, there is a variety additional functions available to improve deficiencies and/or lack of capability to handle specific goals of the optimisation framework. Two such evolutionary functions are to be described below which are implemented alongside the selection, crossover and mutation functions.

Elitism Operator

Methods to maintain genetic diversity of a population of individuals have been achieved through the use of mutation functions to introduce new solutions into the population. Also as mentioned in Section 3.3.2 during selection with replacement, there is a chance of solutions not being selected to participate in a single tournament. This can lead to loss of diversity through random chance. In order to prevent these losses of diversity in the population of solutions most GAs introduce an *Elitism* policy [54].

Elitism is a method to store and preserve the best solutions found from one generation to the next in an external population, that is maintained and updated through each successive generation. In this work, an external elitist set policy of a fixed size dependent on the size of the GA population is defined in Equation 3.13.

$$E_{size} = \frac{1}{10} \times P_{size} \tag{3.13}$$

where, E_{size} is the size of the elitism set and P_{size} is the size of the GA population.

As the elite set is stored externally their costs and constraints must be re-evaluated from one generation to the next, in parallel with the current GA population. After constraint application (Section 3.3.5), the updated elite set is compared with the current population to identify potential new individuals to be added to the external elite set. Once new elite individuals are identified and added to the external elite set, any identical elite individuals are removed with the worst performing elites being culled to the pre-specified size defined in Equation 3.13. The elite set updating procedure is performed in accordance with Algorithm 3.1.

Algorithm 3.1 Elite Set Update

```
Inputs:
        P          : Population of Individuals
        E          : Elite Population
        $E_{size}$ : Elite Set Size
        u          : Acquisition value

P ← Constraint Penalty (P)                                    // See Equation 3.14
E ← Constraint Penalty (E)                                    // See Equation 3.14
procedure Elite Set Updater(P, E, $E_{size}$) do
    $E_{potential} \leftarrow \text{find}\left(P(u) > E_{old}\left(u_{lowest}\right)\right)$
    $E_{new} = \left[E_{old}, E_{potential}\right]$
    $E_{new} \leftarrow \text{Unique Sorting}\left(E_{new}\right)$            // Identical candidate removal
    if Size($E_{new}$) > $E_{size}$ then
        $E_{new} \leftarrow \text{Order Sorting}\left(E_{new}\right)$             // Descending sort
        $E_{new} \leftarrow \text{Removal}\left(E_{new}\left(u_{lowest}\right), E_{size}\right)$   // Remove worst elites
    end if
    return New external elite population, $E_{new}$
end procedure
```

These elite individuals are kept external to the main population across every generation except during mating pool selection tournaments, where they are inserted into the available pool. When the stopping criteria is met they are also added during the final optimal individual selection.

Constraint Functions

In GAs, the optimiser is capable of searching the design space globally and efficiently however in many benchmark problems and manufacturing design problems a feasible search space is required. This search space domain is often defined using the upper and lower bounds on the controllable factors which are incorporated as the real-parameter individuals' genes. In order to ensure that individuals remain within these limits during the optimisation search a *Penalty* or *Constraint* function is applied.

A variety of methods exist to enforce these design space boundaries depending on how strict the boundaries are as well as methods to improve search capabilities close to the design space boundaries [54]. In this work, a strict death penalty [54] is combined with a shrinking border approach in order to maintain strict boundary conditions whilst searching close to boundary conditions.

Constraints are applied after the cost of individuals in a population has been assessed as shown in Figure 3.1. A death penalty constraint function identifies individuals that through crossover, mutation and other evolutionary functions have caused the gene to drift outside their bounds making them infeasible individuals [54]. Upon this identification, their associated cost is reduced to 0, effectively killing the individual as its chance of winning a tournament is removed in accordance with Equation 3.14.

$$\chi(\mathrm{C}) = \chi(\mathrm{C}) * \mathrm{D}_{\mathrm{penalty}} \qquad \mathrm{D}_{\mathrm{penalty}} = \begin{cases} 0, & \text{if } \mathrm{x_i} \geq \mathrm{x_i^U} \\ 0, & \text{if } \mathrm{x_i} \leq \mathrm{x_i^L} \\ 1, & \text{otherwise} \end{cases} \tag{3.14}$$

Where, $\chi(\mathrm{C})$ is the population cost associated with individual C, $\mathrm{D}_{\mathrm{penalty}}$ is the death penalty cost, $\mathrm{x_i}$ is the $\mathrm{i^{th}}$ gene of individual C and $\mathrm{x_i^U}$ and $\mathrm{x_i^L}$ are the upper and lower bounds respectively of the $\mathrm{i^{th}}$ gene of individual C.

As a death penalty constraint is a strict constraint applied to prevent infeasible individuals from being retained within a population. It can inhibit the search for feasible individuals close the boundary as even a slight overstep is a violation. Thus, in order to prevent the death penalty constraint from inhibiting the GA searching close to the boundary, a slightly larger boundary domain is initially allowed. In Figure 3.5, a temporarily increased boundary is utilised to allow for searching close to and beyond the original boundary (Personal Communication, K. Grzędziński 2019).

By gradually shrinking the boundary over many generations, the conver-

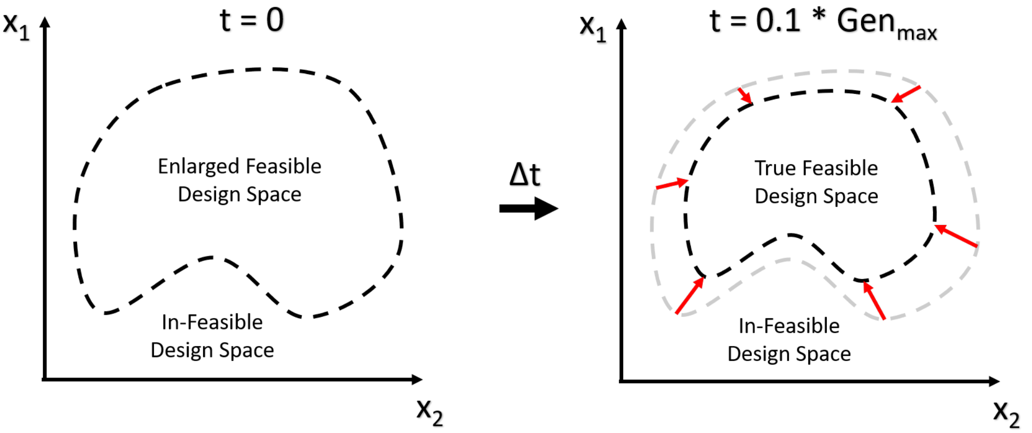


Figure 3.5: Initial increase in search space domain before gradual reduction to the original search space to improve exploration close to boundaries.

gence of potentially beneficial individuals near the boundary can converge within the population. This shrinking occurs slowly over the first 10% of the total generations to allow for increased boundary convergence but is reduced to the original boundary for the remaining generations.

Aside from boundary constraints by using a GA optimiser, it is possible to introduce extra constraints with respect to specific problems which is another benefit in using a GA optimiser for manufacturing-based problems. As problem specific constraints can easily be incorporated into the GA optimiser increasing the domain of problems that DoE with a GA based optimiser can handle [54].

3.3.6 Genetic Algorithm: Parameters

In Section 3.3.1 - Section 3.3.5 a variety of evolutionary functions have been introduced to explain the GAs functionality and set-up. To follow-on from this, there exist a variety of tune-able parameters that require specification for the GA optimiser to run efficiently, as well as reasoning behind their specification. Table 3.1 provides a list of GA parameters that can be varied and their related evolutionary functions.

As with designing a new B-DoE framework that is generalised for application to a variety of problem types in which according to the *No Free Lunch* Theorem [232], there is not a universal solution to its design that is suitable to all problems. In parallel, this theorem can also be applied to the selection and tuning of the GA parameters in which there is no set parameters that will

Table 3.1: GA Parameters that can be selected and tuned depending on the requirements of the optimisation problem.

GA Tunable Parameters		
GA Parameter	GA Function	Symbol
Population Size	Population Generation	P_{size}
Number of Generations	GA Initialisation	N_{gen}
Elite Population Size	Elitism	E_{size}
GA Boundary Stretch Coefficient	Boundary Stretch Coefficient	BS_{size}
Blending Crossover Constant	Crossover	α

prove optimal for all types of problems.

Population size and number of generations in a GA largely influence the total number of function evaluations that will take place during one GA cycle. Thus, the selection of both of these two parameters influences the computational cost of implementing a GA optimiser as shown in Equation 3.15.

$$NF_{evals} = P_{size} \times N_{gen} \tag{3.15}$$

where, NF_{evals} is the total number of function evaluations that will take place during one GA cycle. As such, the choice between both of these parameters falls under two implementation schemes: Large P_{size} with a small N_{gen} or a small P_{size} with a large N_{gen}. Therefore, the choice between both depends on the characteristics of each scheme. Using a large P_{size} increases the populations diversity initially as it allows for a more precise grid of solutions to be used but, a small N_{gen} suggests the GA will not have enough generations to converge to the globally optimal solution. On the other hand using a large N_{gen} provides more opportunities for convergence to the globally optimal solution but limits diversity.

In this work, the large N_{gen} was opted for $\left(N_{gen} = 1000\right)$ as there are evolutionary functions available to increase the diversity of the population as shown throughout Section 3.3, but no functions to ensure convergence within a small number of generations. When considering the population size in the GA, the *Curse of Dimensionality* of problems must be considered. More commonly associated in optimisation literature as *Combinatorial Explosion*, where as the number of controllable factors increases, this leads to the rapid growth of the problem complexity. In order to ensure that there are sufficient individuals in the population with regards to the problem complexity, the population size will scale with the number of controllable factors [54].

$$\mathrm{P}_{\mathrm{size}} = 10 \times \mathrm{d}$$

Where d is the input dimension of the problem of which in DoE is the number of controllable factors. The size of the elitism set must also be chosen prior to the implementation of the GA, which is chosen to be a small subset of the population as specified in Section 3.3.5.

Typically in GAs, the crossover and mutation evolutionary functions often have a probability or rate of occurrence associated with them [54]. These probabilities determine randomly where to crossover between genes of mating pairs or when to mutate particular genes of offspring when using binary encoded GAs. For our real-parameter GA, as long as the individual has been selected using the mating operator it will undergo BLX-a crossover using Equations 3.10 - 3.11 or mutation using Equation 3.12.

The final parameter to be chosen is the size of the shrinking boundary constraint ($\mathrm{BS}_{\mathrm{size}}$) and the speed at which it reduces towards the true problem boundary during the GA. As the purpose of the boundary is to increase the diversity of solutions towards searching near the boundary, the increased size should be small and quickly converge back into the original boundary. Therefore, the boundary is expanded by 10% of its initial size, and reduce back to the original boundary after the first 10% of the $\mathrm{N}_{\mathrm{gen}}$. This ensures the boundaries are fully explored first, with the subsequent generations to converge to the global optimum.

3.4 Model Hyper-parameter Tuning

The GPR surrogate model hyper-parameters will be re-tuned after each subsequent data acquisition in all algorithms in this thesis, as such the model hyper-parameter tuning procedure will be detailed as follows in Section 3.4. As previously mentioned in Section 2.2.1 the GP surrogate models' hyper-parameters are tuned through the optimisation of Equation 2.10, which consists of a cross-validation cost function to evaluate a hyper-parameter set's fitness to the observed data set and a GA algorithm to efficiently search for a globally optimal set of hyper-parameters.

The cross-validation approach was previously introduced in Section 2.2.1 to evaluate how well each hyper-parameters settings to fit to the observed data set, detailed in Algorithm 3.2. However, in order to search for the optimal configuration of hyper-parameter settings an optimiser is needed to search a variety of hyper-parameter settings, in this thesis a GA algorithm is used as shown in Figure 3.1. The cross-validation approach is implemented as the population evaluation component and the setup of the GA tuning parameters is set according to Table 3.1.

Algorithm 3.2 Gaussian Process Model Tuning: K-Fold Cross-Validation

```
procedure GPR$\left(\lambda, D_t, \mathbf{o}\right)$ do
    Initialise $P_i(\theta) \leftarrow$ Random Draw$(\lambda)$
    for $i = 1, 2, \ldots, N_{Gen}$ do
        Partition $D_t$ into k sets
        for $j = 1, 2, \ldots, k$ do
            $D_{test} \leftarrow D_t \in [x, y]_j$
            $D_{train} \leftarrow D_t \in [x, y]_{-j}$
            $GP\left(m, k|\theta_i\right) \leftarrow$ Train Model$\left(D_{train}, P_G\left(\theta_i\right)\right)$
            $\mu_{test} \leftarrow$ Predict Mean$\left(D_{test}, D_{train}, GP(m, k|\theta_i)\right)$   // See Equation 2.21a
            $\sigma_{test} \leftarrow \sqrt{\text{Predict Var}\left(D_{test}, D_{train}, GP(m, k|\theta_i)\right)}$   // See Equation 2.21b
            $\log p_j \leftarrow$ Pred Log Prob$\left(y_{test}|x_{train}, y_{train}, \theta_i\right)$   // See Equation 2.12
        end for
        $L_{PLP} \leftarrow \sum_{j=1}^{k} \log p_j\left(y_{test}|x_{train}, y_{train}, \theta_i\right)$
    end for
    return $\theta_t \leftarrow \arg\max_{\theta_t} L_{PLP}$
end procedure
```

The final consideration in order to utilise the GA algorithm for hyper-parameter optimisation is how the hyper-parameters are initially sampled. As when the GA algorithm is utilised for acquisition function optimisation the initial population is drawn from a defined feature space, this is not the case for hyper-parameters for GP surrogate models. As when fitting GP surrogate models the main concern is overfitting to the observed data which can lead to poor inference [216].

To prevent the issue of overfitting and poor model prediction a hyper-prior distribution is selected for each hyper-parameter from which the initial GA individuals will be drawn from. Prior distributions can be roughly classified into 3 main distribution types:

1. Informative Prior.

2. Non-Informative Prior.

3. Weakly Informative Prior.

An informative prior is a prior which incorporates a large body of knowledge (expert knowledge, literature, exploratory experimentation, etc.) into its selection, which will have a large impact on the tuning of the model hyper-parameters.

A non-informative prior utilises no information but rather lets the data inform the selection of the hyper-parameters. A compromise prior distribution are weakly informative distributions which are to be used in this work.

A weakly informative prior is ideal as the primary goal is the regularization of the hyper-parameter to constrain the hyper-parameter within a suitable range. By using a weakly informative prior distribution, the hyper-parameter is prevented from overfitting to the data whilst also preventing excessive guidance in the selection of the hyper-parameters.

The SE kernel (See Equation 2.7) has been selected and thus there are two main hyper-parameters to be tuned: Length Scale ($\mathrm{L_d}$) hyper-parameter and output variance (σ_f^2) hyper-parameter. Also as specified in Section 2.2.1 an additional noise hyper-parameter is also included, as outside of ideal conditions some noise will exist within the data such as within manufacturing cases. Therefore, an appropriate weakly informative hyper-prior has been selected for each hyper-parameter:

- Length-Scale:

  Inverse Gamma Distribution

- Output Variance:

  Half-Normal Distribution

- Noise:

  Uniform Distribution

3.4.1 Length Scale Hyper-prior

As previously mentioned in Section 2.2.1, the length scale hyper-parameter determines the smoothness of variation along all or specific dimensions of the GPR model. A variety of factors can influence the selection of a suitable hyper-prior distribution for length scale based upon the kernel type, the processing of the data, and prevention of overfitting and/or excessive smoothing.

Firstly, the kernel type being used in this work is the SE kernel shown in Section 2.2.1, however this can be further split into 2 forms based upon handling of multiple input dimensions. A SE kernel length scale can either be set for each individual dimension of inputs, known as an anisotropic kernel or a single hyper-parameter for all input dimensions, known as an isotropic kernel. In this work, we utilise the anisotropic form of the SE kernel (See Equation 3.16)), which is also referred to as the Squared Exponential (SE) Automatic Relevance Determination (SE-ARD) kernel [63].

$$k_{SE-ARD}\left(x_i, x_j\right) = \sigma_f^2 \exp\left(-\frac{1}{2}\left(x_i - x_j\right)\frac{1}{L_d^2}\left(x_i - x_j\right)\right) \tag{3.16}$$

Where, $\mathbf{x}_i$ is experiment i and $\mathbf{x}_i$ is experiment j, σ_f^2 is the signal variance hyper-parameter, and L_d is a diagonal matrix set of length scale hyper-parameters with one parameter for each input dimension d.

Secondly, as a length scale determines the distance in which a new experiment can be extrapolated from previously observed data [63], the processing of the input data will affect the tuning of the hyper-prior distribution. As previously specified in Section 3.1 the inputs will be standardised using Equation 3.1, after which the inputs will range between $[-2, 2]^d$. Thus, the length scales sampled from the hyper-prior distribution should fall within the range, $[-2, 2]^d$.

Finally, in order to overfit or excessively smoothen an SE kernel, the length scale must be either an incredibly large (excessive smoothing) or an incredibly small (overfitting) value. The prior distribution property of interest for these scenarios are sharp-tailed distributions which place infinitesimal probability at the tails of the distribution. To prevent overfitting a sharp left-tailed distribution is required, whilst to prevent excessive smoothing a sharp right-tailed distribution is required.

Whilst prevention of overfitting promotes a well-tuned GPR model without exact interpolation, excessive smoothing is an undesirable property if implemented excessively. Such that the smoothness of a particular length scale hyper-parameter should be explored reasonably, it should also have a soft limitation on the right-tail of the distribution with a sharp left-tail.

Therefore, when choosing a weakly informative hyper-prior distribution, considering the desired distribution properties discussed previously an Inverse Gamma distribution is the most suitable. An Inverse Gamma distribution is defined by its shape (α_{IG}) and scale (β_{IG}) parameters, as shown in Equation 3.17.

$$f\left(x; \alpha_{IG}, \beta_{IG}\right) = \frac{\beta_{IG}^{\alpha_{IG}}}{\Gamma\left(\alpha_{IG}\right)}\left(\frac{1}{x}\right)^{\alpha_{IG}+1} \exp\left(-\frac{\beta_{IG}}{x}\right) \tag{3.17}$$

where, $f\left(x;\alpha_{IG},\beta_{IG}\right)$ is the inverse gamma's probability density function, α_{IG} is the shape parameter, β_{IG} is the scale parameter and $\Gamma(.)$ is the Gamma function.

To ensure that the length scales sampled from an Inverse Gamma distribution fall within $[-2, 2]^d$, the Inverse Gamma distribution parameters are set to $\alpha_{IG} = 5$ and $\beta_{IG} = 5$, where the corresponding probability density function is represented in Figure 3.6.

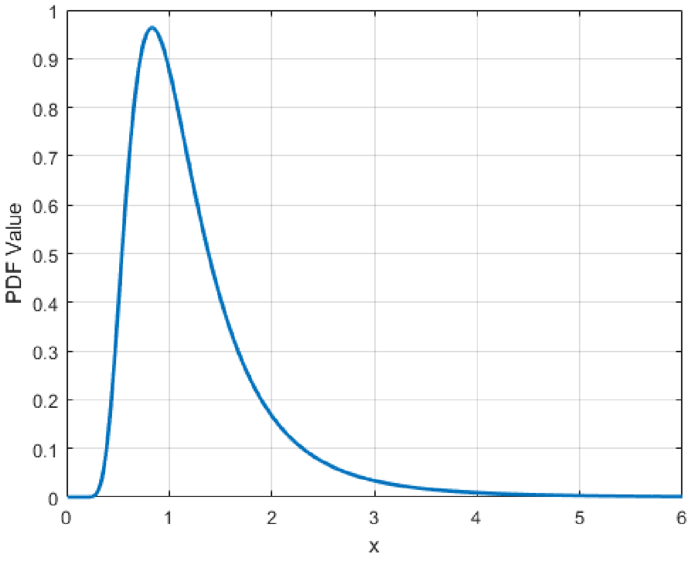


Figure 3.6: Inverse Gamma Probability Density Function with shape factor = 5 and scale factor = 5 .

3.4.2 Output Variance Hyper-prior

As previously mentioned in Section 2.2.1, the output variance hyper-parameter determines the average distance the response function is from its mean, although it is often referred to as a simple scaling factor. As the GPR model is built upon a standardised data set, the range of output values vary between standard normal distribution ($\mu = 0, \sigma = 1$) ranges. Therefore, the range of values sampled from the hyper-prior distribution for the output variance hyper-parameter should be greater than zero but, should not exceed the largest sample from a standard normal distribution.

A Half-Normal distribution fits the requirements for the output variance hyper-parameter perfectly. Whereby the distribution is a standard normal distribution except the for probability of sampling below zero is equal to zero, as defined in Equation 3.18 and illustrated in Figure 3.7.

$$f(x;\sigma) = \frac{\sqrt{2}}{\sigma\sqrt{\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad \text{and} \quad x \geq 0 \tag{3.18}$$

where, $f(x;\sigma)$ is the half-normal's probability density function and σ is the standard deviation.

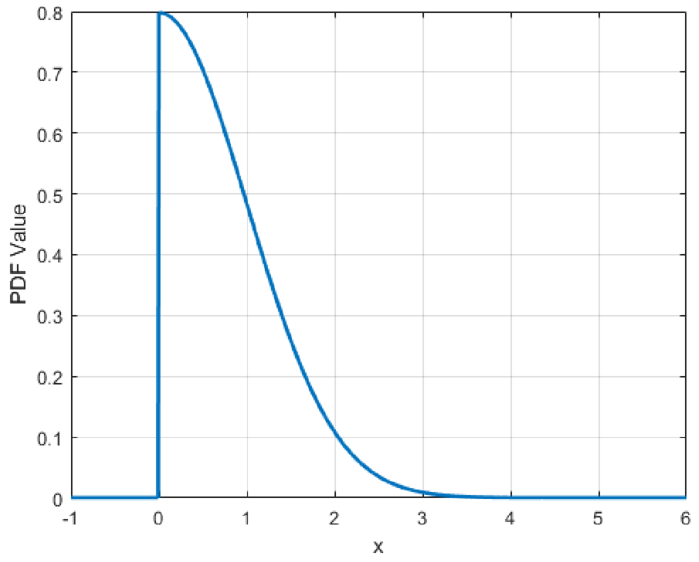


Figure 3.7: Half-Normal Probability Density Function with mean = 0 and standard deviation = 1.

3.4.3 Noise Hyper-prior

As previously shown in Equation 2.10, the noise hyper-parameter is added to models in which noise is present such as manufacturing cases. Although, typically in most simulations, the benchmark problems or case studies are noise-free and thus do not require a noise hyper-parameter. Despite this a small value of noise is typically included within all GPR models to ensure matrix inversion tractability [182]. By adding noise to the kernel, Equation 2.15 to Equation 2.21b are modified into Equation 3.19 to Equation 3.21b.

$$\mathbf{K}_{1:n} = \begin{pmatrix} k(X_1, X_1) & \dots & k(X_1, X_n) \\ \vdots & \ddots & \vdots \\ k(X_n, X_1) & \dots & k(X_n, X_n) \end{pmatrix} + \sigma^2_{\text{noise}} * I \tag{3.19}$$

Which yields a noisy predictive posterior distribution:

$$P\left(f_{n+1}|D_{1:n}, X_{1:n}\right) = \mathcal{N}\left(\mu_n\left(X_{n+1}\right), \sigma_n^2\left(X_{n+1}\right) + \sigma_{noise}^2\right) \tag{3.20}$$

Which can be re-arranged to derive the noisy predictive posterior mean and standard deviation respectively.

$$\mu_t\left(X_{n+1}\right) = k^T\left[K + \sigma_{noise}^2\right]^{-1} f_{1:n} \tag{3.21a}$$

$$\sigma_t^2(X_{n+1}) = k\left(X_{n+1}, X_{n+1}\right) - k^T\left[K + \sigma_{noise}^2\right]^{-1} k \tag{3.21b}$$

Therefore, when choosing a weakly informative hyper-prior distribution for the noise hyper-parameter requires an easily scalable noise hyper-parameter that can be small to ensure inversion tractability for simulation problems or scalable for real world applications.

A continuous uniform distribution with tuneable constraints is the most suitable distribution for the requirements of the noise hyper-parameter, shown in Equation 3.22 and illustrated in Figure 3.8. A continuous uniform distribution has equal probability of selecting values within the constrained limits, of which the limits can be changed to suit the needs to specified problem.

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b, \\ 0 & \text{for } x < a \text{ or } x > b \end{cases} \tag{3.22}$$

where, $f(x)$ is the continuous uniform probability density function, a is the lower bound and b is the upper bound.

Whereby, for a noise-free simulation problem, the noise hyper-parameter limits are set to a $\log_{10}$ scale. This is to transform the range limits to a small but positive range kept reasonably small such as $[10^{-3}, 10^{-6}]$. This prevents the kernel matrix inversion during inference (See Equation 2.21a/ 2.21b) from being non-invertible, whereas this range can be altered for noisy manufacturing cases based upon an expert's knowledge.

3.4.4 Optimiser Adaptations

As described previously the GA framework when tuning the GPR model hyper-parameters is mostly identical to the framework described in Section 3.3, aside from the method of sampling the initial population of hyper-parameters using hyper-priors. Another small modification that is required is to the mutation operator, which operates under the same random mutation mechanism. Rather they are sampled from their respective hyper-prior distribution as op-

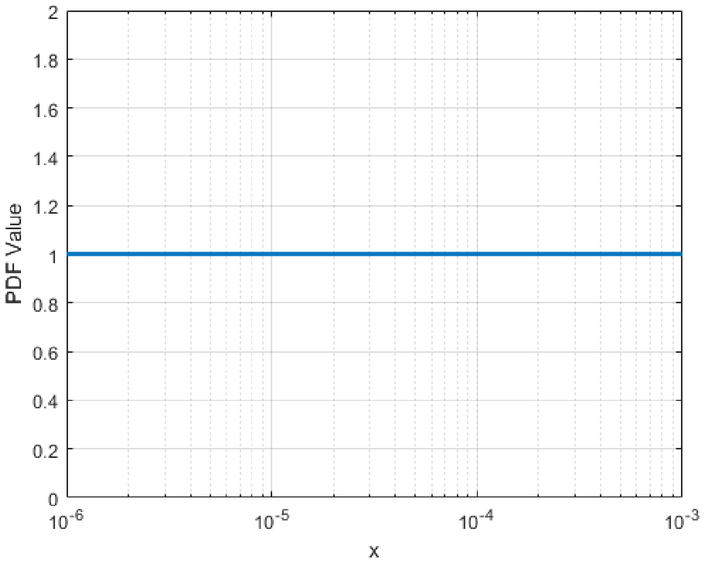


Figure 3.8: Uniform Probability Density Function between 10^{-6} to 10^{-3}.

posed to sampling between the upper and lower limits of the feature space as previously specified.

Chapter 4

Sequential DOE Optimisation

This chapter aims to assess the capabilities of the BO approach for application onto DoE problems with a particular focus on Additive Manufacturing (AM) problems. It also seeks to assess the capabilities and selection mechanisms of the three acquisition function types discussed in Section 2.2.2 to identify which is suitable for extension into a batch and multi-objective batch frameworks in future chapters.

As discussed in Section 2.4 the majority of BO literature applied in a DoE framework on AM in the last five years has primarily been focused on the use of the EI acquisition function [246]. Coupled with the BO literature's primary focus on assessing BO algorithms ability to locate the global optimum, one contribution in this chapter is the investigation and analysis of Improvement [131] [119], Optimistic [208], and Informative [100] [101] [230] acquisition functions for both their capability in building accurate surrogate models and locating the global optimum.

Secondly, the novelty of the framework to be introduced in Algorithm 4.1 lies not in its formulation, but rather the combination of BO components: GP surrogate model, sequential acquisition functions (EI, GP-CB, and MES), and the surrogate-assisted constraint handling optimiser (GA) which have not previously been applied onto a AM DoE manufacturing case study [246].

4.1 Introduction

As previously discussed in Section 2.1.1 when tackling an expensive manufacturing DoE problem, the goal is to improve understanding through a systematic and efficient approach. The approach should seek to gather as much information as possible in as few experiments as possible whilst delivering on the objectives of accurate models and optimal process settings. Within DoE literature the ARSM branch of literature was identified to contain the most suitable methods that could handle the characteristics of interest, in particular BO was chosen to be investigated (see Section 2.4 for review of BO in DoE).

In Section 2.2 an overview of the BO framework was introduced as a potential ARSM approach for expensive manufacturing DoE problems. Although within literature it is most frequently applied as a sequential approach, selecting a single experiment through the optimisation of an acquisition function. Within this study we seek to explore whether this framework is a suitable approach for application onto manufacturing problems. Whilst further extensions onto other problem types such as batch selection or multiple conflicting response criteria are deferred to Chapter 5 and 6 respectively.

4.2 Applicability of BO in DoE

BO was first introduced in Section 2.2 which detailed that the methodology could be classified into three main components:

1. Building a surrogate model (Gaussian Process Regression, Section 2.2.1)
2. Acquisition function choice (Section 2.2.2)
3. Acquisition function optimisation (Section 3.3)

These three core components comprise the main interchangeable elements that will affect the performance of the overall BO framework. Therefore, through the variation of which model, acquisition function, and optimiser is chosen determines the setup of the B-DoE framework and its subsequent performance.

Gaussian Process Regression (GPR) models were compared and contrasted in Section 2.2 as the choice of surrogate model for their flexible modelling, versatility in application, and incorporation of expert knowledge, as well as the thorough investigation throughout the BO literature [195].Whereas for the optimisation component a variety of methods have been implemented throughout literature as discussed in Section 2.2.3. Genetic Algorithm (GA) were selected as the acquisition function global optimiser for its capability in handling multi-modal functions, large input parameter spaces, constraints, and multiple response criteria. As opposed to other gradient-based optimiser's in literature which can require multiple restarts to prevent being trapped in local optima and can only be implemented when gradients can be cheaply evaluated [195].

Subsequently, as two of the three core components have been selected, the primary method to influence performance of the BO framework is the choice of the acquisition function. In order to adequately assess the capabilities of each acquisition function as a B-DoE method for eventual application onto expensive additive manufacturing problems, a series of investigations are required. Firstly, an investigation onto performance of a series of benchmark synthetic optimisation problems of varying characteristics: modalities, dimensions, and separability.

4.3 Simulation Study: Benchmark Problems

Three B-DoE frameworks are to be investigated to evaluate their performance on a series of synthetic benchmark functions to determine their suitability for use in expensive additive manufacturing DoE. Each of the B-DoE algorithms will be run identically as previously described in introduced in Section 2.2 using the GA optimiser introduced in Section 3.3 as shown in Algorithm 4.1. Although, in each instance of the B-DoE algorithm a different acquisition function will be utilised: Expected Improvement (EI), Gaussian Process Confidence Bound (GP-CB), and Max Value Entropy Search (MES) as depicted in Figure 4.1 with blue, red and green lines respectively depicting the differing implementations and black the functions used throughout. Each acquisition function is incorporated into Algorithm 4.1 on line 18 and further details for each acquisition function in Algorithms:

- Expected Improvement (EI) in Algorithm 4.2 [119].
- Gaussian Process Confidence Bound (GP-CB) in Algorithm 4.3 [208].
- Max Value Entropy Search (MES) in Algorithm 4.4 [230].

As illustrated in Algorithm 4.3, only in GP-CB is there an additional parameter that requires definition and selection prior to implementation, the β_t parameter. The β_t parameter is selected based upon the bounds of the GP prior as discussed in Srinivas et al. [208] for which β_t will be set using *Theorem 1*:

$$\beta_t = 2\log\left(\frac{|X|\,t^2\pi^2}{6\delta}\right) \tag{4.1}$$

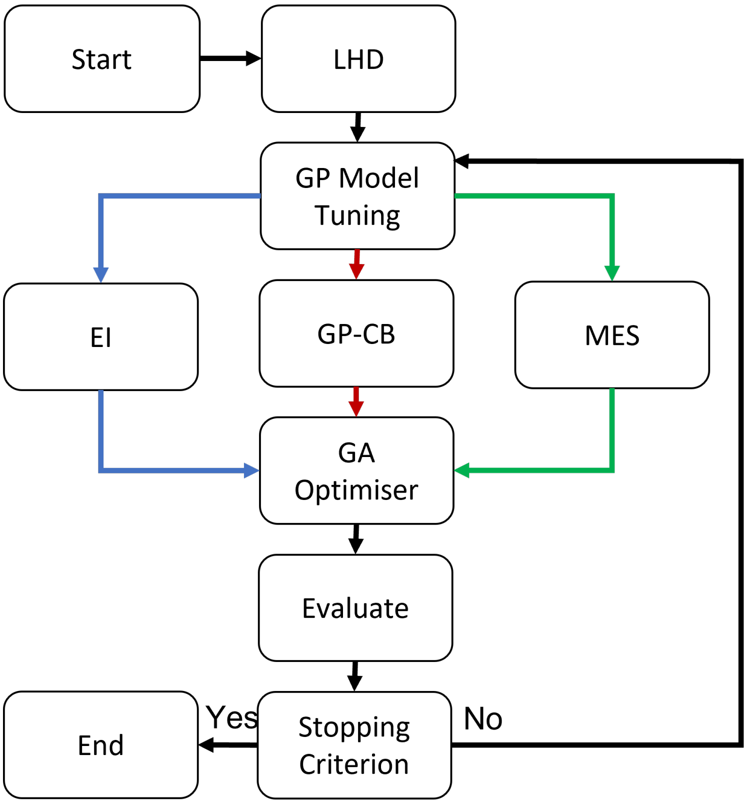


Figure 4.1: High-level flowchart depicting the workflow for the B-DoE shown in Algorithm 4.1, where the core algorithm flow is depicted with a black line and how the three differing acquisition functions being investigated align within the workflow. Expected Improvement (blue), GP Confidence Bound (red) and Max-Value Entropy Search (green).

Where, $\log(\ldots)$ is the natural log of its argument, t is the iteration number, δ is a constant of value 0.1 and $|\mathrm{X}|$ is the number of discretised points. β_t was also further reduced by a factor of 5 from the originally defined value in Equation 4.1 as Srinivas et al. [208] found it to improve the performance of the GP-CB acquisition function.

Accessing the versatility and adaptability of the B-DoE algorithms requires a variety of DoE problem types to ensure good performance in most DoE scenarios. Therefore, the selection a few synthetic benchmark functions with varied optimisation landscapes and characteristics will allow for the assessment of the consistency of the B-DoE algorithms. In this simulation study three synthetic benchmark functions were selected: Branin-Hoo [38], Cosines [6], and Hartman [99].

Algorithm 4.1 B-DoE Algorithm

```
Inputs:
        $n_s$  : Number of LHS samples
        $\lambda$  : Model Hyper-parameter priors
        $\mathbf{o}$  : Vector of optimisation settings
        $T$  : Experimentation budget

 1: $x_t \leftarrow$ Latin Hypercube Sampling$(n_s, d)$
 2: $y_t \leftarrow$ Evaluate$(x_t)$
 3: $D_t \leftarrow \left[x_t, y_t\right]$
 4: $\theta_t \leftarrow$ GPR$\left(\lambda, D_t\right)$                          // See Algorithm 3.2
 5: for $t = 1, 2, \ldots, T$ do
 6:     $x_{t+1} \leftarrow$ B-DoE GA Optimiser$\left(D_t, GP(m, k|\theta_t), \mathbf{o}\right)$     // See line 11
 7:     $y_{t+1} \leftarrow$ Evaluate$(x_{t+1})$
 8:     $D_{t+1} \leftarrow D_t \cup \left[x_{t+1}, y_{t+1}\right]$
 9:     $\theta_{t+1} \leftarrow$ GPR$\left(\lambda, D_{t+1}\right)$                  // See Algorithm 3.2
10: end for

11: procedure B-DoE GA Optimiser$\left(D_t, GP(m, k|\theta_t), \mathbf{o}\right)$ do
12:     Initialise GA Population $\left(P_1 = \left[C_1\right]\right)$
13:     for $G = 1, 2, \ldots, N_{Gen}$ do
14:         $u_{new} \leftarrow u_{AcqFunction}\left(C_G, D_t, GP(m, k|\theta_t), \mathbf{o}\right)$   // See Algorithm 4.2 - 4.4
15:         $P_G \leftarrow$ Update Population $\left[C_G, u_{new}\right]$
16:         if $G = 1$ then
17:             $E \leftarrow \max_{i=1}^{E_{size}} P_G$
18:         else
19:             $E_{New} \leftarrow$ Elite Set Updater$(P_G, E, E_{size})$       // See Algorithm 3.1
20:         end if
21:         $P_G \leftarrow$ Constraint Evaluation$\left(P_G\right)$              // See Section 3.3.5
22:         $Par_G \leftarrow$ Tournament Selection$\left(P_G\right)$             // See Section 3.3.2
23:         $O_G \leftarrow$ Blending Crossover$\left(Par_G\right)$               // See Section 3.3.3
24:         $M_G \leftarrow$ Random Mutation$\left(O_G\right)$                   // See Section 3.3.4
25:         $C_{G+1} \leftarrow O_G \cup M_G$
26:     end for
27:     return $x_{t+1} \leftarrow \arg\max_C u\,(P, E)$
28: end procedure
```

Algorithm 4.2 Acquisition Function: Expected Improvement

```
procedure Expected Improvement$\left(C, D_t, GP(m, k|\theta_t), \mathbf{o}\right)$ do
  if $\mathbf{o} = Max$ then
    $x^+ \leftarrow \max\left(\mathbf{o}, y_t\right)$
  else if $\mathbf{o} = Min$ then
    $x^+ \leftarrow \min\left(\mathbf{o}, y_t\right)$
  end if
  for $i = 1, 2, \ldots, P_{Size}$ do
    $\mu_{t+1}\left(C_i\right) \leftarrow$ Predict Mean$\left(C_i, D_t, GP(m, k|\theta_t)\right)$        // See Equation 2.21a
    $\sigma_{t+1}\left(C_i\right) \leftarrow \sqrt{\text{Predict Var}\left(C_i, D_t, GP(m, k|\theta_t)\right)}$        // See Equation 2.21b
    $Z\left(C_i\right) \leftarrow$ EI-Score$\left(\mathbf{o}, x^+, \mu_{t+1}\left(C_i\right), \sigma_{t+1}\left(C_i\right)\right)$        // See Equation 2.28
    $u_{EI}\left(C_i\right) \leftarrow \left(\sigma_{t+1} * (Z * \Phi(Z))\right) + \left(\sigma_{t+1} * \varphi(Z)\right)$
  end for
  return $u_{EI}$
end procedure
```

Algorithm 4.3 Acquisition Function: Gaussian Process Confidence Bound

```
procedure Guassian Process Confidence Bound$\left(C, D_t, GP(m, k|\theta_t), \mathbf{o}\right)$ do
  $\beta_t \leftarrow$ Trade-off Parameter$(\mathbf{o})$        // See Equation 4.1
  for $i = 1, 2, \ldots, P_{Size}$ do
    $\mu_{t+1}\left(C_i\right) \leftarrow$ Predict Mean$\left(C_i, D_t, GP(m, k|\theta_t)\right)$        // See Equation 2.21a
    $\sigma_{t+1}\left(C_i\right) \leftarrow \sqrt{\text{Predict Var}\left(C_i, D_t, GP(m, k|\theta_t)\right)}$        // See Equation 2.21b
    if $\mathbf{o} = Max$ then
      $u_{GP-CB}\left(C_i\right) \leftarrow \mu_{t+1}\left(C_i\right) + \sqrt{\beta_t} * \sigma_{t+1}\left(C_i\right)$
    else if $\mathbf{o} = Min$ then
      $u_{GP-CB}\left(C_i\right) \leftarrow \mu_{t+1}\left(C_i\right) - \sqrt{\beta_t} * \sigma_{t+1}\left(C_i\right)$
    end if
  end for
  return $u_{GP-CB}$
end procedure
```

Algorithm 4.4 Acquisition Function: Max-Value Entropy Search

```
procedure Max-Value Entropy Search$\left(C, D_t, GP(m, k|\theta_t), \mathbf{o}\right)$ do
    for $i = 1, 2, \ldots, R$ do
        Sample $\widetilde{f_i}(x) \sim \left(m, k|D_t\right)$
        $y_i^* \leftarrow \arg\max_x \widetilde{f_i}(x)$
    end for
    for $j = 1, 2, \ldots, P_{Size}$ do
        $\mu_{t+1}\left(C_j\right) \leftarrow$ Predict Mean$\left(C_j, D_t, GP(m, k|\theta_t)\right)$        // See Equation 2.21a
        $\sigma_{t+1}\left(C_j\right) \leftarrow \sqrt{\text{Predict Var}\left(C_j, D_t, GP(m, k|\theta_t)\right)}$        // See Equation 2.21b
        for $k = 1, 2, \ldots, R$ do
            $\gamma_k\left(C_j\right) \leftarrow \gamma\text{-Score}\left(\mathbf{o}, y_k^*, \mu_{t+1}\left(C_j\right), \sigma_{t+1}\left(C_j\right)\right)$        // See Equation 2.36
            $u_{MES,k}(C_j) \leftarrow \left[\frac{\gamma_k(C_j)\varphi(\gamma_k(C_j))}{2*\Phi(\gamma_k(C_j))} - \log\Phi\left(\gamma_k\left(C_j\right)\right)\right]$
        end for
        $u_{MES}(C_j) \leftarrow \frac{1}{R}\sum_{k=1}^{R} u_{MES,k}(C_j)$
    end for
    return $u_{MES}$
end procedure
```

It is of great importance when assessing the performance of an algorithm that the results are analysed on a diverse collection of synthetic benchmark functions, as no single synthetic benchmark function will universally represent each characteristic possible.

- Modality

  The modality of a synthetic benchmark problem corresponds to the number of local optima within the function landscape [111]. Whereby if a synthetic benchmark function is multimodal there exist many local optima, for which a gradient-based optimisation approach may get stuck within and thus tests the B-DoE capabilities of exploring the search space.

- Separability

  The separability of a synthetic benchmark problem is defined as a measure of the problem difficulty to optimise. Wherein a synthetic benchmark function which is separable consists of its input variables being independently applied. In contrast to a non-separable synthetic benchmark function requiring the use of multiple input variables suggesting inter-relation amongst each variable [111].

- Dimensionality

  The dimensionality of a synthetic benchmark problem rather simply refers to the number of dimensions involved, whereby as the dimension increases the search space increases exponentially [111] and thus the optimisation task increases in difficulty.

4.3.1 Branin-Hoo Test Function

The Branin-Hoo function is a minimisation synthetic benchmark function which has three global minima. Its function characteristics are: Multi-modal, Non-separable, and 2-Dimensional, shown in Equation 4.2 [38]:

$$f(\mathbf{x}) = a\left(x_2 - bx_1^2 + cx_1 - r\right)^2 + s(1-t)\cos\left(x_1\right) + s \tag{4.2}$$

Where, a, b, c, r, s, and t are constants with the following recommended values:

$$a = 1, \quad b = \frac{5.1}{4\pi^2}, \quad c = \frac{5}{\pi}, \quad r = 6, \quad s = 10, \quad \text{and} \quad t = \frac{1}{8\pi}$$

The Branin-Hoo search space is located in the domain $x_1 = [-5, 10]$ and $x_2 = [0, 15]$. The value and locations of its global optima are defined as below:

$$f\left(\mathbf{x}^*\right) = 0.397887 \qquad \mathbf{x}^* = (-\pi, 12.275), \quad (\pi, 2.275), \quad \text{and} \quad (9.42478, 2.475)$$

4.3.2 Mixture of Cosines Test Function

The Cosines function is a maximisation synthetic benchmark function which has a single global maximum. Its function characteristics are: Multi-modal, Separable and 2-Dimensional, shown in Equation 4.3 [6]:

$$f(\mathbf{x}) = 1 - \left(u^2 + v^2 - 0.3\cos(3\pi u) - 0.3\cos(3\pi v) + 0.7\right) \tag{4.3}$$

Where,

$$u = 1.6x_1 - 0.5 \quad \text{and} \quad v = 1.6x_2 - 0.5$$

The Mixture of Cosines search space is located in the domain $x_i = [0, 1]^{d=2}$. The value and locations of its global optima are defined as below:

$$f\left(\mathbf{x}^*\right) = 0.9 \qquad \mathbf{x}^* = (0.3120, 0.3120)$$

4.3.3 Hartman-4 Test Function

The Hartmann function is a minimisation which has a single global minimum. Its function characteristics are: Uni-modal Non-separable, and 4-Dimensional, shown in Equation 4.4 [99]:

$$f(\mathbf{x}) = \frac{1}{0.839}\left[1.1 - \sum_{i=1}^{4} a_i \exp\left(-\sum_{j=1}^{4} A_{ij}\left(x_j - P_{ij}\right)^2\right)\right] \tag{4.4}$$

Where,

$$a = (1.0, 1.2, 3.0, 3.2)^T$$

$$A = \begin{pmatrix} 10.00 & 3.00 & 17.00 & 3.50 & 1.70 & 8.00 \\ 0.05 & 10.00 & 17.00 & 0.10 & 8.00 & 14.00 \\ 3.00 & 3.50 & 1.70 & 10.00 & 17.00 & 8.00 \\ 17.00 & 8.00 & 0.05 & 10.00 & 0.10 & 14.00 \end{pmatrix}$$

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

The Mixture of Cosines search space is located in the domain $x_i = [0, 1]^{d=4}$. The value and locations of its global optima are defined as below:

$$f\left(\mathbf{x}^*\right) = -3.1345 \qquad \mathbf{x}^* = (0.1874, 0.1941, 0.5579, 0.2648)$$

4.3.4 Benchmark Selection Rationale

When selecting these three synthetic benchmark problems the rationale was to select problems of varying degrees of difficulty but to also consider their comparability to AM DoE problems. As the goal of this study is to fully assess the capabilities of the B-DoE algorithm and their three acquisition functions in locating the optimum and fully exploring the design space in as few experiments as possible.

Typically in Additive Manufacturing (AM) DoE studies as illustrated in Ilzarbe et al. [109] and Chia et al. [47] the number of factors considered on average was 2-6. Therefore for the benchmark functions were chosen to incorporate a comparable range of input variables with: Branin-Hoo (2-D), Cosines (2-D) and Hartman (4-D). Initially another 6-D benchmark function was being considered for implementation to obtain the full range of typical input variables evaluated. However, due to the impact of COVID-19 the scale of investigations conducted during the thesis had to be scaled back. This was coupled with the desire to place more investigative importance on the performance of the manufacturing case studies in Sections 4.4 and 5.4 resulting the removal of the 6-D benchmark investigation.

Of the benchmarks considered the number of factors seem to match those performed in literature [109] [47], but in actuality maybe too few factors considered. As an area of increasing literature interest in the AM field is high dimensional optimisation in which a large number of factors are considered. Its likely that the low number of average factors considered in literature for application of DoE on AM is due to preliminary fractional factorial and Taguchi screening designs to select a lower number of more important factors to perform the analysis upon. Therefore in order to truly represent the AM DoE problems more complex benchmarks in: Dimensionality as well as complexity are required. For further discussion and reflection of research decisions, limitations and impacts refer to Section 7.2.

4.3.5 Experiment Software

Algorithm 4.1 and the majority of functions have been self-coded in MATLAB 2017b in Windows 10 using their respective literature sources unless otherwise stated. The code was developed using MATLAB 2017b's parallel computing toolbox to parallelise the Algorithm 4.1 to be run with multiple repeats of implementations in parallel on ShARC in a private High Performance Computing (HPC) cluster for the Automatic Control and Systems Engineering department at the University of Sheffield. The private cluster has access to 2 worker nodes each with access to 28 cores and 384 GB of RAM of which 20 cores and 20 GB were used to run Algorithm 4.1 according to the details specified in Table 4.1.

The following functions were taken from existing libraries in public repositories rather than being self-coded: Branin-Hoo test function [209], Mixture of Cosines test function [209], Hartman-4 test function [209], and MES function shown in Algorithm 4.4 [230].

4.3.6 Experiment Details

The experiment conditions for the benchmark simulation studies will be set according to the values specified in Table 4.1. Also included within Table 4.1 are the tuning parameter settings for the B-DoE algorithms to be assessed.

Table 4.1: Benchmarking experiments details and settings for assessment as well as B-DoE settings.

Benchmark Study Settings		
Parameter	Symbol	Value
Experimentation Budget	t	50
Number of Repeats		50
Number of Latin Hypercube Samples	n_s	5
B-DoE Settings		
Parameter	Symbol	Value
GP-CB β_t Tuning Parameter	δ	0.1
MES Number of Monte Carlo Estimates	ω	100
Length-Scale Prior	λ_{LS}	$\text{InvGamma}(5, 5)$
Output-Variance Prior	λ_{OV}	$\text{HalfNorm}(0, 1)$
Noise Prior	λ_N	$\text{Uniform}\left(10^{-6}, 10^{-3}\right)$

In Table 4.1, an experimentation budget of 50 experiments was chosen as the purpose of this study is to evaluate the B-DoE framework's different acquisition functions capabilities in both exploring and exploiting the design space. Therefore, enough experiments were required to ensure that the surrogate models fit would converge sufficiently as well as potentially locate the global optimum. Although, an excessive amount couldn't be chosen to prevent too many experiments increasing the computational time thus minimising the amount of data to be analysed. Therefore, a careful balance of sufficient budget was required to ensure enough data to examine the varying characteristics of the experiment selection mechanisms for analysis in Section 4.6 whilst minimising computational cost.

Similarly, the number of repeats were set to 50 to ensure that if good performance is observed it is not due to a degree of randomness within the algorithm but rather by design. However, the number of repeats were not set too high as to prevent excessive time costs with implementing the study. As B-DoE is aimed at the implementation on expensive to evaluate optimisation problems in this case AM DoE [40], the number of initial experimental data was also set to be small at roughly 1/10$^{\text{th}}$ the experiment budget.

For Algorithm 4.1 acquisition function parameters: GP-CB [208] and MES [230] were set according to the suggestions in their original papers. Whilst the hyper-prior's for the GP surrogate models were investigated and chosen according to the discussion in Section 3.4.

4.3.7 Results and Discussion

An exploratory assessment of the three most popular B-DoE algorithm acquisition functions is conducted within, assessing their capabilities in terms of both exploration (model error) and exploitation (regret) on a series of benchmark problems of varying characteristics: Modality, Separability and Dimensionality.

The results are seperated into the different benchmark optimisation problems they have been assessed upon, Figures 4.2 and 4.3 are for the Branin-Hoo test function [38]. Figures 4.4 and 4.5 are for the Mixture of Cosines test function [6]. Figures 4.6 and 4.7 are for the Hartmann test function [99].

Benchmark Results: Branin-Hoo (2D)

As discussed in Section 4.3.1, Branin-Hoo is a multi-modal 2-D minimisation problem with three global optima. Of the three benchmark functions considered in this study Branin-Hoo is the easiest to optimise due to having a smooth response surface with a small number of controllable variables. This would allow for the GPR to fit a surrogate model to the underlying process more easily than the highly multi-modal Cosines or the Hartmann-4 with four controllable inputs.

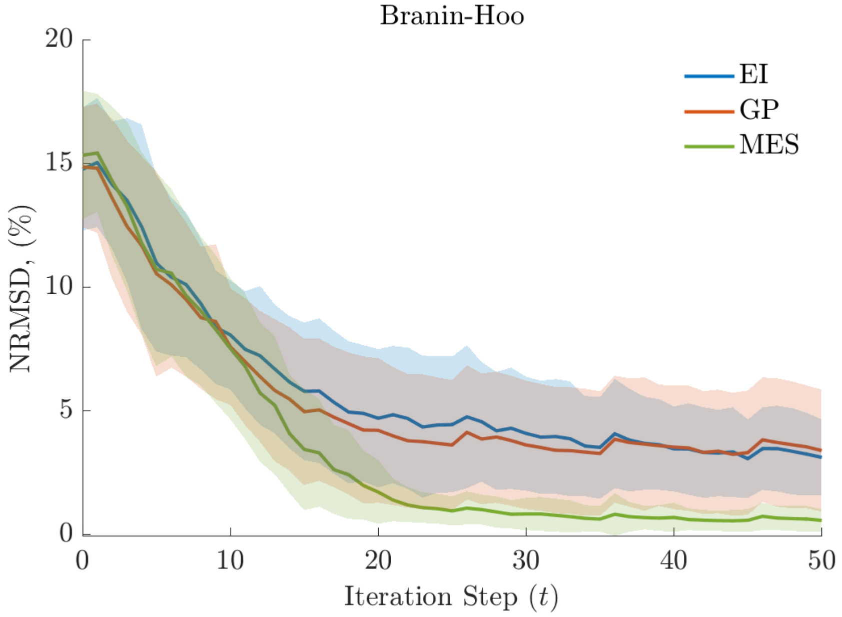


Figure 4.2: Comparison of NRMSD performance metric between three B-DoE algorithm acquisition functions: EI, GP-CB, and MES on the Branin-Hoo benchmark function. The B-DoE algorithm was run according to the settings in Table 4.1 and plotted by taking the mean of the 50 repeats for the NRMSD. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the NRMSD.

Figure 4.2 illustrates the change in NRMSD with each iteration step (t) for each acquisition function as the B-DoE algorithm optimises the Branin-Hoo benchmark function. The results suggest that the MES acquisition function has the best performance achieving a NRSMD of 1 % by the 25th experiment performed in contrast to an NRMSD of 4.5 % for EI and NRMSD of 3.5 % for GP-CB. The 2nd best performance in regard to NRMSD on the Branin-Hoo is less clear, as initially GP-CB NRMSD improves at a faster rate in comparison to EI until the 40th experiment in which their NRMSD's are approximately the same.

Although the MES acquisition function outperforms the alternates in the first 25 experiments for NRSMD, all three acquisition functions plateau reaching an NRMSD $<$ 3.5 % prior to reaching the evaluation budget of 50 experiments which would indicate relatively accurate surrogate models of the underlying process.

Figure 4.3 shows the change in Regret for each acquisition function as

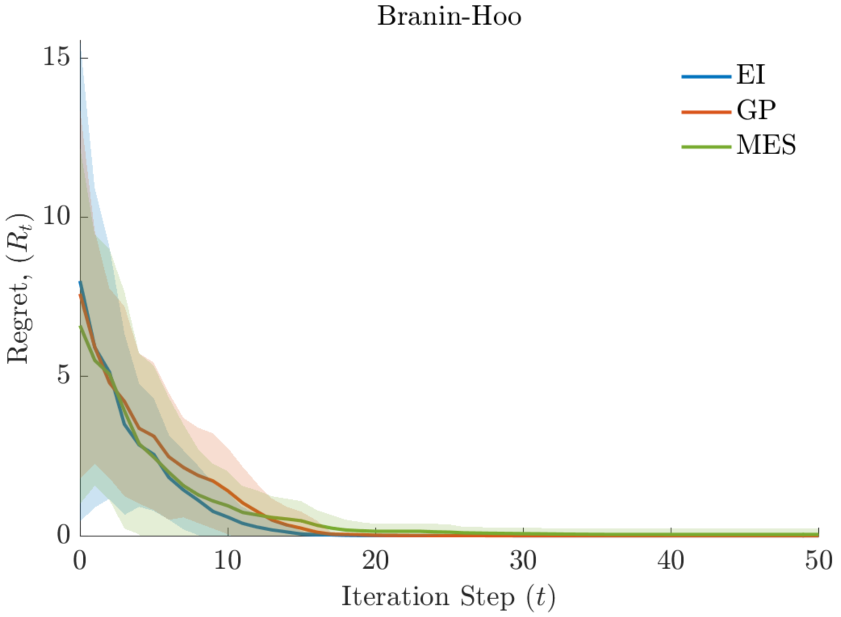


Figure 4.3: Comparison of Regret performance metric between three B-DoE algorithm acquisition functions: EI, GP-CB, and MES on the Branin-Hoo benchmark function. The B-DoE algorithm was run according to the settings in Table 4.1 and plotted by taking the mean of the 50 repeats for the Regret. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the Regret.

the B-DoE algorithm optimises the Branin-Hoo benchmark function. It can be seen that the all three acquisition functions perform reasonably well in determining the process parameter settings relatively close ($R_t < 0.1$) to the global optimum within 30 experiments. However, both EI and GP-CB achieve a Regret of ($R_t < 0.1$) in significantly fewer experiments, 16 and 18 respectively in comparison to 30 for MES.

Benchmark Results: Mixture of Cosines (2D)

As discussed in Section 4.3.2, Mixture of Cosines is a multi-modal 2-D maximisation problem with one global optima. Among the benchmark functions used for assessment, its level of optimisation difficulty ranks in-between the other two benchmark functions due to its many local optima.

In Figure 4.4 it can be seen that all three acquisition functions perform similarly for the first 20 experiments after which there is some variability in modelling performance. Again much alike in Figure 4.2 the MES outperform the alternative acquisition functions significantly reaching an NRMSD of 1% by the end of the evaluation budget. Whilst EI also generated a relatively accurate

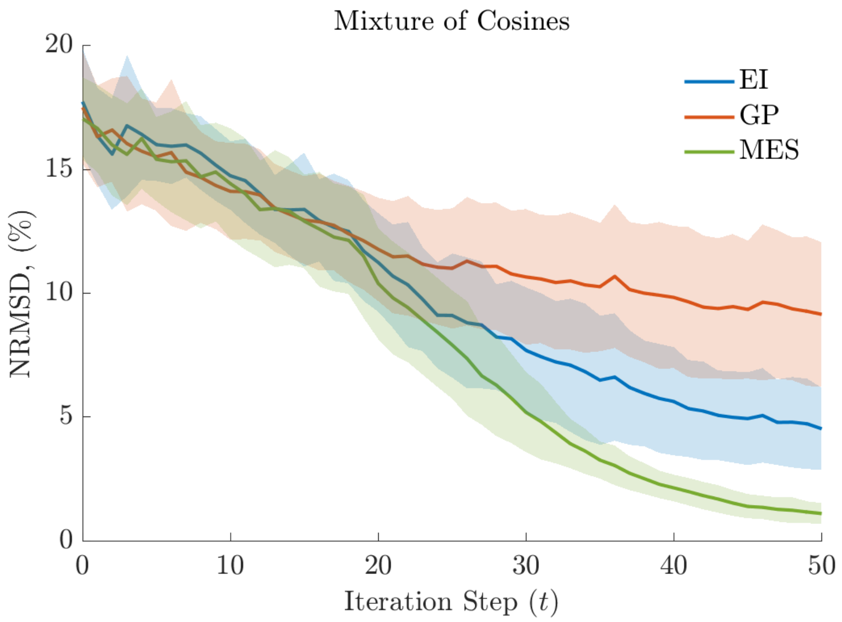


Figure 4.4: Comparison of NRMSD performance metric between three B-DoE algorithm acquisition functions: EI, GP-CB, and MES on the Mixture of Cosines benchmark function. The B-DoE algorithm was run according to the settings in Table 4.1 and plotted by taking the mean of the 50 repeats for the NRMSD. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the NRMSD.

GP surrogate model reasonably well within the evaluation budget achieving the 2nd best NRMSD of 4.5%, with GP-CB GP surrogate model having the worst NRMSD of 9%.

In Figure 4.5 the performance of all three acquisition functions in locating the global optimum on the Mixture of Cosines benchmark problem is mostly identical. Whereby, all three acquisition functions achieve a Regret of $R_{30} < 0.025$ by the 30th experiment, which is relatively close to the global optimum considering a starting Regret of $R_0 = 0.5$. Whilst it can be seen that EI approaches the global optimum at a slightly faster rate of convergence and GP-CB converges onto the optimum 10 experiments after EI and MES the performance difference is marginal.

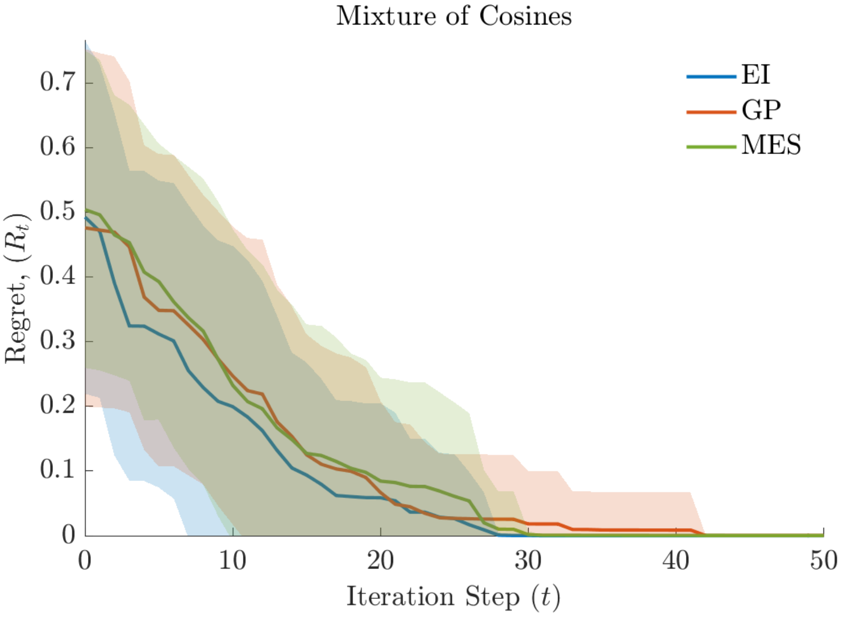


Figure 4.5: Comparison of Regret performance metric between three B-DoE algorithm acquisition functions: EI, GP-CB, and MES on the Mixture of Cosines benchmark function. The B-DoE algorithm was run according to the settings in Table 4.1 and plotted by taking the mean of the 50 repeats for the Regret. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the Regret.

Benchmark Results: Hartmann (4D)

As discussed in Section 4.3.3, Hartmann is a multi-modal 4-D minimisation problem with one global optima. Among the benchmark functions used for assessment, its level of optimisation difficulty ranks as the hardest when compared to the other two benchmark functions due to its increased dimensionality and with a multi-modal characteristic.

Figure 4.6 demonstrates decreased performance of all three acquisition functions in comparison to those on Figure 4.2/ 4.4 most likely due to increased optimisation difficulty on a higher dimension benchmark function. Regardless of the acquisition function applied, the lowest NRMSD achieved after fully utilising all of the evaluation budget is a NRMSD of 10 % using MES. For the initial 20 experiments selected the NRMSD determined at each iteration step (t) between all three acquisition functions remained relatively constant before variations began to occur between each of the acquisition functions. Again the MES achieves the best NRSMD of 10 % followed by EI with an NRSMD of 11.3 % and GP-CB with an NRSMD of 13.7 %.

A dissimilar pattern emerges in the Hartmann benchmark problem when

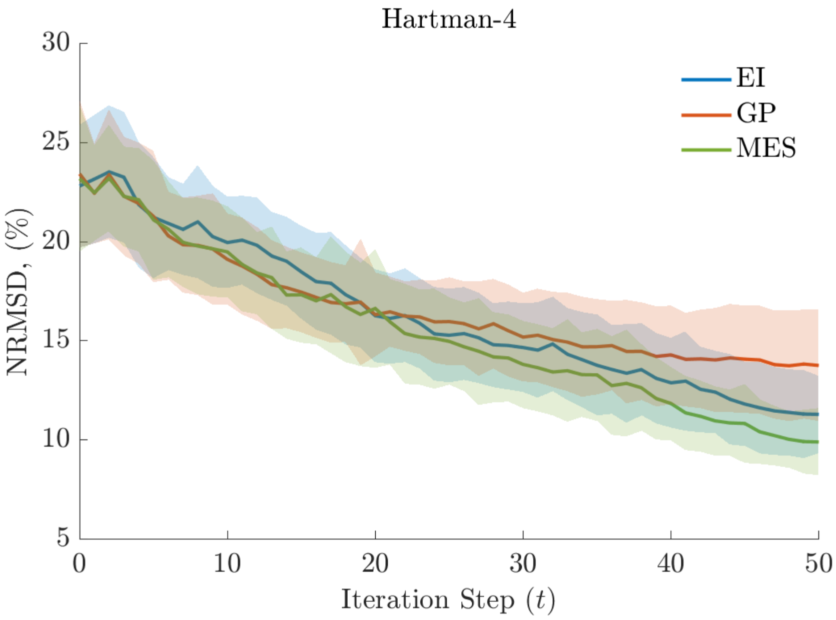


Figure 4.6: Comparison of NRMSD performance metric between three B-DoE algorithm acquisition functions: EI, GP-CB, and MES on the Hartman benchmark function. The B-DoE algorithm was run according to the settings in Table 4.1 and plotted by taking the mean of the 50 repeats for the NRMSD. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the NRMSD.

assessing the Regret as shown in Figure 4.7, possibly due to the increased problem complexity when dealing with a greater number of input variables. Whilst EI remains the best performing acquisition function approaching the global optimum $R_{25} < 0.1$ on average within 25 experiments, GP-CB achieves the same performance by the 30th experiment on average. However, MES seems to struggle in locating the global optimum after utilising the full evaluation budget to find $R_{50} = 0.25$ in contrast to the alternate approaches which require significantly less experiments.

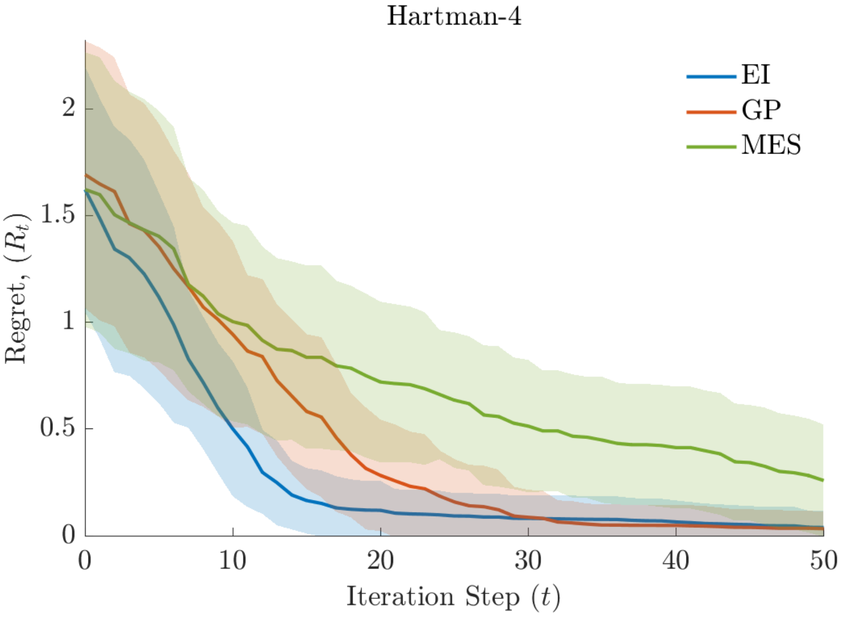


Figure 4.7: Comparison of Regret performance metric between three B-DoE algorithm acquisition functions: EI, GP-CB, and MES on the Hartman benchmark function. The B-DoE algorithm was run according to the settings in Table 4.1 and plotted by taking the mean of the 50 repeats for the Regret. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the Regret.

4.4 Manufacturing Case Study: Selective Laser Melting

4.4.1 Introduction

Thus far the B-DoE algorithms have had their performance assessed upon benchmark functions in Section 4.3 to evaluate their suitability for expensive DoE problems. In this section we seek to expand the assessment onto an additive manufacturing DoE problem. This case study was set to determine if a B-DoE algorithm could select experiments in a more efficient scheme than previously chosen using a traditional DoE method, the Factorial Design.

This case study investigates the underlying process mechanisms of defect formation in a Selective Laser Melting (SLM) to develop a model of the process, as well as find an optimal operation setting for the four process parameters. The process parameters to be assessed are Laser Power (W), Spot Size (mm), Hatch Spacing (mm), and Exposure Time (s).

4.4.2 Background

A SLM process is a metal additive manufacturing method which utilises a high-power density laser to melt and fuse metal powders together, see Figure 4.8. The metal powder is spread in thin layers onto a processing surface, after which a laser following a CAD profile melts the powder to fuse the powder into a cohesive object. This process is sequentially repeated to add layer by layer of metal powder to the surface of previously melted powder to generate a 3-D printed metal object of desired shape.

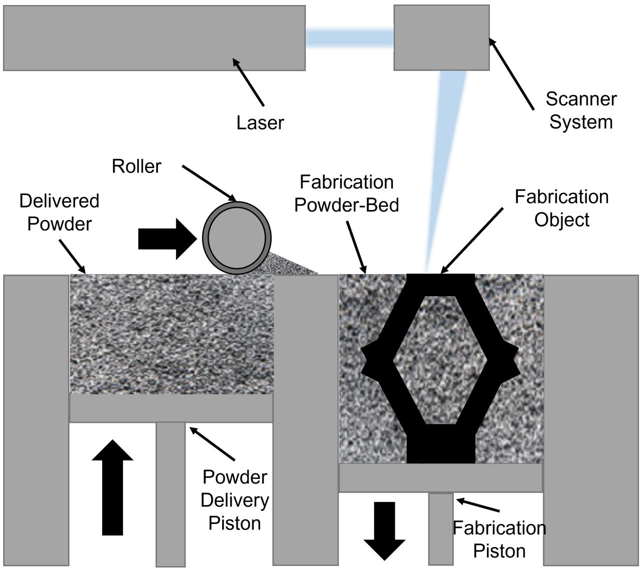


Figure 4.8: Illustrated representation of the manufacturing process of a Selective Laser Melter.

Despite the potential manufacturing benefits of a 3-D metal manufacturing process it is not without disadvantages. The melting process can lead to the formation of various microstructural defects within multiple layers of the 3-D designed object, such as: Lack of fusion (or cracking), and Thermal Strain.

Lack of fusion is a defect formed via the entrapment of gas within the solid material layers rather than the solidification of a cohesive structure throughout. As this defect is formed through the inability of uniform melting across the processing surface, it is most likely formed through insufficient Laser Power or excessively fast nozzle tracking speeds. On the other hand, Thermal Strain leading to cracking is caused by the rapid heating of the metal powders into the rapid cooling of the 3-D metal object. This defect forms cracks via the rapid exchange of heat within a melted material and so is most likely affected by the Laser Power and Exposure Time.

4.4.3 Aims and Objectives

The manufacturer originally implemented a traditional DoE scheme (Factorial Design) to select a set of 67 experiments to perform on an expensive additive manufacturing process. The aim of the study was to identify the optimal process parameter settings to minimise the formation of defects in a SLM process. However, given that the experiments had already been performed, the study involved retrospective optimisation in determining the optimal sequence of experimental settings amongst those carried out. For our case study the objectives are to:

1. Using the data set of 67 experiments, assess if it were possible to select experiments in a more efficient DoE scheme to gain more information in fewer experimental runs.
2. Evaluate the performance of B-DoE algorithms on an additive manufacturing DoE problem.
3. Compare and contrast the acquisition function performance on an additive manufacturing DoE problem.

4.4.4 Methodology

Firstly, as new experiments could not be performed due to the 67 initial experiments run using a factorial design, the initial GP surrogate models training data would have to be from the 67 experiment data set. In order to fairly compare the order of sequential experiment selection from the previously run experiments the training data should be taken from the first set of experiments performed. In the factorial design the experiments were run as part of sets of 7-9 experiments and the first set consisted of 7 experiments which would be partitioned to become the training data set.

$$D_{\text{train}} = [x, y]_{i=1}^{7} \quad \text{and} \quad D_{\text{remainder}} = [x, y]_{i=8}^{67}$$

After the removal of the 7-9 experiment training set, the remaining 58-60 experiments are set aside to become the pseudo design space. Following on from the partitioning of the data sets, the data is pre-processed using standardisation with Equation 3.1. Once the data has been partitioned and processed, a continuous GP surrogate model is fit to the discrete data. Whilst the data available in this study is discrete due to new evaluations being unable to be performed, the controllable inputs and the response variables are both continuous. Using Algorithm 3.2 the GP surrogate models hyper-parameters are fit to the training data set as detailed in Section 2.2.1 and Section 3.4.

As the design space of interest in this study are a set of discrete experiment, Algorithm 4.1 is not implemented directly in which new experiment locations are evaluated for their acquisition value but rather the already observed discrete experiments are enumerated through. These discrete experiment have their acquisition value determined and then optimised to select the next optimal choice for the discrete list of experiments. After each experiment is selected from the remaining data set, they are subsequently removed. For the replication of the experiments selected using the factorial design the remaining data set list is in the original selection order and thus the factorial design experiments are selected from the top of the remaining experiment list in order.

$$D_{t+1} = D_t \cup [x, y]_i \quad \text{and} \quad D_{remainder} = D_{remainder} \setminus [x, y]_i$$

Whilst the purpose of this case study is to verify the suitability and performance of B-DoE algorithms which are sequential in nature, the SLM case study has multiple output objectives. Therefore, in order for B-DoE to be applied onto this case the Multi-Objective Problems (MOP) requires conversion to a Single-Objective Problems (SOP). In order to achieve this the modelling, acquisition assessment, and performance metrics for Lack of Fusion and Thermal Strain will be calculated for each of the separate objectives.

In order to convert the MOP into a SOP, a variety of scalarisation approaches can be used as previously discussed in Section 2.3.2. Each of the scalarisation approaches discussed had varying properties each with their own advantages and disadvantages to their uses. For example the Weighted Sum approach described in Equation 2.45 performs well on convex Pareto Front (PF) geometries but poorly on non-convex PF geometries [58]. Whilst in comparison the Weighted Tchebycheff described in Equation 2.47 outperforms Weighted Sum on non-convex Pareto Front (PF) geometries however, the search capabilities scale poorly with increasing number of output objectives, whilst Weighted Sum does not [114] [228].

Therefore, careful consideration must be taken when selecting the scalarisation approach although the motivation for this case study is not to perform Multi-Objective Optimisation (MOO) in order to minimise the LOF and TS but rather investigate the use of B-DoE on a AM problem. For this study the

LOF and TS will be aggregated into a SOP using the weighted sum approach with an equal weighting of $w_i = 0.5$, as there currently exists no preference to either output objective.

4.4.5 Metrics

As the manufacturing search space is discrete with no closed-form function for its output, the assessment of its modelling error cannot be performed as demonstrated in Section 3.2.2. Thus, prediction of the outputs will be performed over all experiments in the remaining data set that have not been currently selected within the iteration of the B-DoE. Therefore, as each experiment is removed, the testing set for model error will gradually decrease. Thus, the model error will be compared between the B-DoE selection scheme and the traditional DoE scheme at each iteration, to evaluate the rate of model improvement.

In the first instance when the traditional DoE was used a surrogate model was not built and so no prediction values can be obtained for comparison. To alleviate this, the initial GP surrogate model will be set as a baseline GP model for both schemes. It will be updated with each experiment selected according to the order of the traditional DoE scheme to produce a model error profile for comparison with B-DoE schemes.

On the other hand, as the Regret is comparing the current best output with the global optima as shown in Equation 3.3, it is not affected by differences between a continuous or discrete search space.

4.4.6 Experiment Software

Similarly to Section 4.3.5, the set-up of algorithms and functions used were in-line with the benchmark assessments but re-coded for use on the AM case study specifics as discussed in Section 4.4. Algorithm 4.1 and the majority of functions have been self-coded in MATLAB 2017b in Windows 10 using their respective literature sources unless otherwise stated. The code was run on a personal PC with the following specifications: AMD Ryzen 5 1600 6 core processor with 12 threads and 24 GB of RAM of which all were used to run Algorithm 4.1 according to the details specified in Table 4.1. In this study no existing libraries or repositories were used.

4.4.7 Results and Discussion

Figures 4.9; and 4.10 illustrate the model prediction error results, whilst Figures 4.11; and 4.12 illustrate the regret results which depict the optimisation performance. Each figure corresponds to one of the output objectives with each acquisition function used for comparison against the DoE selection scheme used originally by the researchers, Factorial Design.

Comparison of DoE Methods: Model Error

As the data set is a limited discrete set an external testing set could not be provided to calculate the predicted model error, NRMSD. Instead the remaining data set at each iteration step (t) was used as the testing set and thus as the DoE proceeded with each experiment removed from the available data set it was also removed from the testing set. Consequently, with each experiment removed from the testing set, the NRMSD would begin to converge towards zero, whereas if a large testing were available the prediction error could be evaluated across the region of interest equally in which the NRSMD could both increase or decrease depending upon if the model were well-tuned or poorly fitted. Therefore, the key characteristic to analyse is the rate of convergence of the NRMSD as a large testing set is unavailable.

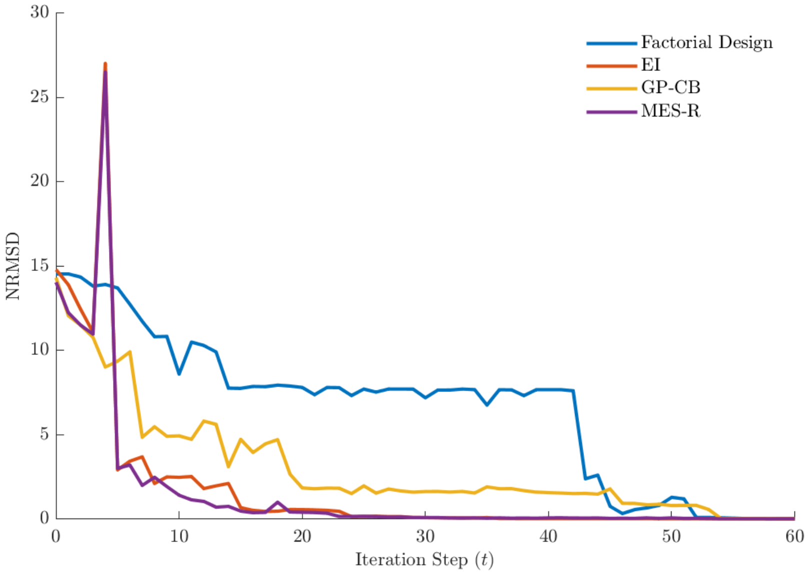


Figure 4.9: Comparison of model error performance for Thermal Strain parameter using B-DoE algorithms from the discrete set of 67 experiments in the SLM manufacturing case against the original scheme: Factorial design

Firstly, in Figure 4.9 each of the B-DoE algorithm acquisition functions outperform the Factorial design in the rate of convergence for the Thermal Strain parameter. Wherein all three acquisition functions achieve a NRMSD < 2 within the first 20 experiments selected, whilst the Factorial design scheme requires 45 experiments to reach the same NRMSD. Although at the 4th iteration step (t) for both EI and MES the NRMSD spikes, this is likely due to the re-tuning of the GP surrogate model hyper-parameters. It could also be due to both methods selecting the experiment which locates the global optimum for the Lack of Fusion output as shown in Figure 4.12. After these initial spikes in NRMSD both EI and MES begin to reduce at a much faster rate in comparison to GP-CB but, all three B-DoE algorithm acquisition functions outperform the Factorial design.

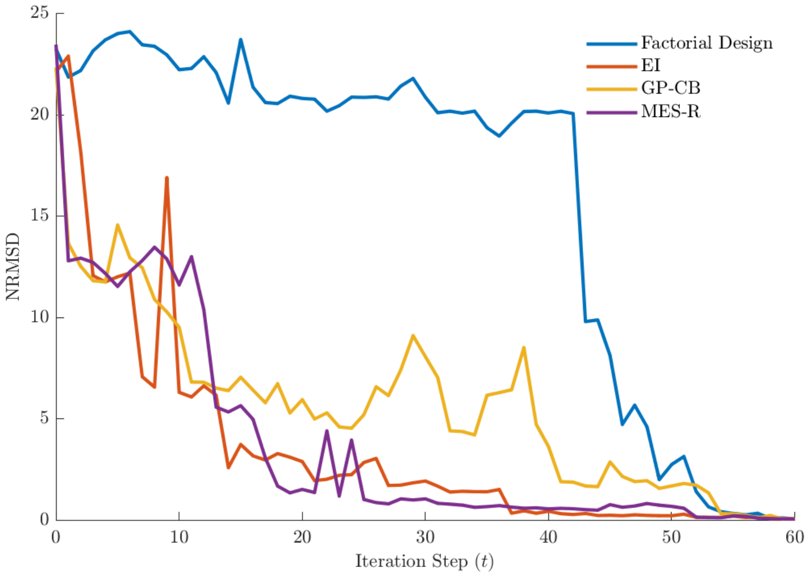


Figure 4.10: Comparison of model error performance for Lack of Fusion parameter using B-DoE algorithms from the discrete set of 67 experiments in the SLM manufacturing case against the original scheme: Factorial design

Secondly, in Figure 4.10 again each of the B-DoE algorithm acquisition functions outperform the Factorial design in the rate of convergence for the Lack of Fusion parameter. However, in comparison to Figure 4.9 all three B-DoE algorithm acquisition functions NRMSD converge towards zero at roughly identical rates until the 15th iteration step (t). Wherein both EI and MES converge at a much faster rate than GP-CB, but all three B-DoE algorithm acquisition functions outperform the Factorial design.

Comparison of DoE Methods: Regret

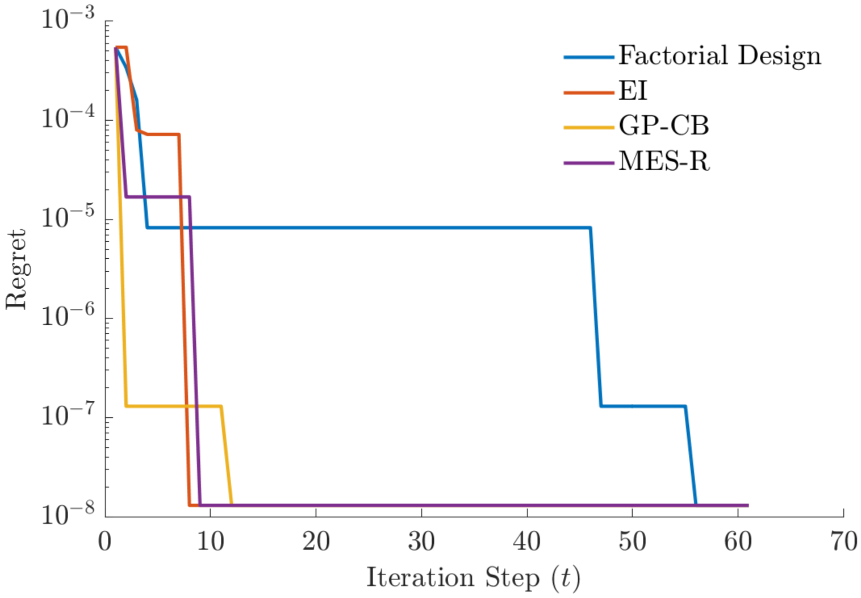


Figure 4.11: Comparison of regret performance for Thermal Strain parameter using B-DoE algorithms from the discrete set of 67 experiments in the SLM manufacturing case against the original scheme: Factorial design

Firstly, in Figure 4.11 all three B-DoE algorithm acquisition functions outperform the Factorial design in locating the global optimum within 12 experiments. Wherein EI is the quickest in identifying the global optimum in 8 experiments, MES next in 9 experiments and finally GP-CB in 12 experiments with the factorial design requiring 56 experiments. This illustrates the potential strength of the B-DoE algorithms over traditional DoE approaches especially on AM problems.

Secondly, in Figure 4.12 only EI and MES outperformed the Factorial design in locating the global optimum with the GP-CB performing significantly worse in locating the global optimum with regard to the Lack of Fusion parameter. Both EI and MES locate the global optimum for the Lack of Fusion parameter in 5 experiments where the Factorial design locates it in 15 experiments, however GP-CB requires 55 experiments. This is likely due to the theoretical properties involved in the formulation of the GP-CB acquisition function in comparison to EI and MES.

Whereby, in the calculation of the acquisition cost typically the predicted mean and standard deviation of an experiment are used as function components to determine the acquisition cost of an experiment which is the case for

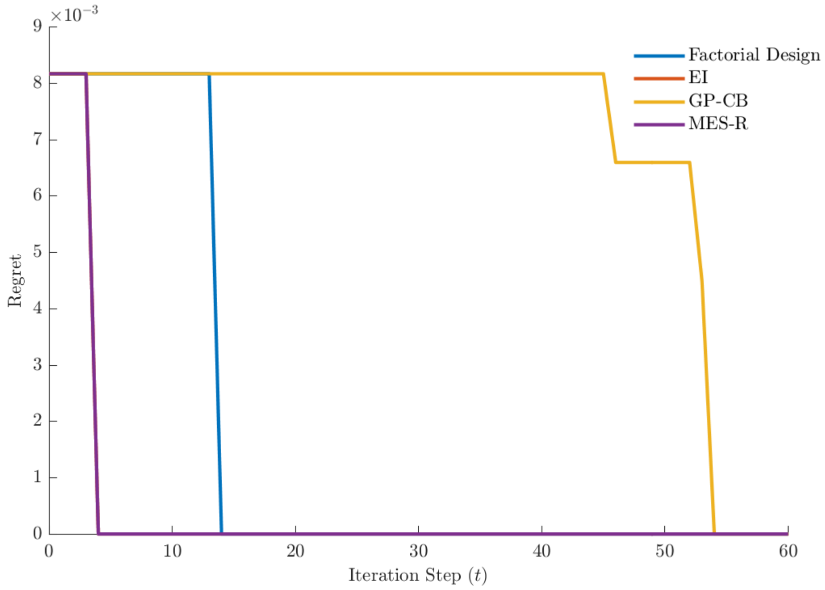


Figure 4.12: Comparison of regret performance for Lack of Fusion parameter using B-DoE algorithms from the discrete set of 67 experiments in the SLM manufacturing case against the original scheme: Factorial design

EI and MES. On the other hand for GP-CB the predicted mean and standard deviation of an experiment directly relate to its acquisition cost as shown in Equation 2.29.

Thus, even in the event of the global optimum being located for one output objective, the experiments acquisition cost with regard to that output objective may only decrease slightly for GP-CB. Whilst EI and MES may see significant reduction and thus the combined acquisition cost becomes dominated by the 2^{nd} output objective, this may not necessarily be this case for GP-CB. This is likely the reason as to why the GP-CB Regret for the Thermal Strain parameter in Figure 4.11 shows the greatest performance of the four DoE methods in contrast to its performance being the worst for the Lack of Fusion parameter in Figure 4.12.

The aim of this work was to demonstrate the capabilities of the B-DoE algorithms in contrast to a traditional DoE scheme and assess the suitability for AM manufacturing problems. In regard to the objectives of this work it can be clearly seen that the B-DoE algorithms provide an alternative approach to efficiently sample the design space to achieve greater performance in both model building and process parameter optimisation for manufacturing processes. In terms of building the GP surrogate model all three acquisition functions provide improved performance with EI and MES slightly outperforming GP-CB.

On the other hand in regard to process parameter optimisation both EI and MES outperformed the traditional DoE scheme whilst GP-CB outperformed for one output objective but not the other likely due to acquisition function structure.

4.5 Theoretical Analysis of BO Acquisition Function Properties

Previously, the B-DoE acquisition functions EI, GP-CB, and MES were assessed for the suitability of application onto DoE problems and more specifically, AM DoE problems using benchmark synthetic functions and a manufacturing case study. Whilst this provides a demonstration of BOs capabilities onto DoE problems the objective of this work is to also provide novel algorithmic development for use-cases of interest in AM. These use-cases detailed in Section 2.1.6 aim to utilise existing BO acquisition functions as a base to extend the B-DoE algorithm for those particular use-cases. The first use-case scenario which will be explored in Chapter 5 is the development and assessment of a Batch Bayesian Experimental Design Optimisation (BB-DoE) algorithm.

The choice of which acquisition function to use as a base is not arbitrary as their properties will impact the approach taken to derive the BB-DoE methodology. Therefore, an analysis of their properties will be performed throughout this section to ascertain which acquisition function theoretically will serve as the most suitable base. The BB-DoE methodology to be developed in the next Chapter has a few pre-specified characteristics that must be considered during the analysis of each respective acquisition function, as specified in Section 2.3.1 restated below:

- Non-Greedy Batch Selection
- Low Computational Complexity

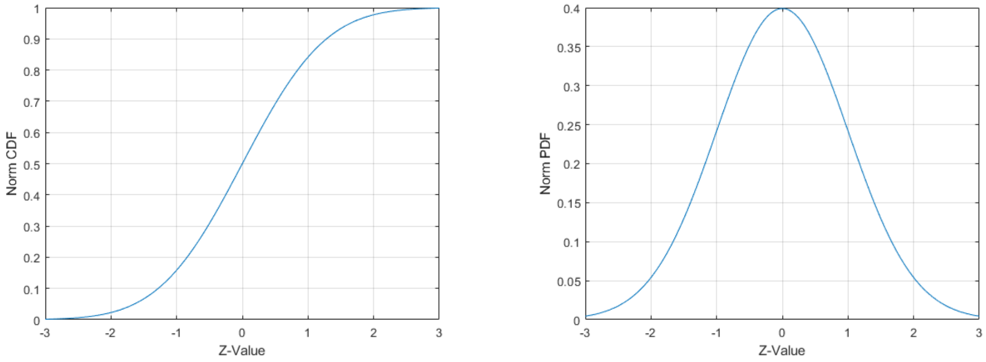
4.5.1 Expected Improvement

Expected Improvement (EI) is one of the more popular acquisition functions available in literature as it has both a simple closed-form (See Equation 2.27), and does not require any tuning due to no hyper-parameters in its formulation. As no hyper-parameters are used to drive the selection pressure of EI, it instead uses the improvement function (See Equation 2.25) as its base to choose the next best candidate.

The improvement function provides the selection pressure necessary to discern which candidate experiments are expected to provide improvement over the current iteration's incumbent. Whereby, the incumbent is predominantly set to the current best experiment performed thus far, such that for a candidate to have a favourable EI acquisition it must be relatively close in value to the incumbent.

This presents an issue, as despite EI providing an improved exploratory nature over methods such as Probability of Improvement (PI), it still tends towards over-exploitation in many situations especially in the event of locating a local optimum [26]. The over-exploitation is derived from the characteristic which its popularity is made, the improvement over the incumbent. Theoretically, as shown in Equation 2.27 its acquisition value is assigned to any candidate which has a standard deviation greater than zero but, in practice that isn't the case.

The EI acquisition value is guided by two main functions within its closed-form: Standard Normal Cumulative Density Function Φ and Standard Normal Probability Density Function φ of each candidates' Z (See Equation 2.28). Figure 4.13 demonstrate the possible multipliers achievable by candidates depending on their Z ranging from $[-3, 3]$, where a $Z = 0$ is obtained when a candidate's output is identical to the incumbent.



(a) Plot of standard normal cumulative density function in the range [-3,3]

(b) Plot of standard normal probability density function in the range [-3,3]

Figure 4.13: Plots of the standard normal CDF and PDF profiles for EI's Z in the range [-3,3]

In comparison of Figure 4.13 and Equation 2.27 it can be observed how EI can still lead to over-exploitation of local optimums. As previously discussed, Equation 2.27 can be separated into 2 components: the predicted mean (output) contribution and the predicted standard deviation (uncertainty) contribution.

The φ value is maximised at zero when the candidate solution outputs are identical to the current best point. The φ value also decreases regardless of when the candidate improves or worsens over the incumbent, as shown in Figure 4.13b. Thus, even if a candidate is located at maximum uncertainty at the same predicted output of the incumbent, the φ can only contribute a small acquisition value to the total.

On the other hand, the Φ value does not decrease once the candidates' output surpasses the incumbent but continues to increase to a pre-determined limit as seen in Figure 4.13a. Therefore, as long as the $Z \geq 0$ its contribution will dominate the EI acquisition value. This can be illustrated by looking at a combined probability plot of Φ and φ, depicted in Figure 4.14.

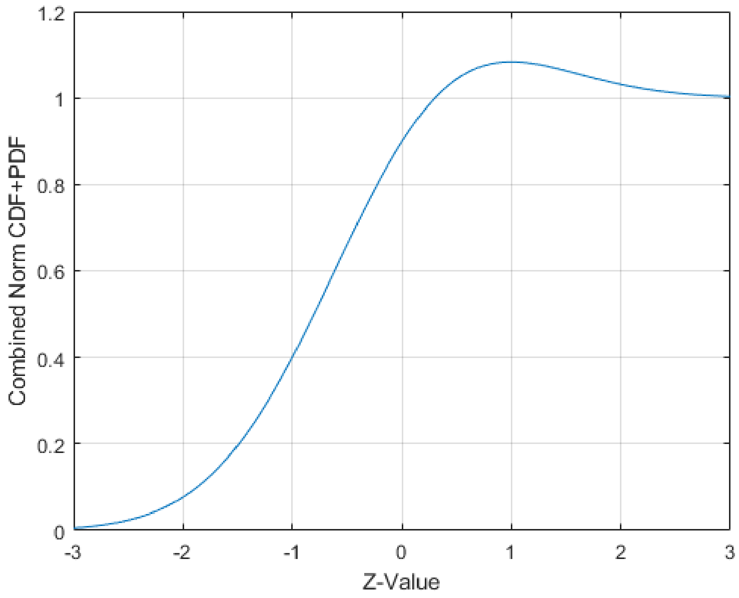


Figure 4.14: Plot of the combined standard normal density functions (CDF and PDF) of EI's Z in the range [-3,3]

As it can be seen in Figure 4.14 as the Z increases it maximises at a small improvement over the incumbent, this settles to a probability of one for any $Z > 0$. Hence, the EI acquisition value is maximised at any candidate that improves over the incumbent but only places minimal value at areas of high uncertainty especially if they are predicted below the current incumbent. This can be demonstrated by calculating the EI for a range of predicted mean (μ_t) and standard deviation (σ_t) combinations to generate an EI surface, the varied ranges are set as in Table 4.2.

Table 4.2: EI surface plot for varied predicted mean (μ_t) and standard deviations (σ_t) values for all combinations that equate to the specified $Z-$score range.

EI Surface Ranges		
Parameter	Lower Bound	Upper Bound
Z Value	-3	inf
Predicted Mean (μ_t)	-3	3
Predicted Standard Deviation (σ_t)	0	1
EI Surface Settings		
Parameter	Value	
Incumbent	0	
Increment	0.01	

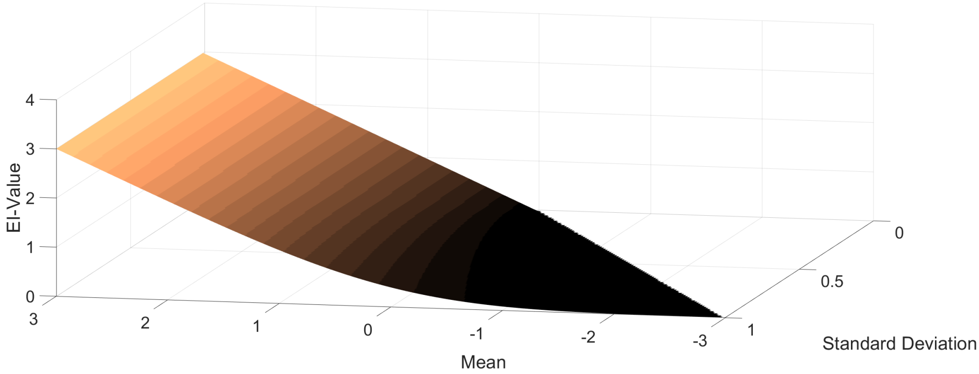


Figure 4.15: Plot of a EI surface for the predicted mean (μ_t) and standard deviation (σ_t) combinations as specified according to the range and settings set out in Table 4.2

As was previously stated when the predicted mean (μ_t) is less than or equal to the incumbent regardless of the predicted uncertainty (σ_t) the EI acquisition value remains relatively small. However, once the predicted mean (μ_t) surpasses the incumbent, their $Z-$score also become positive and the EI acquisition begins to climb steadily. The increase in EI corresponds with the increase in the difference between the predicted mean (μ_t) and the incumbent, as shown in Equation 2.27.

Thus, in the event that the incumbent has located a local optimum, if the global optimum is located in a region of high uncertainty but low predicted output which is much lower than the incumbent, it is unlikely to be explored. In literature some works sought to alleviate this issue by adding an additional hyper-parameter to artificially increase or decrease the incumbent by a small value [146]. Wherein Equation 2.28 is modified by an additional hyper-parameter (ζ) to present a modified Z Equation 4.5 below:

$$Z = \begin{cases} \frac{\left(\mu(\mathbf{x}) - f\left(\mathbf{x}^{+}\right) - \zeta\right)}{\sigma(\mathbf{x})} & \text{if} \quad \sigma(\mathbf{x}) > 0 \\ 0 & \text{if} \quad \sigma(\mathbf{x}) = 0 \end{cases} \tag{4.5}$$

Where, ζ is a small tunable constant to balance the exploitation-exploration trade-off within the EI acquisition function. The inclusion of the additional parameter promotes the EI acquisition value to perform more exploration as it artificially reduces the value of the incumbent. Intuitively Lizotte [146] suggested to use a time-varying hyper-parameter that promotes early exploration before exploitation but found this to be ineffective. Despite these methods' potential, it is infrequently applied in literature as the addition of this hyper-parameter requires strict tuning as to prevent over or under-exploration if tuned incorrectly. This inherently detracts from the intended benefit of no tuning required for the EI acquisition function.

Berk et al. [26] also suggested using Thompson Sampling (TS) for the incumbent instead of the current best solution in their E^3I algorithm. TS was introduced to estimate the incumbent due to two useful properties. Firstly, in the absence of noise, the sampled functions will agree with the current sample points exactly. As such the optimum of the TS will occur at either the true optimum or will over-estimate the optimum.

$$g^{*} = \max_{\mathbf{x}} g(\mathbf{x}) = f\left(\mathbf{x}^{+}\right) = y^{+} \qquad \text{or} \qquad g^{*} > y^{+}$$

Secondly, as the number of iterations increase the TS should converge towards to true function, hence the incumbent will tend towards the true incumbent. This allows for an increased level of exploration but still returns to the exploitation properties of EI as the number of iterations increase. This approach is more advantageous than Lizotte [146] approach as it does not require additional tuning parameters but improves upon the over-exploitative nature of EI. Although E^3I algorithm is not without issues as the sampling of TS increases the computational complexity, which scales with both the number of data points and number of input variables [26].

As determined through the analysis of EIs properties the acquisition function is prone to over-exploitation due to the choice of the incumbent, despite some attempts in literature to alleviate this issue. Wherein as long as a candidate improves over the incumbent, a high acquisition value is likely. Whilst if the $Z < -3$ regardless of whether the candidate is in a highly uncertain region that has been unexplored, it is likely to have negligible acquisition value.

This over- exploitatory nature of the EI acquisition function detracts from its suitability for use as the base of the non-greedy BB-DoE algorithm to be developed in Chapter 5. As candidates are chosen which represent a wide range of characteristics as well as their intra-batch interactions, for which in EI the selection of the incumbent may inhibit exploration. Whereby, regions of high uncertainty may be unexplored in favour of local exploitation for multiple candidates within a batch set.

4.5.2 Max-Value Entropy Search

Max Value Entropy Search (MES) is one of the most recent additions of entropy or information-based acquisition functions within the literature. MES determines the expected information gain about the location of the global optimum in the output space [230]. Where the primary benefit of using an information-theoretic acquisition function is to provide a probabilistic approach to quantify the expected increase in global knowledge, which is not captured by improvement or optimistic approaches [100].

Despite the probabilistic approach adopted by information-theoretic acquisition functions most suffer from similar issues to varying degrees. Whereby, in order to calculate the entropies of posterior distributions of the global optimiser, these distributions are intractable and require approximations. Historically, ES [100], and PES [101] required the approximation of the $p\left(\mathbf{x}^* \middle| D\right)$ which was a d-dimensional distribution and thus subsequently required expensive approximations using methods such as Expectation Propagation (EP) [230].

The core contribution of Wang and Jegelka [230] is to find candidates that maximise the gain of information about the global optimum in the output space or y^*. This information takes the form of $p\left(y^* \middle| D\right)$ which is a one-dimensional intractable distribution, thus lowering the computational complexity in comparison to ES and PES. Despite the lowered computational complexity involved in MES, an approximation of the global optimum output y^* is required. This is achieved by sampling functions from the posterior Gaussian distribution and maximising the sampled functions [230].

While this provides a lower computational complexity information-theoretic approach in sequential selection, it may not be the case when considering the selection of Q candidates simultaneously. As when adapting for the selection of Q candidates, in regard to MES, this equates to the selection of Q candidates' outputs.

$$\mathrm{f}_{\mathbf{x}_Q} = \left[\mathrm{f}_{\mathbf{x}_1}, \mathrm{f}_{\mathbf{x}_2}, \ldots, \mathrm{f}_{\mathbf{x}_q}\right]$$

By extending MES in this manner Equation 2.34 requires modification into Equation 4.6. Also, much like the assumption in the derivation of the original MES where $\mathrm{f}_{\mathbf{x}} \leq \mathrm{f}\left(\mathbf{x}^*\right)$ in the derivation of a batch method, an extended assumption would also be made $\mathrm{f}_{\mathbf{x}_Q} \leq \mathrm{f}\left(\mathbf{x}^*\right)$.

$$u_{\mathrm{MES-B}} = \mathrm{H}\left[\mathrm{p}\left(\mathrm{f}_{\mathbf{x}_Q} \middle| \mathrm{D}_t\right)\right] - \mathbb{E}_{\mathrm{p}\left(\mathrm{f}_{\mathbf{x}_Q} \middle| \mathrm{D}_t\right)} \mathrm{H}\left[\mathrm{p}\left(\mathrm{f}_{\mathbf{x}_Q} \middle| \mathrm{f}_{\mathbf{x}_Q} \leq \mathrm{f}\left(\mathbf{x}^*\right), \mathrm{D}_t\right)\right] \tag{4.6}$$

Where, $\mathrm{H}[.]$ is the differential entropy of probability distribution $\mathrm{p}(.|.)$, $\mathrm{p}\left(\mathrm{f}_{\mathbf{x}_Q} \middle| \mathrm{D}_t\right)$ is a q-dimensional Gaussian distribution and $\mathrm{p}\left(\mathrm{f}_{\mathbf{x}_Q} \middle| \mathrm{f}_{\mathbf{x}_Q} \leq \mathrm{f}\left(\mathbf{x}^*\right), \mathrm{D}_t\right)$ is a multivariate truncated normal distribution.

In this modified variation of the MES for batch selection, the first term has become a q-Dimensional Gaussian distribution whose entropy can still be analytically calculated [212]. However, the second entropy term is for a multi-variate truncated normal distribution which either would require an approximation approach for conversion into an analytically tractable form, or a greedy selection approach [212].

Similarly, in Henrández-Lobato et al. [101] a multivariate truncated normal distribution can be approximated as a Gaussian distribution using Expectation Propagation (EP). This approximation subsequently makes the entropy calculation in the second term of Equation 4.6 analytical, but also introduces expensive approximations into the formulation.

Alternatively, the batch MES acquisition function could be solved iteratively using a greedy batch selection scheme approach. Whereby, as a greedy selection approach the MES acquisition function would be solved sequentially to select a single candidate at a time, as detailed in Section 2.3.1. After which the selected candidates' output is *hallucinated* in order to allow for subsequent selection to be conditioned on additions to the batch set previously [212]. This process is repeated until the batch set is filled and the subsequent experiments are performed to update the data set.

After review of the properties of MES acquisition function in order to extend it towards a batch selection scheme, the core benefit of its properties would lead into greater computational complexity. As in order to select a batch set of candidates simultaneously a multivariate truncated normal distribution would require approximation onto a Gaussian distribution using EP.

This would significantly increase the computational complexity of the chosen scheme which deviates from the type of acquisition function that is intended to be the base for development in Chapter 5. In order to alleviate these concerns a greedy selection scheme could be utilised but the desired approach is for a non-greedy batch selection scheme as previously specified in Section 4.5.

4.5.3 Gaussian Process Confidence Bounds

Gaussian Process Confidence Bound (GP-CB) acquisition functions are most commonly split between the upper bound variant (GP-UCB) for maximisation problems and lower bound variant (GP-LCB) for minimisation problems. Regardless of which variant is implemented the three core components remain unchanged; predicted mean, predicted standard deviation, and time-varying exploitation-exploration trade-off β_t, as defined in Equation 2.29.

Much like EI, GP-CB has a simple closed form expression allowing for easy computation but on the other hand also contains a hyper-parameter to control the trade-off between exploring uncertain regions or exploiting areas of a potential optimum. As was originally demonstrated in Figure 2.11 the GP-CB acquisition value corresponds to a candidates' upper or lower limit of their confidence bound. Whereby, the β_t acts as a multiplier to the degree in which the bound (standard deviation) is stretched from the predicted mean. The tuning of the β_t predominately effects the selection pressure of the GP-CB acquisition function during the optimisation. As a result, no additional modifications or functions are applied to the candidates' predicted mean and standard deviation.

Subsequently, the choice and tuning of the β_t becomes of paramount importance in determining the performance of GP-CB algorithms [157]. Merrill et al. [157] reviewed a variety of β_t settings on a set of synthetic benchmark functions and observed highly variable performance. The design of the β_t in literature suggests early exploitative behaviour prior to an increase in exploration in later stages of the optimisation. Srinivas et al. [208] originally developed GP-CB and suggested a variety of β_t settings dependent on their domain knowledge in their Theorem's 1 and 2 as defined in Equation 4.1 and Equation 4.7 respectively.

$$\beta_t = 2\log\left(\frac{t^2 2\pi^2}{3\delta}\right) + 2d\log\left(t^2 dbr\sqrt{\log\left(\frac{4da}{\delta}\right)}\right) \tag{4.7}$$

where, $\log(\ldots)$ is the natural log of its argument, t is the iteration number, d is the number of input variables, r is the largest range on the input variables and a, b, and δ are constants of value 1, 1, and 0.1 respectively.

For comparison in Equation 4.1, $|\,X\,|$ or the number of discretised points is set to 1000 points for each input dimension [157]. However in Srinivas et al. [208], despite the β_t settings for Equations 4.1/4.7 having proven regret bounds, the authors often found the algorithms improved by scaling the β_t down by a factor of 5 [208]. Kandasamy et al. [121] also suggested an alternative β_t setting in Equation 4.8.

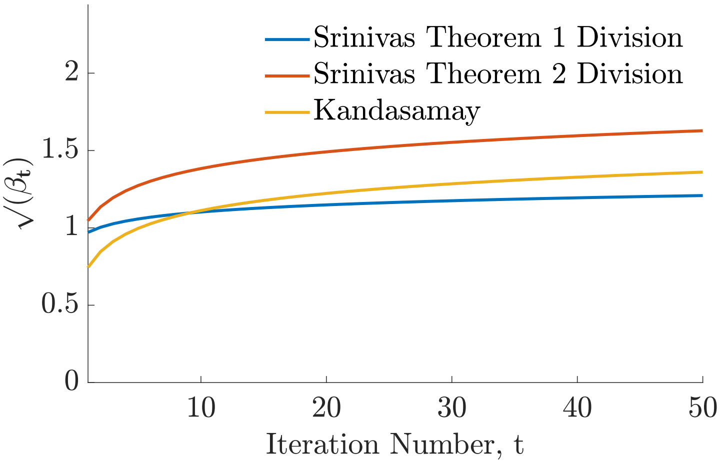
$$\beta_t = 0.2d \log(2t) \tag{4.8}$$

Figure 4.16 illustrates how the β_t varies during the B-DoE algorithm as each candidate is chosen. Regardless of whether a problem with a lower input dimension as shown in Figure 4.16a or higher input dimension as shown in Figure 4.16b each β_t initially starts small before growing in value.

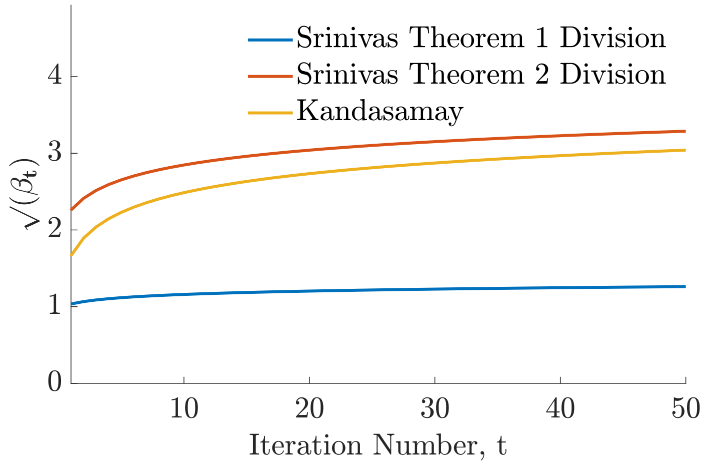
Given the β_t acts as a multiplier onto a candidates' predicted standard deviation, it seeks to manipulate the exploration aspect of GP-CB. In the case of literature promoting early exploitation with later exploration, as exemplified by the β_t profiles in Figure 4.16. Fundamentally by setting the β_t to a small value it encourages the B-DoE algorithm to select candidates that exploit the current optimal regions in the design space. However, Figure 4.16 suggests the β_t in literature are encouraging exploitation early which is a potentially limiting approach, causing optimisations to be entrapped in local optimums.

Using a more exploitative β_t profile may be beneficial initially when the knowledge of the design space is low and there is no information of desirable regions in the search space. Therefore, by following an initially exploitative β_t allows for quick identification of potentially beneficial regions, while also being a limiting approach in the event of multiple local optimums. As in the event of many local optimums, early exploitation may locate a single optimum solution and be entrapped for a majority of the experimentation budget.

Another drawback surrounding the β_t profiles suggested is that their definitions with provable regret bounds are based upon known domain knowledge [157]. This is unusual in the field of BO which explores global optimisation on *unknown* black-box optimisation problems. The assumption of domain knowledge is only practical when assessing performance on known problems, or a situation in which a domain expert can provide additional information. Assessing performance on known synthetic benchmarks may lead to good performance but, for later expansion on manufacturing DoE problems without a domain expert could lead to under-performance using the same β_t profiles.



(a) Plot of Literature β_t profiles: Srinivas et al. [208] Theorem 1 and 2 (divided by factor of 5) and Kandasamy et al. [121], for a small number of input dimensions d = 2



(b) Plot of Literature β_t profiles: Srinivas et al. [208] Theorem 1 and 2 (divided by factor of 5) and Kandasamy et al. [121], for a larger number of input dimensions d = 10

Figure 4.16: β_t profiles of: Srinivas et al. [208] Theorem 1 and 2 (divided by factor of 5) and Kandasamy et al. [121], varied along 50 optimisation iterations for a representative low (d = 2) and high number of input (d = 10) dimensions.

From the context of a manufacturing DoE standpoint a variety of DoE goals need to be satisfied as specified in Section 2.1.9, such as building a well-defined and accurate model alongside locating optimal process parameters in as few experiments as possible. Furthermore, non-greedy BB-DoE algorithms aim to select batches of candidates that maximise the information gained in each batch, suggesting a more exploratory β_t in early batches to explore uncertain regions is more desirable. Therefore, by providing a well-rounded and scalable β_t profile which promotes early exploration could enhance the GP-CB acquisition function effectiveness to serve as a base for a BB-DoE algorithm.

Through the analysis of the GP-CB acquisition functions properties the primary concern is the tuning of the β_t profile. As the β_t profile has been shown in the literature to have a large impact on optimisation performance and the time-varying profiles developed thus far tend to over-exploit initially [157]. In spite of this, the GP-CB has a simple closed-form whose value corresponds to either the upper or lower bound of a candidates' predicted mean and standard deviation. This simple formulation could allow for additional functionality to be added to the acquisition function, increasing its suitability for use as the base of the non-greedy BB-DoE algorithm to be developed in Chapter 5.

4.6 Conclusion

The evaluation and assessment of the BO framework for use within DoE were presented in this chapter as an ARSM approach for expensive manufacturing DoE problems with a particular focus on Additive Manufacturing domains. Wherein, three BO acquisition cost functions were assessed as part of a B-DoE algorithm to determine which among them would be a suitable and capable approach for implementation within a B-DoE algorithm for use on expensive additive manufacturing DoE problems.

The evaluation of the three acquisition functions were split into three methodological assessments: performance on synthetic benchmarks of varying difficulties, performance on a preliminary SLM manufacturing case study and the analysis of each acquisition cost functions properties. By comparing and contrasting the performance of each acquisition cost function on all assessment methods, an analysis can be drawn on their overall performance for use on expensive manufacturing DoE problems. In particular, the overall assessment of the three acquisition cost functions determines which acquisition cost function is most suitable for further algorithmic development onto both batch and multi-objective batch algorithms later within this thesis.

In regard to the goal ascertaining which acquisition cost function is most suitable for further algorithmic development, a key difference between the DoE literature and the BO literature in their respective fields must be drawn.

Whereby in BO literature the goal is to locate the global optimum as efficiently as possible, whereas in DoE literature the goals include both optimising the process parameter settings and developing an accurate model. Thus, the purpose of the assessments was to ascertain if there was merit for using BO for DoE in a B-DoE algorithm and of the three acquisitions functions investigated which was the most suitable in regard to both global optimisation and model development.

Firstly, the EI acquisition function demonstrated the greatest exploitative performance among the three acquisition functions during both the synthetic benchmarks and SLM manufacturing case study, efficiently locating the global optimum in as few experiments as possible. Coupling this performance with the property of having no hyper-parameters to tune would suggest that the EI acquisition function as the most desirable approach for further algorithmic development. However, as mentioned in the theoretical analysis EI is prone to over-exploitation and getting caught in local optima [146]. This is due to the incorporation of the improvement function which places high acquisition value on experiments that improve over the current best. This becomes a detrimental property early during the optimisation if the solution gets trapped in a local optimum, as the EI places low acquisition value in areas of high uncertainty that do not improve over the incumbent.

Therefore, when considering the EI acquisition function for use in DoE it is likely to under-explore the design space especially in multi-modal design space landscapes, inhibiting its own performance. However, the inclusion of constraints or trade-off parameters could improve the performance as has been suggested in [146] [26], but their fine-tuning would become the deciding factor in performance.

Secondly, in contrast to the EI acquisition function MES demonstrated the best modelling performance amongst the three acquisition functions during both the synthetic benchmarks and the SLM manufacturing case study. Also aside from the Hartmann synthetic benchmark, the exploitative performance was close to that of EI if not slightly worse. In a similar vein to EI the issues arise regarding its theoretical properties suggesting poor integration within a B-DoE framework.

The core of the MES acquisition function is its probabilistic approach to maximise the gain in information about the location of the global optimum output using the entropy of probability distributions. The issues stem from the definition of these probability distributions in that some are analytically intractable and so require approximations increasing the computational complexity of the acquisition function.

Whilst MES does provide a lower computational complexity in comparison to other information-based acquisition functions, if extended to a batch scheme the computational complexity would increase. This would be due the premise discussed in Section 4.5.2 where a batch derivation would result in optimising the entropy of a multi-variate truncated normal distribution. This requiring approximation using Expectation Propagation (EP) or optimising using a greedy selection scheme, which is against the desired goal of a non-greedy batch B-DoE scheme.

Finally, GP-CB on both the synthetic benchmarks and the SLM case study was middle of the pack in terms of exploitative performance but showed poor modelling performance in each synthetic benchmark performing slightly worse than the next best acquisition function. Although GP-CB had a very poor performance on the Mixture of Cosines NRMSD, upon further analysis of the data sets this was found to be due to over-exploitation once the global optimum had been located leading to excessive wasted experiments.

From the theoretical analysis it was discerned that GP-CBs performance in BO is predominately determined by the selection and tuning of the β_t [157]. Wherein the BO literature the β_t often opts for an early exploitation with increased exploration as the optimisation proceeds. However, by modifying the β_t profile for increased early exploration, the B-DoE algorithm should be able to improve the modelling performance and prevent getting caught in local optima. Another beneficial characteristic of the GP-CB acquisition function is much like EI as it has a simple closed-form but is not limited by a transformative acquisition function. As such, additional functionality could be added to the acquisition function formulation such as methods to prevent over-exploitation for example; constraints. These properties allow for the potential of improving the performance of the GP-CB acquisition function for the desired DoE goals specified previously in Section 4.5.

Hence, EI and MES are not suitable design choice for a non-greedy batch DoE algorithm which will developed further in Chapter 5 and 6 for batch and multi-objective batch DoE respectively, rather GP-CB is the most optimal choice. Whilst GP-CB initially had reasonable B-DoE performance in exploitative aspects and under-performance in explorative aspects, these could be resolved through improved tuning of the β_t and additional functions to manage over-exploitative properties.

Chapter 5

Batch DOE Optimisation

In Chapter 4, an investigation was conducted to evaluate whether BO would be a suitable approach for a sequential DoE in an additive manufacturing setting, with the intention of identifying which acquisition function would be suitable for adaptation and extension into a Batch B-DoE. GP-CB was determined to be the most suitable acquisition function for this task due to its theoretical properties and performance on both benchmark and manufacturing studies.

The primary novelty within Chapter 5 stems from the structure of the developed BB-DoE algorithm as follows: GP surrogate model with cross-validation tuning, a novel non-greedy batch optimistic-based acquisition function and the novel design of two batch constraints (IBC and ODC) included within the surrogate-assisted constraint handling GA optimiser. The development of this algorithm is novel as currently BB-DoE schemes rely heavily upon greedy selection methods [192], which will be assessed upon a series of synthetic benchmark functions used in Section 4.3.

Also a novel application of the developed BB-DoE algorithm (Algorithm 5.2) onto a real-world additive manufacturing case investigating the Dendritic Arm Spacing of nickel super-alloy using Directed Energy Deposition (DED) is presented [246]. Currently in BO DoE literature applied onto AM problems according to [246] there are zero BB-DoE applications, although there has been Constrained Bayesian Optimisation (CBO) implemented it was using cEI [198] rather than surrogate-assisted constraint-handling in Algorithm 5.2.

The final novelty of Chapter 5 is the development of a new stopping criterion, that will suggest early termination of the optimisation once the model predictions become consistent. This will be assessed in parallel with the to be developed BB-DoE scheme on both the synthetic benchmark functions and on the real-world additive manufacturing case.

5.1 Introduction

In this chapter, the aim is to derive a suitable Batch Bayesian Experimental Design Optimisation (BB-DoE) approach to tackle potential future manufacturing issues with a particular focus in additive manufacturing. Additive manufacturing would benefit immensely from a suitable BB-DoE approach due to the reduced manufacturing scale allowing for operation of multiple processes in parallel. Also, additive manufacturing can build multiple different parts in parallel (Nesting) which would suit a BB-DoE approach.

As previously discussed in Section 2.3.1, a Greedy Batch scheme whilst fast and easy to modify from a sequential scheme, has a lower likelihood of locating the global optimum. Each sequential selection will influence subsequent experiments selected within the same batch set. This can lead to non-optimal batch choices each iteration preventing the identification of the global optimum [95]. In contrast, a Non-Greedy Batch scheme accounts for the inter-batch interactions without any prior selections, thus allowing for the identification of an optimal combination of experiments within a batch set. Whilst beneficial theoretically it is not without its demerits, due to increased computational complexity in implementation to identify the optimal batch set combination.

5.2 Batch Extension of GP-CB Function

The goal of BB-DoE is to optimise an objective function, which is unknown but can be evaluated pointwise exactly or in the presence of noise in multiple locations in parallel [192], as defined in Equation 5.1.

$$B_{t+q} = \arg\max u\left(B_t \middle| D_t\right) \quad B_{t+q} = [x_{t+1}, x_{t+2}, \ldots, x_{t+q}] \tag{5.1}$$

Where, B_{t+q} is the batch vector set of Q candidates to be evaluated. At each iteration, a batch set of Q points must be selected for evaluation or in the context of DoE for experimentation. We seek to select the batch of experiments Non-Greedily such that all members of the batch set are chosen whilst accounting for their interactions with each remaining member.

5.2.1 Batch Conditioning

To optimise Equation 5.1 a suitable BB-DoE acquisition function must be derived using the GP-CB function as a base. Suppose a total acquisition cost was defined as the joint total of each individual member comprising the batch set as shown in Equation 5.2.

$$u_{\mathrm{Batch}}\left(B_t \middle| D_t\right) = u_{\mathrm{GP-CB}}\left(x_{t+1} \middle| D_t\right) + \ldots + u_{\mathrm{GP-CB}}\left(x_{t+q} \middle| D_t\right) \tag{5.2}$$

Although, this formulation would likely lead to the selection of experiments in identical locations within the search space. In greedy BB-DoE schemes this issue is alleviated through the sequential selection and hallucination of outputs, see Definition 2.3.1. These mechanisms periodically update the acquisition costs within a batch set as each experiment is selected since each experiment will influence subsequent selections. However, in non-greedy BB-DoE schemes the selection occurs simultaneously, requiring a mechanism accounting for the interactive effect of different members of the batch set on the joint acquisition value.

For the GP-CB acquisition function it is only comprised of three elements: predicted mean, predicted standard deviation, and the time-varying β_t parameter. In order to influence the batch acquisition function cost calculation, one of these elements requires modification. [59] first noted that the predictive standard deviation of an experiment could be conditioned upon the selection of unobserved experiments as Equation 2.21b is solely dependent upon the inputs.

Therefore, as each greedy optimal experiment is added to the batch set the predictive variance of the remaining experiments within the search space can be updated by conditioning upon the inputs of the current greedy batch set experiment. This process of optimisation, selection, and updating the predictive variance of the search space is repeated until the batch set is filled. However, the desired goal is for a Non-Greedy Batch GP-CB (NGB-GP-CB) acquisition function which accounts for all batch set interactions rather than conditioning upon the most recently added batch set experiment. Therefore, the conditioning step must occur simultaneously on a batch set combination rather than sequential conditioning on each greedily selected experiment, to prevent unexplored optimal combinations.

Definition 5.2.1 (Conditioning).
The evaluation of a function under the belief that a specified action, result or event has already occurred. E.g. The acquisition value of A given B has occurred is $u(A|B)$.

The change in acquisition function value for one point in a batch set can be analytically evaluated by setting the remaining points within the batch as *conditioned data* when calculating the acquisition cost. This can then be repeated for each experiment in the batch set until each acquisition value has been updated, then the total acquisition cost of the batch set can be evaluated as a whole.

Using an example of batch size Q = 2, both batch set members must be evaluated for their acquisition cost once conditioned upon by the other members of their batch set. Firstly, x_{t+1} is conditioned with x_{t+2} by assuming that the experiment has already been performed, as such updating the observed data set as follows with its known inputs and hallucinated outputs.

$$u_{GP-CB}\left(x_{t+1}\middle|D'_{t+1}\right) \quad \text{with} \quad D'_{t+1} = \left[D_t, \left(x_{t+2}, \mu_{t+2}\right)\right]$$

Secondly, in a similar process to evaluating x_{t+1} by conditioning on x_{t+2} with a hallucinated output, the same process is repeated for x_{t+2} as below:

$$u_{GP-CB}\left(x_{t+2}\middle|D'_{t+2}\right) \quad \text{with} \quad D'_{t+2} = \left[D_t, \left(x_{t+1}, \mu_{t+1}\right)\right]$$

Once the acquisition costs for the conditioned and hallucinated data sets are performed the joint acquisition cost is calculated in Equation 5.4:

$$u_{Batch}\left(B_t\middle|D_t\right) = u_{GP-CB}\left(x_{t+1}\middle|D'_{t+1}\right) + u_{GP-CB}\left(x_{t+2}\middle|D'_{t+2}\right) \tag{5.3}$$

When conditioning data sets and hallucinating unobserved outputs for batch sizes greater than Q > 2, each batch set member is conditioned on all of the remaining batch set points as shown below for Q = 3 on x_{t+1}:

$$u_{GP-CB}\left(x_{t+1}\middle|D'_{t+1}\right) \quad \text{with} \quad D'_{t+1} = \left[D_t, \left(x_{t+2}, \mu_{t+2}\right), \left(x_{t+3}, \mu_{t+3}\right)\right]$$

Thus, experiments in close proximity to one another will receive a lower predictive variance which subsequently will lower their GP-CB acquisition cost. This in turn will lower the NGB-GP-CB acquisition cost of a batch if too many points are in close proximity to one another. In order to evaluate the total acquisition value of a batch set, the expected GP-CB acquisition cost of each point in the batch set are summed. Following the previous example of Q = 2, the NGB-GP-CB acquisition function can be derived in Equation 5.4.

$$u_{NGB-GP-CB}\left(B_t\middle|D_t\right) = \sum_{q=1}^{Q} u_{GP-CB}\left(x_{t+q}\middle|D_t, B_t \setminus x_{t+q}\right) \tag{5.4}$$

Where, $B_t \setminus x_{t+q}$ is the batch vector set containing every candidate except for the batch candidate q that is being evaluated. By optimising Equation 5.4 on a variety of batch combinations within the design space the next optimal batch set can be determined for selection at each iteration.

5.2.2 Hallucinated Outputs

Unlike the predictive variance, the predictive mean values at the unobserved experiments are dependent upon the observed outputs of the current observed data set. Therefore, the predictive means cannot be updated prior to experimentation taking place. Another mechanism incorporated into Desautels et al. [59] to assist in this issue is the hallucination of outputs (see Definition 2.3.1). The hallucination of the outputs to be supplemented in the greedy batch set can be estimated through a variety of methods. In particular Ginsbourger et al. [85] introduced two strategies to tackle this issue: Kriging Believer and Constant Liar.

A Kriging Believer is based upon the kriging model (See Section 2.2.4) whereby the hallucinated outputs are estimated using the previous iteration's kriging predictor [85] or predictive mean in GPR. Next the Constant Liar, rather than estimating the output by using the previous iterations predictions, is set to a constant value determined by the user. This constant value does not change with each iteration but rather is set after the initial design, which in Ginsbourger et al. [85] used three variations: minimum, mean, and maximum of the output. In particular for Gaussian Process Batch Upper Confidence Bound (GP-BUCB) each batch set experiments' output is hallucinated through the estimation via their predictive mean values using the previous batch iteration GP model similarly to Kriging Believer.

Rather, for implementation in the NGB-GP-CB acquisition function, the predictive means are calculated for each member of the prospective batch set as shown in Equation 5.5, prior to NGB-GP-CB optimisation. These predictive means of the batch set can then be used as hallucinated outputs within Equation 5.4 in the Non-Greedy batch optimisation.

$$\bar{y}_{t:t+q} = \left[\mu_{t+1}\left(x_{t+1}\right), \mu_{t+1}\left(x_{t+2}\right), \ldots, \mu_{t+1}\left(x_{t+q}\right)\right] \tag{5.5}$$

Where, $\bar{y}_{t:t+q}$ are the hallucinated outputs for the $t + 1$ to Q batch set candidates using their most recently updated predictive means.

5.2.3 Exploration-Exploitation Trade-off

In the design of a new NGB-GP-CB function to be used as part of a BB-DoE approach, two of its three main components have been previously modified or updated so far in Chapter 5: using Conditioning (see Definition 5.2.1) and Hallucinated outputs (see Definition 2.3.1). However, another core mechanism that has a large impact on the exploration and exploitation properties of the algorithm and is of paramount importance for the algorithmic performance, is the β_t parameter [157].

As discussed in Section 4.5, in the original GP-CB function [208] the authors sought to exploit early and explore later in the time-varying profile. Although the β_t parameter still was reduced by a factor of 5 due to its excessive exploration, Berk et al. [27] suggested modifying or redesigning the β_t profile. However, for use in the NGB-GP-CB function the time-varying profile desired is to promote early exploration with later exploitation.

Subsequently, the β_t profile was split into two key components: the time-varying profile and a scaling factor that adjusted the parameter based-upon the DoE problem difficulty, or more simply its dimensionality. Firstly, the time-varying profile was set to start with a large value and gradually decrease towards exploitation as the experiments neared the limit of the experimentation budget by modelling a logarithmic graph using Equation 5.6.

$$\frac{\log\left((BT)^2\right)}{\log\left((T+1)^2\right)} \tag{5.6}$$

Where, T is the total experimentation budget and BT is the current experimentation batch counter. In Equation 5.6 the time-varying profile generated will begin at 1 and tend towards 0 when the numerator begins at the same value as the denominator and reduces to 1. Therefore, in order to achieve this a maximum experimentation budget counter was attached (BT) and reduced with each batch of experiments selected with its formulation ($BT = T + 1 - t$).

Secondly, now the time-varying profile had been set, the scaling component of the β_t parameter is required, which scales the exploration level with the dimensionality of the problem. This component is required due to the nature of *Combinatorial Explosion* in which as the dimensionality of the DoE problem increases, the number of experiments to fully explore the design also increases exponentially. As such the secondary β_t parameter component was modelled after the β_t parameter in [121] which scaled the parameter by $2 * d$.

Combining both the logarithmic time-varying profile with the dimensionality scaling, the new β_t profile is detailed in Equation 5.7 as shown in Figure 5.1.

$$\beta_t = 2 * d * \frac{\log\left((BC)^2\right)}{\log\left((T+1)^2\right)} \tag{5.7}$$

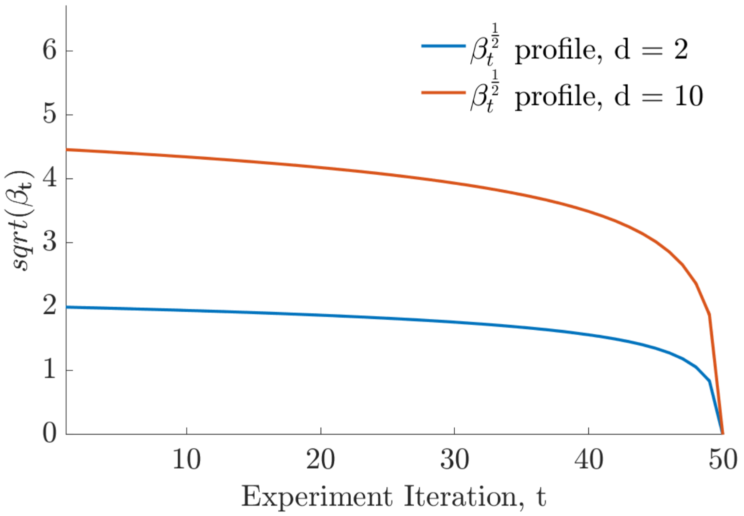


Figure 5.1: New time-varying β_t parameter profile for Non-Greedy Batch GP-CB (NGB-GP-CB) scheme modelled using Equation 5.7, demonstrated at two different DoE problem dimensions with the following settings: T = 50, and d = 2 and 10.

5.2.4 Batch Acquisition Constraints

The updated Non-Greedy Batch GP-CB (NGB-GP-CB) acquisition function accounts for the reduction in variance between experiments within the same batch set, which prevents over-exploration of areas within the search space. However, in the event of multiple experiments being chosen within a close proximity of an area of high exploitation value, the reduction in predictive variance may not significantly impact its potential acquisition value to prevent over-exploitation. To prevent over-exploitation events for which conditioning based predictive variance reductions do not hinder, over-sampling batch constraints were added to the optimiser.

In batch selection there are two scenarios in which over-exploitation can occur:

- Intra-batch over-exploitation
- Observed data over-exploitation

Intra-batch over-exploitation occurs when multiple batch set experiments are located in the same region of search space within a close proximity in either unexplored regions or close to already observed experiments. This is undesirable as when batch set experiments are in too close proximity, potentially good experiments could be overlooked due to lack of exploration.

Furthermore, observed data over-exploitation occurs when potential batch set experiments are placed in close proximity to previously observed experiments. Whilst this may be desirable in areas of high exploitation value at the end of the optimisation, if experiments are permitted to be placed nearby at the start of optimisation, then insufficient exploration may occur.

In both over-exploitation circumstances the repetitive over sampling in areas of good predictive mean value can lead to unexplored areas of the search space. For which a lack of exploration may lead to the unsuccessful identification of the global optimum, poorly defined models and excessive experimentation costs.

One method implemented within literature to mitigate these risks is to utilise constraints to alter the acquisition value which works separately from the acquisition function. Gonzalez et al. [91] first introduced using constraints in conjunction with Greedy BB-DoE to create exclusion zones using Lipschitz constant around experiments chosen within the batch set, see Figure 5.2

In BBO-LP the Lipschitz constant and the maximisation of the observed outputs (M, red line in Figure ??) are used to estimate the exclusion zones originating from the sampled locations expand towards their upper limit determined by L and M. The effect of the local penalizer is to reduce the value of the acquisition in the neighbourhood of the previously sampled points. While this application of constraints has its merits, the Lipschitz constant required for application cannot be calculated directly, rather it is approximated. The Lipschitz exclusion constraint approximation maximises the norm of the gradient of the sampled outputs on a very fine grid, in which the greater the number of samples the more accurate the approximation [91]. Despite a method for approximation being available as it requires a sizable set of samples, in an expensive manufacturing problem domain this is not a practical solution.

For a more practical approach, defining a constraint that is not dependent upon excessive sampling to accommodate the approximation of the exclusion constraint for a BB-DoE method is more desirable. The constraints to be implemented in this thesis, were derived using inspiration from Gonzalez et al. [91] work on BBO-LP as well as the method of inference used within GPR by utilising covariance.

As demonstrated in Section 2.2.1 by calculating the covariance between all observed data using Equation 2.17, the covariance between each experiment

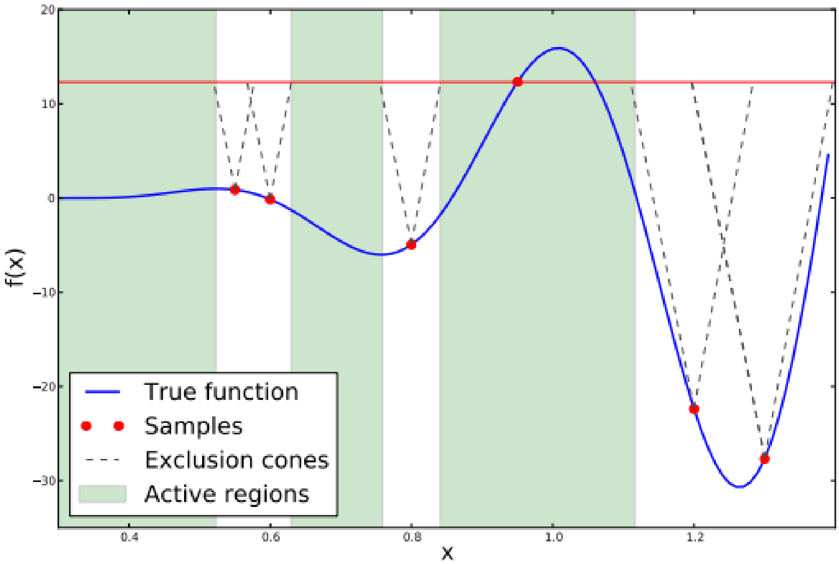


Figure 5.2: Batch Bayesian Optimisation via Local Penalisation constraint function on a 1-D Forrester function $f(x) = (6x-2)^2 \sin(12x-4)$ set in the interval [0.3,1.4]. There are 5 evaluations represented as red dots with their local penalisation zones set as dashed line determined by L and M (red line). Where L is the Lipschitz constant and M is the maximum observed output. After local penalisation is applied the regions of the design space not penalised are represented as the green active regions, taken from [91].

is defined. These covariance values can then be used to find the predictive mean and variance of unobserved experiments through inference using Equation 2.21a and Equation 2.21b respectively. Thus, the covariance between two experiments can simply be thought of as the similarity between two experiments. A low covariance experiment will have a small influence on the inferred values, whilst a large covariance will have a large influence on the inferred values.

In consequence, the covariance can be utilised as the basis for two new constraint functions as demonstrated in Figure 5.3: Observed Data Constraint (ODC) and Intra-Batch Constraint (IBC).

The ODC centres around the previously observed sampling locations similarly to [91], in which when evaluating a batch candidate the covariance between the previously observed data point and the batch candidate is evaluated. If the covariance is found to be within a pre-specified constraint limit the acquisition cost would be reduced by a constraint penalty shown in Equation 5.8. For example the batch candidate shown in Figure 5.3 next to data point x_3 would have its acquisition cost constrained but the other batch can-

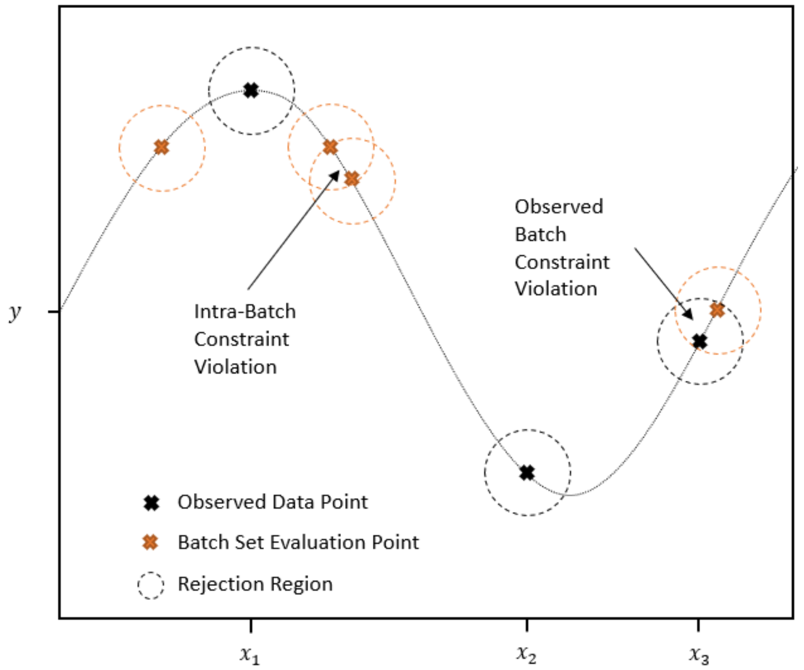

y
Intra-Batch Constraint Violation
Observed Batch Constraint Violation
Observed Data Point
Batch Set Evaluation Point
Rejection Region
x_1
x_2
x_3

Figure 5.3: A conceptual diagram of the NGB-GP-CB constraints: Intra-Batch constraint (IBC) and Observed Data Constraint (ODC) on a 1-D sine wave function. The previously observed data point are shown in black with potential batch candidates shown in orange with the covariance rejection region surrounding each point to illustrate overlaps and thus constraint violations.

didate would not be impacted by ODC.

The IBC centres around each batch candidate, in which the covariances between each batch candidate are evaluated. If the covariance between two candidates are found to be within a pre-specified constraint limit the acquisition cost would be reduced by a constraint penalty shown in Equation 5.8. The goal of the IBC is to prevent over-sampling of batch candidates within similar locations within the search space and thus promote further exploration into uncertain areas to improve model knowledge and potentially locate a global optimum. This is illustrated in Figure 5.3 to the right of data point x_1.

As discussed in Section 3.3.5, a variety of constraint penalties can be utilised within literature such as the *Death* penalty function shown in Equation 3.14. A death penalty constraint is suitable for implementation when the feasibility of an experiment has been lost, such as being located outside the bounds of the search space. However, for the Intra-Batch Constraint (IBC) and Observed Data Constraint (ODC) if a death penalty is used it will significantly reduce the search capabilities of the GA optimiser.

A suitable penalty would then be a severe reduction penalty function, which reduces the overall acquisition value of an experiment upon the condition of violating the set constraint.

$$\chi(\mathrm{BC}) = \chi(\mathrm{BC}) \times \mathrm{S}_{\mathrm{penalty}} \qquad \mathrm{S}_{\mathrm{penalty}} = \begin{cases} 0.25, & \text{if } \mathrm{k}\left(\mathrm{x}_{\mathrm{i}}, \mathrm{x}_{-\mathrm{i}}\right) \geq \mathrm{IBC}_{\mathrm{con}} \\ 0.25, & \text{if } \mathrm{k}\left(\mathrm{x}_{\mathrm{i}}, \mathrm{D}_{\mathrm{t}}\left(\mathrm{x}_{1:\mathrm{t}}\right)\right) \geq \mathrm{ODC}_{\mathrm{con}} \end{cases} \tag{5.8}$$

Where, $\chi(\mathrm{BC})$ is the population cost associated with batch individual BC, $\mathrm{S}_{\mathrm{penalty}}$ is the severe reduction penalty cost, x_{i} is the i^{th} individual C in BC. $\mathrm{x}_{-\mathrm{i}}$ is every individual in the batch individual BC excluding C, $\mathrm{D}\left(\mathrm{x}_{1:\mathrm{t}}\right)$ is all the input locations are previous experiments, $\mathrm{IBC}_{\mathrm{con}}$, and $\mathrm{ODC}_{\mathrm{con}}$ are the IBC and ODC batch set over-exploitation constraints respectively.

Using a reduction penalty is beneficial over a death penalty in this scenario as there may be multiple violations of the constraint per batch set. This aggregation of penalties is applied multiplicatively to the NGB-GP-CB acquisition value. For example, a batch set that contains a high initial acquisition value with multiple constraint violations (See Batch 1 below) may not out perform a batch set of low initial acquisition value with one constraint violation (See Batch 2 below). If a death penalty was used in the prior example, both batch sets would be deemed of equal non-value. Whereas, if a severe penalty is used instead, a batch may be identified that outperforms the other depending on initial acquisition value and the number of constraint violations.

$$\begin{array}{ll}
\text{Batch 1 :}\quad \text{2 Violations} & \text{Batch 2 :}\quad \text{1 Violation} \\
\mathrm{u}_{\mathrm{NGB-GP-CB}} = 10 & \mathrm{u}_{\mathrm{NGB-GP-CB}} = 5 \\
\mathrm{u}_{\mathrm{NGB-GP-CB}} = 10 \times \mathrm{D}_{\mathrm{penalty}}{}^{2} = 0 & \mathrm{u}_{\mathrm{NGB-GP-CB}} = 5 \times \mathrm{D}_{\mathrm{penalty}}{}^{1} = 0 \\
\mathrm{u}_{\mathrm{NGB-GP-CB}} = 10 \times \mathrm{S}_{\mathrm{penalty}}{}^{2} = 0.625 & \mathrm{u}_{\mathrm{NGB-GP-CB}} = 5 \times \mathrm{S}_{\mathrm{penalty}}{}^{1} = 1.25
\end{array}$$

As has been described in multiple instances throughout the BB-DoE framework, a key mechanism of note is the trade-off between exploration and exploitation as the number batches increase. This trade-off benefits from strict initial constraints for exploration but would also benefit from loose constraints for exploitation in latter batches. Therefore, this time-varying mechanism can be built into the constraint violation function.

For the constraint function a time-varying profile was chosen to model the constraint value after a plateau function as shown in Equation 5.9 and Equation 5.10 for a pre-specified rate of convergence, illustrated in Figure 5.4.

$$YP = YP_{max} - \left(YP_{max} - YP_{min}\right) \times \exp\left(-k_{plat} \times t\right) \tag{5.9}$$

$$k_{plat} = \frac{5}{N_{batch}} \tag{5.10}$$

Where, YP is the plateau function output, YP_{max} is the maximum plateau output, YP_{min} is the minimum plateau output and k_{plat} is the rate constant of the plateau function.

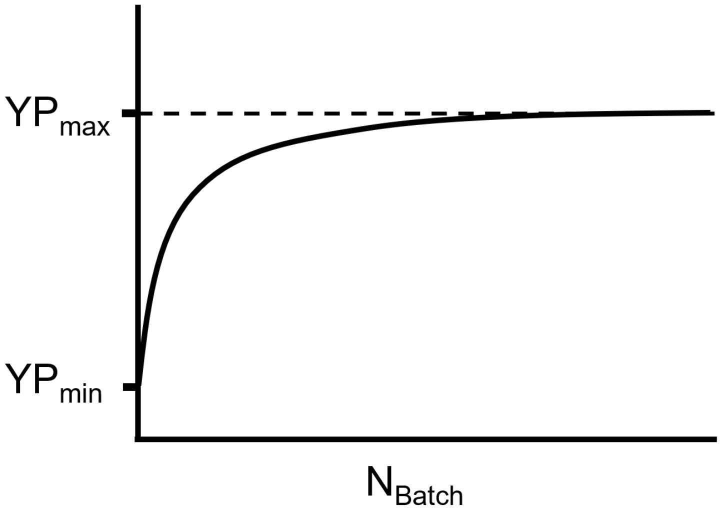
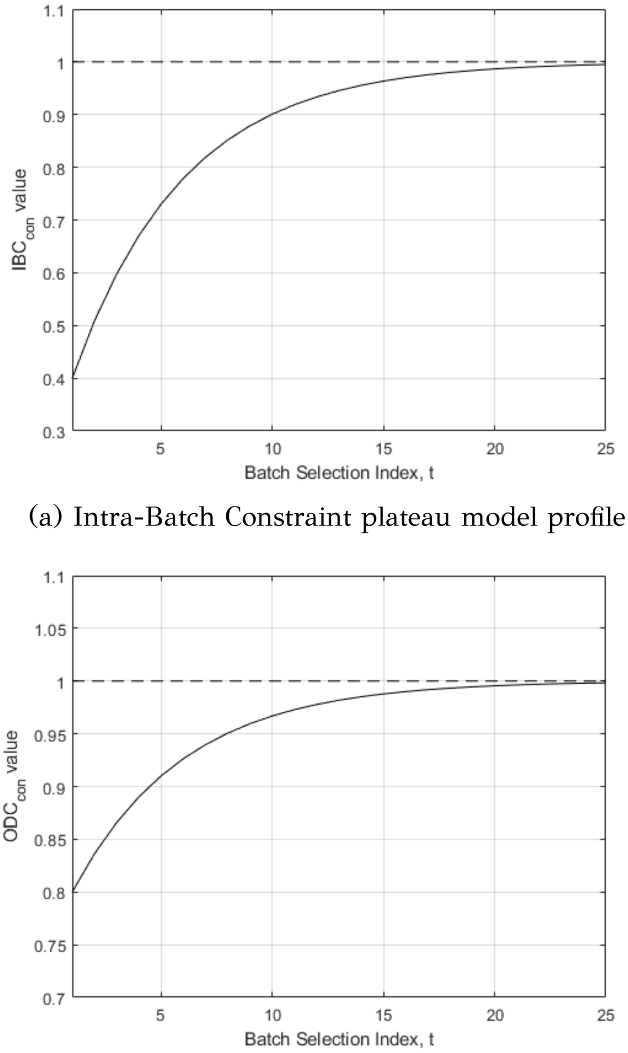


Figure 5.4: Time-varying plateau model profile for use within the NGB-GP-CB constraints.

As the Plateau constraint min and max values have been set within a [0,1] standard range, the GP σ_f^2 is used to scale the constraint to the equivalent scale for covariance comparison.

Whilst the rate of convergence for the plateau functions is ideal for both over-exploitation constraints, the intra-batch constraint should be stricter initially to encourage exploration. Therefore, to differentiate the application of both constraints a smaller minimum starting value was chosen for the intra-batch constraint. Figure 5.5 illustrates the plateau profiles of both constraints with Figure 5.5a for intra-batch constraint and Figure 5.5b for observed data constraint.

The only significant difference that can be observed between the two constraints as shown between Figure 5.5a and Figure 5.5b is the difference in the



(a) Intra-Batch Constraint plateau model profile

(b) Observed Data Constraint plateau model profile

Figure 5.5: BB-DoE constraint function time-varying constraint profiles

initial starting values for both constraints. For the intra-batch constraint, the initial value is significantly smaller at 0.4, indicating a much stricter constraint. This is to encourage early exploration of the search space more thoroughly to produce more well refined model and increase the chances of locating the global optima. The observed data constraint has an initial value which is significantly less strict at 0.8, as the purpose of this constraint is to prevent early over-exploitation and lower the likelihood of the optimiser being stuck in a local optimum.

5.2.5 Batch GA Encoding

The final stage for updating the B-DoE into a Non-Greedy BB-DoE is to make modifications to the GA optimiser in order to handle the selection of batch sets in contrast to individual experiments. Whereby, previously in Section 3.3.1, an individual encodes all input variables that constitute an experiment.

To extend this for use on a Non-Greedy BB-DoE problem, a batch individual (BC) is designed which is a concatenated series of individuals that constitute a batch set. Therefore, in each batch individual BC encoding will be a vector of each of the batch candidate's concatenated, for which each batch candidate individual is comprised of their input variable combinations. Thus, each (BC) individual in the GA population will then correspond to a batch combination set as exemplified in Figure 5.6.

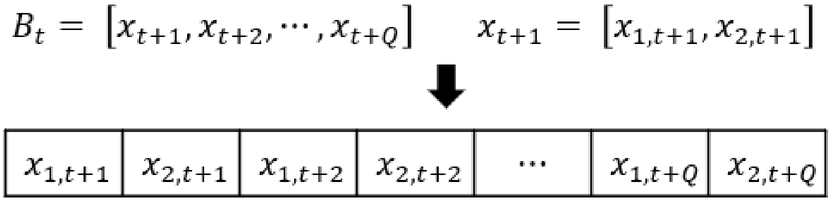

$B_t = [x_{t+1}, x_{t+2}, \cdots, x_{t+Q}]$
$x_{t+1} = [x_{1,t+1}, x_{2,t+1}]$
$x_{1,t+1}$
$x_{2,t+1}$
$x_{1,t+2}$
$x_{2,t+2}$
$\cdots$
$x_{1,t+Q}$
$x_{2,t+Q}$

Figure 5.6: Batch individual (BC) encoding for use in a GA population to be supplied to the BB-DoE. BC is comprised of batch set members concatenated in series where each gene of the BC corresponds to $x_{i,t+j}$, wherein i is the variable number index and j is the batch candidate number.

In Figure 5.6 each gene in the BC corresponds to a input variable $x_{I,t+j}$ indexed by the variable number i and batch candidate number j. Hence the 2nd variable for the 3rd batch set member would be $x_{2,t+3}$ and appear on the 6th gene if there were only 2 input variables per individual. In order to avoid repetitive sample points from occurring within the batch encoding, this is handled during the optimisation through the use of the IBC and ODC constraints introduced in Section 5.2.4. Wherein, repetitive sampling is discouraged within a batch set and at previously chosen experiments by using constraints which severely reduce to acquisition value of a BC.

Subsequently, each process involving the NGB-GP-CB optimisation procedure can be collated together into a pseudo-code representation as shown in Algorithm 5.1 which will be utilised as part of the whole in the overall BB-DoE framework as shown in Algorithm 5.2

Algorithm 5.1 Acquisition Function: Non-Greedy Batch Gaussian Process Confidence Bound

```
procedure NGB-GP-CB$\left(\mathrm{BC}, \mathrm{D_t}, \mathrm{GP}(\mathrm{m}, \mathrm{k}|\theta_\mathrm{t}), \mathbf{o}\right)$ do
    $\beta_\mathrm{t} \leftarrow$ Trade-off Parameter $(\mathbf{o})$                // See Equation 5.7
    for $\mathrm{i} = 1, 2, \ldots, \mathrm{P_{Size}}$ do
        $\mathrm{C_{1:Q}} \leftarrow$ Decode Batch Encoding$\left(\mathrm{BC_i}, \mathbf{o}\right)$
        $\mu_{\mathrm{t+1}}\left(\mathrm{C_{1:Q}}\right) \leftarrow$ Predict Mean$\left(\mathrm{C_{1:Q}}, \mathrm{D_t}, \mathrm{GP}(\mathrm{m}, \mathrm{k}|\theta_\mathrm{t})\right)$
        for $\mathrm{j} = 1, 2, \ldots, \mathrm{Q}$ do
            $\left[\mathrm{C_j}, \mathrm{C_{-j}}\right] \leftarrow \mathrm{C_{1:Q}}$
            $\bar{\mathrm{D}}_{\mathrm{t+Q-1}} \leftarrow \left[\mathrm{D_t}, \left\{\mathrm{C_{-j}}, \mu_{\mathrm{t+1}}\left(\mathrm{C_{-j}}\right)\right\}\right]$      // See Section 5.2.1
            $\sigma_{\mathrm{t+1}}\left(\mathrm{C_j} \middle| \mathrm{C_{-j}}\right) \leftarrow \sqrt{\text{Predict Var}\left(\mathrm{C_j}, \bar{\mathrm{D}}_{\mathrm{t+Q-1}}, \mathrm{GP}(\mathrm{m}, \mathrm{k}|\theta_\mathrm{t})\right)}$
            if $\mathbf{o}$ = *Max* then
                $\mathrm{u_{NGB-GP-CB}}\left(\mathrm{C_j}\right) \leftarrow \mu_{\mathrm{t+1}}\left(\mathrm{C_j}\right) + \sqrt{\beta_\mathrm{t}} * \sigma_{\mathrm{t+1}}\left(\mathrm{C_j} \middle| \mathrm{C_{-j}}\right)$
            else if $\mathbf{o}$ = *Min* then
                $\mathrm{u_{NGB-GP-CB}}\left(\mathrm{C_j}\right) \leftarrow \mu_{\mathrm{t+1}}\left(\mathrm{C_j}\right) - \sqrt{\beta_\mathrm{t}} * \sigma_{\mathrm{t+1}}\left(\mathrm{C_j} \middle| \mathrm{C_{-j}}\right)$
            end if
        end for
        $\mathrm{u_{NGB-GP-CB}}\left(\mathrm{BC_i}\right) \leftarrow \sum_{\mathrm{j=1}}^{\mathrm{Q}} \mathrm{u_{NGB-GP-CB}}\left(\mathrm{C_j}\right)$
    end for
    return $\mathrm{P_G}\left(\mathrm{u_{NGB-GP-CB}}\right)$
end procedure
```

5.2.6 Stopping Criterion

Whilst BO is a sample-efficient optimisation approach, its evaluation budget (T) must be defined in advance as once fully utilised the optimisation is terminated. However, in the event of the optimum being located and the model being well defined, any excess experiments may incur increased experimentation cost for little improvement. Thus, a stopping criterion provides the BO algorithm the chance to stop early to prevent over-experimentation and reduce the potential experimentation costs incurred.

Despite the potential benefits of integrating an early stopping criterion into BO algorithms, the research literature on this topic is relatively underdeveloped with only a few approaches available described hereafter. Most frequently the BO algorithm is run until the evaluation budget has been completely utilised. Another simple approach is to stop the BO algorithm once no better solution

Algorithm 5.2 BB-DoE Algorithm

```
Inputs:
        $n_s$      : Number of LHS samples
        $\lambda$  : Model Hyper-parameter priors
        $\mathbf{o}$ : Vector of optimisation settings
        $Q$        : Batch Size
        $N_B$      : Number of Batches (experimentation budget)

 1: $x_t \leftarrow$ Latin Hypercube Sampling$(n_s, d)$
 2: $y_t \leftarrow$ Evaluate$(x_t)$
 3: $D_t \leftarrow \left[x_t, y_t\right]$
 4: $\theta_t \leftarrow \mathrm{GPR}\left(\lambda, D_t\right)$                          // See Algorithm 3.2
 5: for $t = 1, 2, \ldots, N_B$ do
 6:     $B_{t+Q} \leftarrow$ BB-DoE GA Optimiser$\left(D_t, \mathrm{GP}(m, k|\theta_t), \mathbf{o}\right)$     // See line 11
 7:     $y_{t+1:t+Q} \leftarrow$ Evaluate$\left(B_{t+Q}\right)$
 8:     $D_{t+1} \leftarrow D_t \cup \left[x_{t+Q}, y_{t+Q}\right]$
 9:     $\theta_{t+1} \leftarrow \mathrm{GPR}\left(\lambda, D_{t+1}\right)$                  // See Algorithm 3.2
10: end for

11: procedure BB-DoE GA Optimiser$\left(D_t, \mathrm{GP}(m, k|\theta_t), \mathbf{o}\right)$ do
12:     Initialise GA Population $\left(P_1 = [C]\right)$
13:     for $G = 1, 2, \ldots, N_{Gen}$ do
14:         $u_{new} \leftarrow u_{NGB-GP-CB}\left(BC_G, D_t, \mathrm{GP}(m, k|\theta_t), \mathbf{o}\right)$     // See Algorithm 5.1
15:         $P_G \leftarrow$ Update Population $\left[C_G, u_{new}\right]$
16:         if $G = 1$ then
17:             $E \leftarrow \max_{i=1}^{E_{size}} P_G$
18:         else
19:             $E_{New} \leftarrow$ Elite Set Updater$(P_G, E, E_{size})$              // See Algorithm 3.1
20:         end if
21:         $P_G \leftarrow$ Constraint Evaluation$\left(P_G\right)$                    // See Section 5.2.4
22:         $Par_G \leftarrow$ Tournament Selection$\left(P_G\right)$                  // See Section 3.3.2
23:         $O_G \leftarrow$ Blending Crossover$\left(Par_G\right)$                    // See Section 3.3.3
24:         $M_G \leftarrow$ Random Mutation$\left(O_G\right)$                         // See Section 3.3.4
25:         $C_{G+1} \leftarrow O_G \cup M_G$
26:     end for
27:     return $B_{t+1} \leftarrow \arg\max_C u\,(P, E)$
28: end procedure
```

is found to improve over the current incumbent solution after a set threshold of iterations [152].

One of the first types of early stopping criterion introduced in literature assesses the expected change in acquisition costs of improvement-based acquisition functions: PI [147] and EI [171]. Whereby, for improvement-based acquisition functions as the number of evaluations increase the acquisition values of PI/EI tend to decrease [171]. Therefore, a threshold can be determined in order to monitor the acquisition costs of chosen experiments, such that when their acquisition value falls below the threshold the BO can be stopped early, as shown in Equation 5.11.

$$\text{Continue ?} = \begin{cases} \text{yes}, & \text{if} \quad u_{EI} > \kappa \\ \text{no}, & \text{if} \quad u_{EI} \leq \kappa \end{cases} \tag{5.11}$$

Where, κ is the stopping criterion threshold parameter. More recently Makarova et al. [152] proposed a stopping criterion which constructed a high probability confidence bound on the simple regret which cause the early termination of the BO when it crossed the threshold being less than the standard deviation of the estimate, as shown in Equation 5.12.

$$R_t < \sqrt{\sigma^2_{t+1}\left(y^+_t\right)} \tag{5.12}$$

Where, $\sigma^2_{t+1}\left(x^+_t\right)$ is the predictive variance at the incumbent or best observed input in the optimisation seen up until iteration (t). Although these early stopping criteria have beneficial properties, they are not suitable for application in this work. The use of BO until there is no large improvement over the incumbent solution is highly dependent upon the user's specification of the threshold. Similarly, in the stopping criterion proposed by Lorenz et al. [147] and Nguyen et al. [171], performance is highly dependent on the selection of the threshold which may not be intuitive. Makarova et al. [152] proposed a well-designed approach with theoretical bounds, which assumes that the GP surrogate model is well calibrated. Wherein in this work, the GP surrogate model is not assumed to be well calibrated, and another goal of the BO algorithm is to produce a well-tuned model.

Considering the conflicts between the desired goals of this work and literature approaches, a new stopping criterion is proposed. The aim of this stopping criterion is to assess the model prediction performance as the optimisation proceeds and terminates once the model is well calibrated over a period of multiple iterations. The premise behind this criterion is that it evaluates the current exploration of the design space such that it terminates the BO once the GP surrogate model is well calibrated. On the other hand, previous criterions evaluated the current exploitation of the design space in which termination occurred once global optimum was located, or the current optimum did not see significant improvement.

The stopping criterion evaluates the change in predictions of outputs for a testing set between the previous iteration step $(t-1)$ and the current iteration step (t) by modifying Equation 3.4 into Equation 5.13.

$$\rho_t = \frac{1}{\psi}\sum_{i=1}^{\psi}\left(\mu_{t-1}\left(x_i\right)-\mu_t\left(x_i\right)\right)^2 \tag{5.13}$$

Where, ρ_t is the stopping criterion for iteration step (t), ψ is the total number of stopping criterion grid points sampled using a Sobol sequence. $\mu_{t-1}\left(x_{1:\psi}\right)$ are the predicted means at the stopping criterion grid points for the previous iteration step, and $\mu_{t-1}\left(x_{1:\psi}\right)$ are the predicted means at the stopping criterion grid points for the current iteration step.

Whereby, as the GP surrogate model prediction is re-tuned and calibrated alongside the BO the prediction accuracy of each point in the design space will improve, and thus the deviation in predictions between each iteration step (t) will also decrease. As such when the deviation (ρ_t) becomes significantly small, and a threshold (κ) is met, the BO will be terminated.

The termination strategy used in this stopping criterion uses a two-stage approach.

1. The BO is eligible for termination once the deviation is below a chosen threshold.

2. The deviation remains below the threshold for Ψ of iterations.

The purpose of introducing a secondary check to the stopping criterion is to ensure that the BO is not terminated prematurely due to momentary consistency in predictions. This situation may occur in the event of the BO being caught in a local optimum. Combining the deviation metric in Equation 5.13 with the termination strategy produces the proposed stopping criterion demonstrated in Equation 5.15

$$\Psi_t = \begin{cases} 1, & \text{if} \quad \rho_t \le \kappa \\ 0, & \text{if} \quad \rho_t > \kappa \end{cases} \qquad \Psi_{t:\Psi_T} = \left[\Psi_t, \Psi_{t-1}, \ldots, \Psi_{t-\Psi_T}\right] \tag{5.14}$$

$$\text{Stop} = \begin{cases} \text{Yes} & \text{if} \quad \sum_{t=1}^{\Psi_T} \Psi_t = \Psi_T \\ \text{No} & \text{if} \quad \sum_{t=1}^{\Psi_T} \Psi_t < \Psi_T \end{cases} \tag{5.15}$$

Where, Ψ_t is the stopping counter for iteration t, and Ψ_T is the total number of iterations the stopping criterion must remain below the threshold. Whereby, the stopping counter only stores the threshold evaluations for the Ψ_T assessments, which acts as a moving window. By implementing the stopping criterion in this manner only in the event in which the stopping criterion is below the threshold successfully for Ψ_T iterations will the algorithm stop.

However, by introducing the Ψ_T parameter it increases the total number of parameters to tune along with the threshold for deviation which increases the complexity of implementation, but these can be examined during benchmark assessments.

5.3 Simulation Study: Benchmark Problems

5.3.1 Introduction

In this chapter in order to assess the performance of the developed NGB-GP-CB function and the modifications made to the B-DoE algorithm to produce the BB-DoE framework, a diverse set of synthetic benchmark functions will be used to assess both its exploration and exploitation characteristics. The synthetic benchmark problems to be assessed will be the same as those used in Section 4.3: Branin-Hoo [38], Mixture of Cosines [6], and Hartman [99].

Additionally, the stopping criterion introduced in Section 5.2.6 will be evaluated alongside: NRMSD and Regret metrics. Through its use on the synthetic benchmark functions we will be able to determine a suitable threshold (κ). This will assist in the batch manufacturing case study discussed later in this chapter as well as work in Chapter 6.

As discussed in Section 2.3.1 currently in literature there exists two types of BB-DoE methodologies, Greedy and Non-Greedy. The goal of Chapter 5 is not to simply develop a Non-Greedy BB-DoE approach and assess its performance on a series of benchmark functions but also compare its performance against current state of the art algorithms available within the BB-DoE literature.

Therefore, in order to assess BB-DoEs performance with regard to current state of the art algorithms both Greedy and Non-Greedy, two comparison methods were also assessed alongside NGB-GP-CB:

1. Greedy BB-DoE: Gaussian Process Batch Upper Confidence Bound (GP-BUCB)
2. Non-Greedy BB-DoE: Parallel Predictive Entropy Search (PPES)

Gaussian Process Batch Upper Confidence Bound (GP-BUCB) was chosen for the greedy selection scheme due to its similarities with NGB-GP-CB, wherein it uses the GP-CB as the core acquisition function. Another similarity is GP-BUCB was one of the 1st algorithms to leverage the concept of conditioning experiments on unexplored experiments to provide an updated predictive variance to modify the GP-CB for batch selection, as discussed in Section 5.2.1.

Parallel Predictive Entropy Search (PPES) was chosen for the non-greedy selection scheme as it is one of the few literature approaches that utilises a non-greedy selection scheme. However, there was no publicly available code for PPES and thus in order to compare performance the results will be contrasted against those published in [192].

5.3.2 Experiment Software

Algorithm 5.2 and the majority of functions have been self-coded in MATLAB 2017b in Windows 10 using their respective literature sources unless otherwise stated. The code was developed using MATLAB 2017b's parallel computing toolbox to parallelise the Algorithm 5.2 to be run with multiple repeats of implementations in parallel on ShARC in a private High Performance Computing (HPC) cluster for the Automatic Control and Systems Engineering department at the University of Sheffield. The private cluster has access to 2 worker nodes each with access to 28 cores and 384 GB of RAM of which 20 cores and 20 GB were used to run Algorithm 5.2 according to the details specified in Table ??.

The following functions were taken from existing libraries in public repositories rather than being self-coded: Branin-Hoo test function [209], Mixture of Cosines test function [209], Hartman-4 test function [209].

5.3.3 Experiment Details

Due to a lack of publicly available repository for the PPES code, in order to compare and contrast the performance of NGB-GP-CB in a BB-DoE algorithm against PPES. The same experimental conditions that were set according to [192] will also be used as specified in Table 5.1 for Q, N_{batch} and T.

Table 5.1: Batch synthetic benchmark experimentation details and settings for assessment of NGB-GP-CB.

Benchmark Study Settings		
Parameter	Symbol	Value
Experimentation Budget	T	50
Number of Latin Hypercube Samples	n_s	5
Number of Batches	N_{batch}	15
Batch Size	Q	3
Number of Repeats		50
Length-Scale Prior	λ_{LS}	$\mathrm{InvGamma}\,(5, 5)$
Output-Variance Prior	λ_{OV}	$\mathrm{HalfNorm}\,(0, 1)$
Noise Prior	λ_N	$\mathrm{Uniform}\left(10^{-6}, 10^{-3}\right)$
BB-DoE/GA Settings		
Parameter	Symbol	Value
GP-CB β_t Tuning Parameter	δ	0.1
Intra-Batch Constraint minimum	YP_{min}	0.4
Intra-Batch Constraint maximum	YP_{max}	1
Observed-Data Constraint minimum	YP_{min}	0.8
Observed-Data Constraint maximum	YP_{max}	1
Death Penalty	$D_{penalty}$	0
Severe Reduction Penalty	$S_{penalty}$	0.25

In PPES [192], their implementation on benchmarks: Branin-Hoo and Mixture of Cosines utilised a batch set size of Q = 3 for 15 batches, thus in order to compare performance of NGB-GP-CB and GP-BUCB against PPES these conditions were replicated in this investigation. By utilising these settings alongside an initial set of 5 experiments the total experimentation budget equalled 50 experiments. As this is also comparable to the settings set out in Table 4.1 which would allow for comparison between batch selection methods with sequential selection methods upon further analysis.

The settings for the GP surrogate model hyper-priors were also kept consistent with Table 4.1 and the GP-BUCB's tuning parameter δ will also be set according to [59]. Finally, the BB-DoE algorithm settings were developed and discussed in Section 5.2 prior to collation into Table 5.1.

5.3.4 Results and Discussion

The results are broken down and separated into each of the synthetic benchmark problems tackled whose details are described in Section 5.3, with figures illustrating the performance on: NRMSD (exploration), Regret (exploitation) and stopping criterion. Figures 5.7, 5.8, and 5.9 represent the Branin-Hoo test function [38]. Figures 5.10, 5.11, and 5.12 represent the Mixture of Cosines test function [6]. Figures 5.13, 5.14, and 5.15 represent the Hartmann test function [99].

Benchmark Results: Branin-Hoo (2D)

In Figure 5.7, it can be seen that the non-greedy BB-DoE algorithm using NGB-GP-CB was capable of achieving a much lower NRMSD at every stage of the optimisation budget (T) reaching a NRMSD of < 2 %. Whilst on the other hand the greedy BB-DoE algorithm using GP-BUCB surrogate models accuracy was three times greater, plateauing at an NRMSD of < 6.5 %. Thus, already demonstrating the performance improvement in regard to exploration between greedy GP-BUCB and non-greedy NGB-GP-CB.

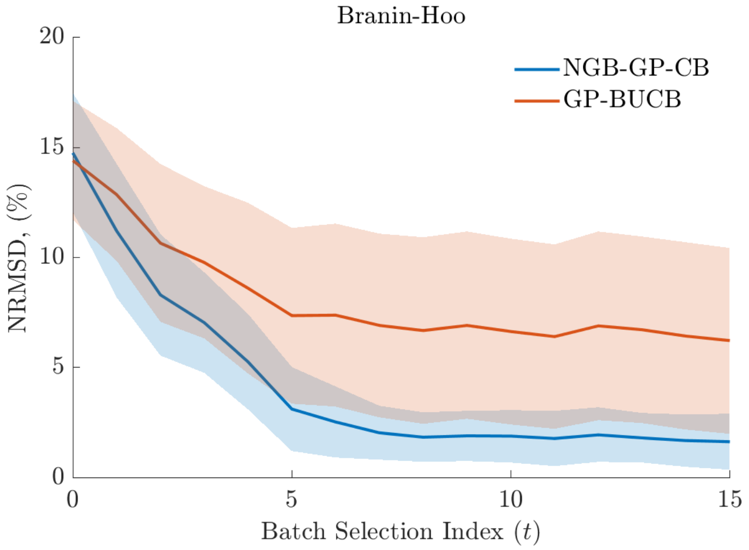


Figure 5.7: Comparison of NRMSD performance metric between a Greedy BB-DoE (GP-BUCB) and Non-Greedy BB-DoE (NGB-GP-CB) approach on the Branin-Hoo benchmark function. The BB-DoE algorithm was run according to the settings in Table 5.1 and plotted the mean of 50 repeats for the NRMSD. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the NRMSD.

In contrast to the improved performance with regard to exploration, GP-BUCB has an improved performance when assessing the Regret over the experimentation budget as shown in Figure 5.8. Whereby, GP-BUCB approximately approaches a Regret (< 0.1) after 6 batches in comparison to NGB-GP-CB which achieves the same level of Regret after 14 batches. In Shah and Ghahramani [192] the PPES achieves a similar rate of convergence as

NGB-GP-CB with both having a Regret of ($R_7 \approx 1$) by the 7^{th} batch.

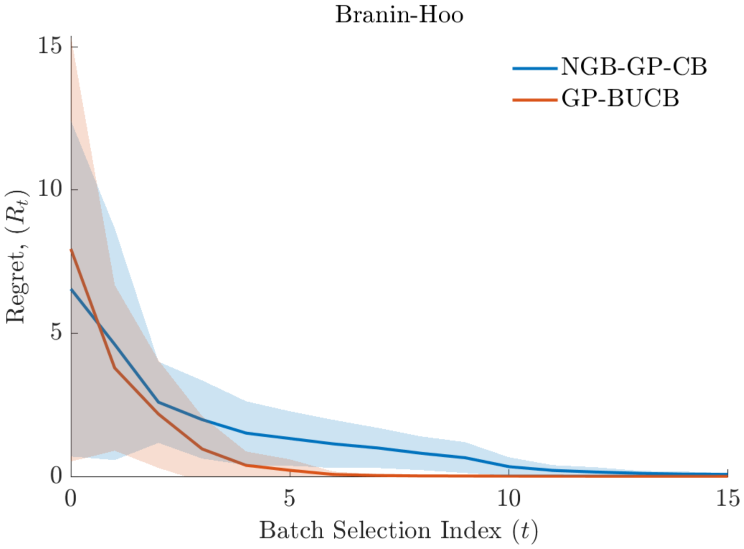


Figure 5.8: Comparison of Regret performance metric between a Greedy BB-DoE (GP-BUCB) and Non-Greedy BB-DoE (NGB-GP-CB) approach on the Branin-Hoo benchmark function. The BB-DoE algorithm was run according to the settings in Table 5.1 and plotted the mean of 50 repeats for the Regret. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the Regret.

Finally, the stopping criterion can be seen to steadily decrease with each successive batch for both BB-DoE approaches, with the stopping criterion plateauing at approximately 5x10^{-3} for GP-BUCB. Whilst NGB-GP-CB continued to decrease until reaching 1x10^{-4} by the end of the experimentation budget. It can be seen when comparing Figure's 5.7/ 5.9 the GP-BUCB stopping criterion plateau's when the NRMSD also plateau's thus preventing excessive experimentation when there is no further modelling improvement.

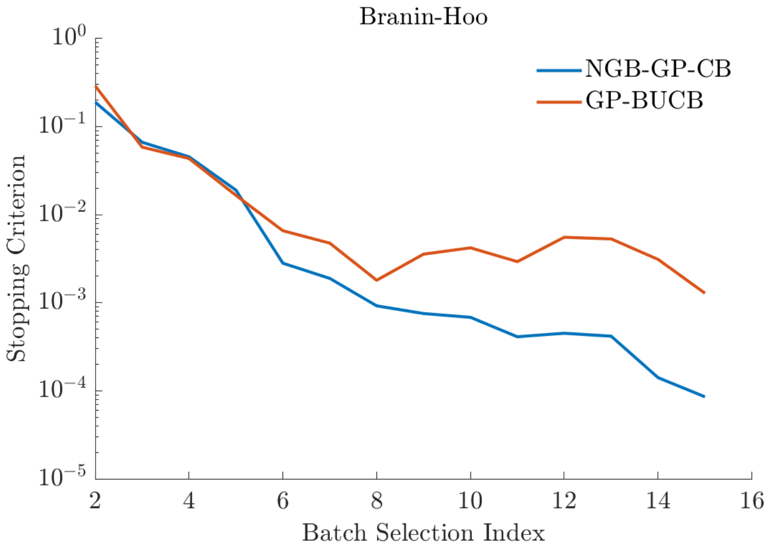


Figure 5.9: Comparison of Stopping criterion between a Greedy BB-DoE (GP-BUCB) and Non-Greedy BB-DoE (NGB-GP-CB) approach on the Branin-Hoo benchmark function. The BB-DoE algorithm was run according to the settings in Table 5.1 and plotted the median of 50 repeats for the Stopping criterion.

Benchmark Results: Mixture of Cosines (2D)

In Figure 5.10 as seen previously in Figure 5.7 a similar pattern of performance with regard to NRMSD is observed with GP-BUCB having a slow improvement in NRMSD and NGB-GP-CB having significant improvement in NRSMD over the experimentation budget. Wherein GP-BUCB slowly converges to an NRMSD of 12.8 % by the end of the experimentation budget.

However, in regard to NRMSD both BB-DoE methods depsite being applied onto another 2-Dimensional problem, the NRMSD is higher than when applied onto the Branin-Hoo problem previously. This is most likely due to differences in optimisation landscape and the total number of global optimum's. Whereby, in Branin-Hoo there are 3 global optima potentially encouraging greater exploration in comparison to Mixture of cosines with many local optimums and only one global optimum, suggesting a lower exploration pressure within the design space. Also NGB-GP-CB achieved a NRMSD of < 7 % by the end of the experimentation budget, which is an improvement over the performance observed in Figure 4.4 for the sequential GP-CB.

In Figure 5.11 both BB-DoE methodologies: greedy and non-greedy achieved a similar Regret profile over the experimentation budget as previously seen in Figure 4.5. Although NGB-GP-CB achieved a Regret of $R_8 < 0.01$ by the 8^{th} batch, slightly outperforming the GP-BUCB scheme reaching a Regret of

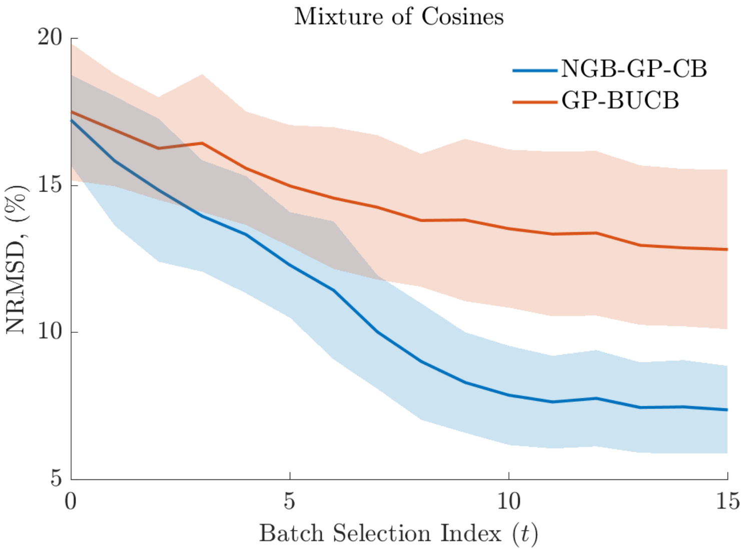


Figure 5.10: Comparison of NRMSD performance metric between a Greedy BB-DoE (GP-BUCB) and Non-Greedy BB-DoE (NGB-GP-CB) approach on the Mixture of Cosines benchmark function. The BB-DoE algorithm was run according to the settings in Table 5.1 and plotted the mean of 50 repeats for the NRMSD. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the NRMSD.

$R_8 < 0.05$ by the same batch. Again in Shah and Ghahramani [192] the PPES appears to achieve a similar level of performance to NGB-GP-CB obtaining a Regret of $R_8 < 0.01$ by the 8th batch.

Finally, the stopping criterion followed a simply pattern to its application upon the Branin-Hoo synthetic benchmark previously, in which both BB-DoE approaches stopping criterion value decreased at roughly the same rate until plateau's occurred in the NRMSD. Whereby, the GP-BUCB plateau'd earlier with a greater NRMSD and subsequently the stopping criterion plateau'd earlier with a larger value of $5\text{x}10^{-3}$. Whilst on the other hand NGB-GP-CBs stopping criterion continued to decrease in value to $6\text{x}10^{-4}$, although its value did not plateau showing signs of potential for further experiments.

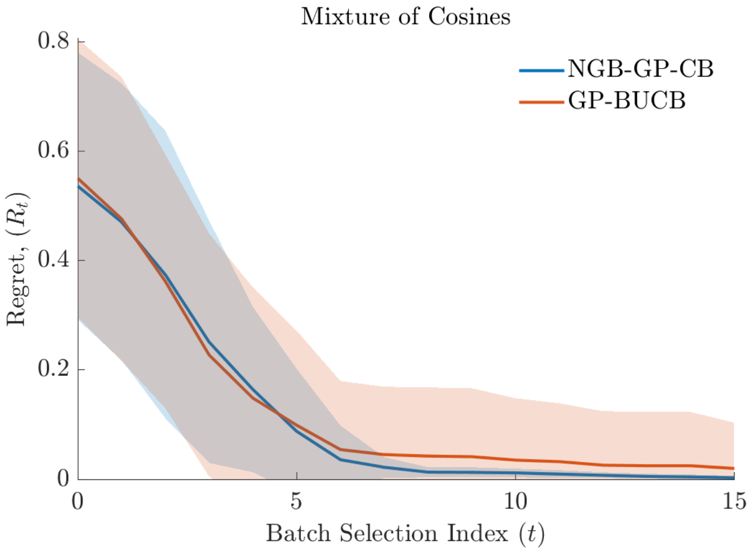


Figure 5.11: Comparison of Regret performance metric between a Greedy BB-DoE (GP-BUCB) and Non-Greedy BB-DoE (NGB-GP-CB) approach on the Mixture of Cosines benchmark function. The BB-DoE algorithm was run according to the settings in Table 5.1 and plotted the mean of 50 repeats for the Regret. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the Regret.

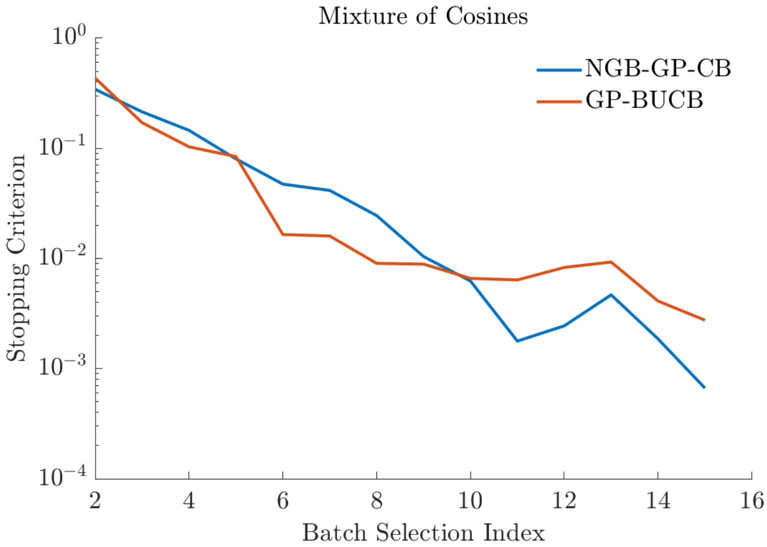


Figure 5.12: Comparison of Stopping criterion between a Greedy BB-DoE (GP-BUCB) and Non-Greedy BB-DoE (NGB-GP-CB) approach on the Mixture of Cosines benchmark function. The BB-DoE algorithm was run according to the settings in Table 5.1 and plotted the median of 50 repeats for the Stopping criterion.

Benchmark Results: Hartmann (4D)

The previous two benchmark functions were both 2-Dimensional problems, whilst the Hartman benchmark function is a 4-Dimensional problem that increases the optimisation difficulty. In Figure 5.13 the NRMSD improvement over the first 5 batches for both the GP-BUCB and the NGB-GP-CB improved at approximately the same rate. From the fifth batch to the full experimentation budget the non-greedy NGB-GP-CB function seemingly outperformed the greedy GP-BUCB approach achieving a NRMSD of 13 %. The NGB-GP-CB did improve upon the sequential B-DoE performance achieved in Figure 4.6 marginally, as well as outperforming the greedy GP-BUCB approach which achieved an NRMSD of 15.7 % after the full experimentation budget.

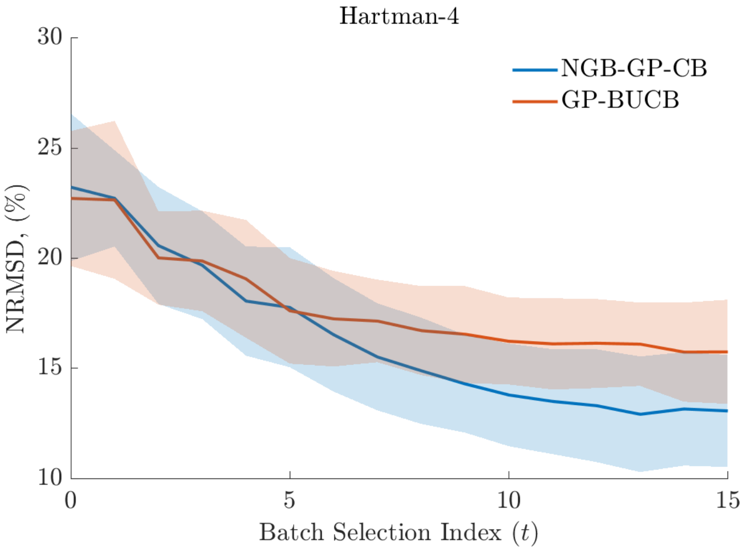


Figure 5.13: Comparison of NRMSD performance metric between a Greedy BB-DoE (GP-BUCB) and Non-Greedy BB-DoE (NGB-GP-CB) approach on the Hartmann benchmark function. The BB-DoE algorithm was run according to the settings in Table 5.1 and plotted the mean of 50 repeats for the NRMSD. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the NRMSD.

In Figure 5.14 the Regret profile over the experimentation budget performed as expected with the greedy BB-DoE scheme (GP-BUCB) achieving the desired Regret in as few experimental batches as possible. Wherein the optimum is located by the 8th batch with a Regret $R_8 < 0.1$. Whereas for the non-greedy BB-DoE approach a similar regret is achieved by the 12th batch, but as shown previously has more significant out performance in achieving a better surrogate model accuracy.

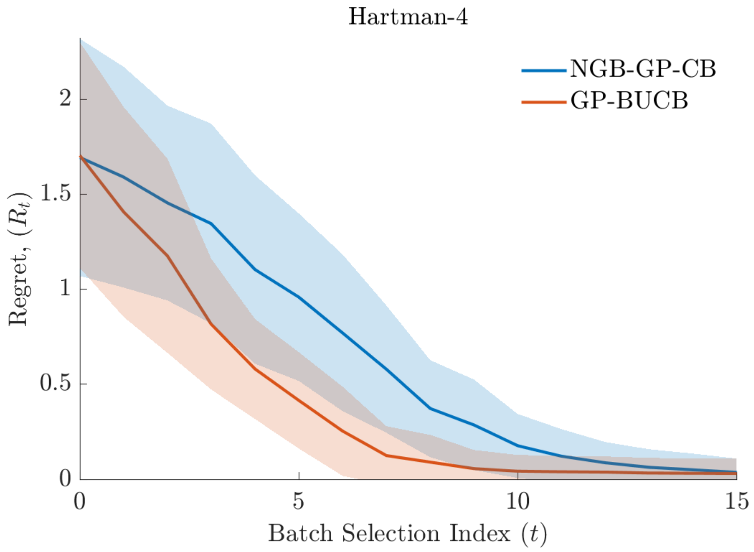


Figure 5.14: Comparison of Regret performance metric between a Greedy BB-DoE (GP-BUCB) and Non-Greedy BB-DoE (NGB-GP-CB) approach on the Hartmann benchmark function. The BB-DoE algorithm was run according to the settings in Table 5.1 and plotted the mean of 50 repeats for the Regret. The color shaded regions represent one standard deviation confidence bands around the mean taken from the 50 repeats for the Regret.

Finally, in Figure 5.15 the stopping criterion decreased at a similar rate for both the greedy and non-greedy BB-DoE approaches reaching a stopping criterion value $\approx$ 1x10^{-2}. However, in comparison to the stopping criterion values achieved on both the Branin-Hoo and Mixture of Cosines synthetic benchmark functions the Hartman function stopping criterion value is two order of magnitudes greater. This suggests that for the Hartman synthetic benchmark function that despite showing good Regret or exploitative performance, the BB-DoE approaches could have benefited from more batches for exploration and thus benefit from an increased experimentation budget.

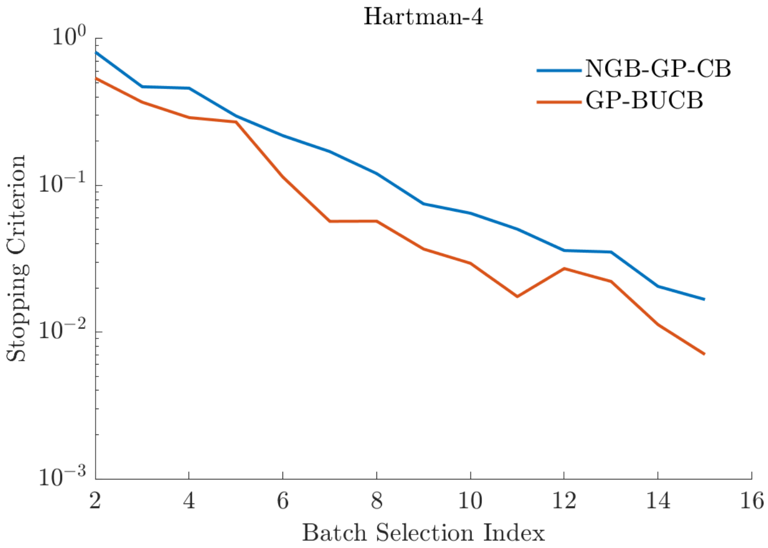


Figure 5.15: Comparison of Stopping criterion between a Greedy BB-DoE (GP-BUCB) and Non-Greedy BB-DoE (NGB-GP-CB) approach on the Hartmann benchmark function. The BB-DoE algorithm was run according to the settings in Table 5.1 and plotted the median of 50 repeats for the Stopping criterion.

5.4 Batch Manufacturing Case Study

5.4.1 Introduction

This investigation discusses the DoE potential of a BB-DoE algorithm on an additive manufacturing case. In Section 5.3, the algorithm was assessed on a variety of synthetic benchmark functions under ideal conditions, whereas in this investigation it will be stress-tested on a real-world expensive manufacturing case. The intended goal of this case study is to test the reproducibility of initial performance on benchmark functions as well as assess whether it can be replicated on real-world additive manufacturing applications.

This case study investigates the unknown mechanisms and interactions of the Directed Energy Deposition (DED) process parameters for their impact on the Dendritic Arm Spacing (DAS), which is linked to a variety of metallurgical properties. The process parameters to be assessed are Hatch Spacing (mm), Laser Power (W), and Nozzle Velocity (mm/min).

5.4.2 Background

A DED process is a powder-fed AM process which feeds powder directly into the path of a high-power laser to melt and form desired products, see

Figure 5.16. The laser is fed down the centre of the DED head and is focused using a set of lenses that can manipulate the laser to a small singular spot for focused melting [3]. The metal powder can be fed directly into the path of the centred laser from a variety of angle's depending upon the DED head design, to allow for greater control in the manufacturing process [3].

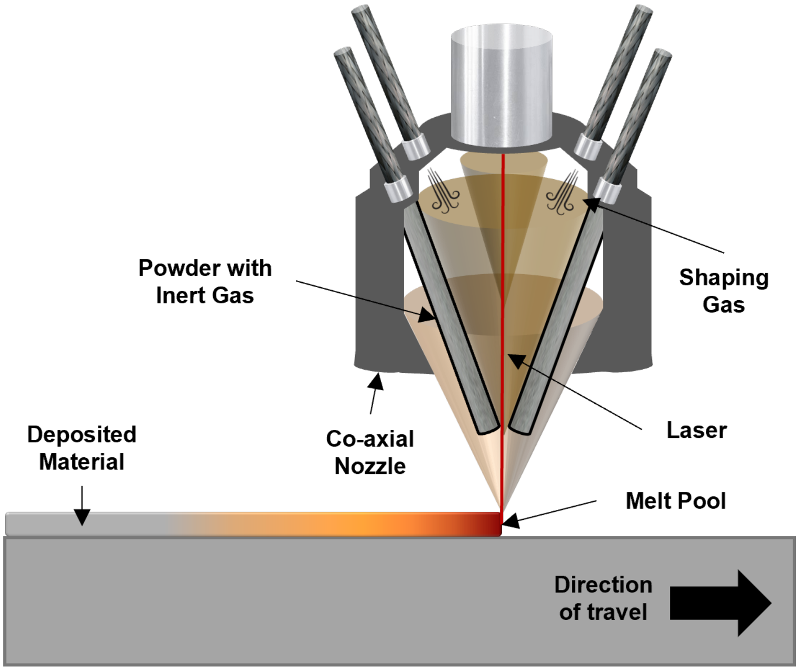


Figure 5.16: Illustrated representation depicting the manufacturing process using Directed Energy Deposition.

One property that provides insight into a wide variety of metallurgical properties is the DAS which can influence a material's mechanical, electrical, and chemical properties. DAS is a microstructural property that measures the spacing of crystalline grains called dendrites, these form during the solidification process in metals. Thus, by developing a model of the process parameter settings and their influence on the DAS, a manufacturer would be capable of designing a product with the whichever metallurgical properties were desired.

In this case study the DED manufacturer is building nickel super-alloy walls to analyse the process parameters interactions and their correlation towards a desired DAS setting. In previous studies of similar alloys, the DAS was identified to vary in a range of 2.3 μm to 9.0 μm, upon which preliminary experimentation of this alloy found DAS between 2.3 μm to 3.0 μm supporting the DAS specified range. Despite preliminary experimentation no experiments returned DAS values within a range of 4.5 μm to 9.0 μm, and therefore further exploration of this unknown region is also desired particularly in terms of manufacturability.

5.4.3 Aims and Objectives

The aims for this manufacturing case study can be split into two areas: the algorithmic assessment, and the improvement of process understanding for the manufacturers. The manufacturer aims to identify the process parameter settings that locate a desirable DAS of 4.5 μm or the median of the known alloy range, in order to achieve the desired metallurgical properties. With respect to the aims in algorithm assessment of the BB-DoE, the objectives of the case study are to:

1. Evaluate performance of BB-DoE algorithms on an AM DoE problem.
2. Assess the exploration vs exploitation capabilities of BB-DoE with a particular interest in the exploration of the unknown 4.5 μm to 9.0 μm range.

Equally important, the manufacturers aim is to identify the optimal process parameter settings to locate the desired DAS of 4.5 μm. Although an added complexity to this optimisation is that the 4.5 μm falls as a median value in the output range. Therefore, the objectives for the case study with respect to the manufacturer are to:

1. Builds well-tuned surrogate GP model of the DED process interactions effect on DAS.
2. Optimise the process parameter settings to locate the optimal DAS.
3. Minimise the total number of experiments required to locate the optimum.

5.4.4 Methodology

A unique characteristic to this manufacturing case study is that the desired optimum is located at the median of a range. Typically, in an optimisation problem the goal of the optimiser to isolate either a minimum or maximum for a particular objective function. However, in this case the desired goal is a median, and thus either the objective function or methodology must be modified in order to facilitate this goal.

A simple solution to this issue would be to transform the process output, such that the desired optimisation goal is converted to either a maximum or minimum value. This could be achieved by centring the modulus of the DAS.

$$\mathrm{DAS-D} = |\,\mathrm{DAS} - 4.5\,|$$

This conversion of DAS into DAS-D means the optimisation now seeks a minimum of 0. However, through the conversion to DAS-D it introduces new disadvantages that require modifications for implementation in the optimisation.

Firstly, by taking the modulus of the centred DAS, it is not possible to discern whether the DAS-D's original DAS existed above or below 4.5 μm. This is of importance as the DAS region of 4.5 μm to 9.0 μm is unexplored, therefore by not being able to evaluate which experiments are exploring this region, one objective of the case study would be unaccomplished. To alleviate these concerns, the GP surrogate models would be built on the original DAS and only when a DAS-D is required for the evaluation of the NGB-GP-CB acquisition function will the DAS be converted to DAS-D for optimisation.

Table 5.2: Directed Energy Deposition (DED) manufacturing process input variable constraints.

Input Variable Constraints			
Input Variable	Lower Limit, x^L	Upper Limit, x^U	Resolution
Hatch Spacing (mm)	0.3	0.7	0.01
Laser Power (W)	200	600	1
Nozzle Velocity (mm/min)	500	3000	1

Additionally, when dealing with real-world manufacturing DAS problems, additional constraints aside from defining the search space domain (See Table 5.2) may also be required. As the optimiser of choice is a GA, its capability in handling additional constraints can be tackled by the addition of another constraint function, whether it be death or severe punishment penalty function. For this manufacturing DoE case study, the additional constraints are to be placed upon the maximum and minimum allowable values of the DAS with a death penalty as shown in Equation 5.16.

$$\chi(\mathrm{BC}) = \chi(\mathrm{BC}) * \mathrm{D}_{\mathrm{penalty}} \tag{5.16}$$

$$\mathrm{D}_{\mathrm{penalty}} = \begin{cases} 0, & \text{if DAS} \geq 9.0\mu\text{m or DAS} \leq 2.3\mu\text{m} \\ 1, & \text{otherwise} \end{cases} \tag{5.17}$$

Next the BB-DoE optimiser settings required specification are as shown in Table 5.3. These settings were kept consistent with Table 5.1 for the BB-DoE optimiser settings but the experimentation budget for the AM case study was chosen to have a batch size of Q = 5 for 10 batches. Whilst the GA optimiser settings remained unchanged from the implementation in Table 4.1. To see the algorithms framework, refer to Algorithm 5.2.

Table 5.3: Batch Bayesian Experimental Design Optimisation (BB-DoE) settings.

BB-DoE Algorithm Settings		
BB-DoE Parameter	Algorithm Function	Setting
Batch Size	General	5
Number of Batches	General	10
IBC YP_{min}	Batch Constraint	0.4
IBC YP_{max}	Batch Constraint	1
ODC YP_{min}	Batch Constraint	0.8
ODC YP_{max}	Batch Constraint	1
Death Penalty	Constraint	0
Severe Reduction Penalty	Constraint	0.25

5.4.5 Experiment Software

Similarly to Section 5.3.2, the set-up of algorithms and functions used were in-line with the benchmark assessments but re-coded for use on the AM case study specifics as discussed in Section 5.4. Algorithm 5.2 and the majority of functions have been self-coded in MATLAB 2017b in Windows 10 using their respective literature sources unless otherwise stated. The code was run on a personal PC with the following specifications: AMD Ryzen 5 1600 6 core processor with 12 threads and 24 GB of RAM of which all were used to run Algorithm 5.2 according to the details specified in Table 5.3. In this study no existing libraries or repositories were used.

5.4.6 Results and Discussion

Prior to the implementation of the BB-DoE algorithm, the manufacturing client who desired to optimise the DED process performed a set of 15 preliminary experiments. These 15 experiments were carried using a traditional DoE scheme using 3 factors and 3 levels as set out in Table 5.4. Of these 15 experiments, a subset of five experiments were selected for use as the training data set to be reflective of the first uninformed batch.

Table 5.4: Preliminary experimentation performed by the DED manufacturer from which an initial data set will be retrieved to initialise the BB-DoE algorithm.

DED Manufacturing DAS, Manufacturer Selected				
Experiment Number	Hatch Spacing (mm)	Laser Power (W)	Nozzle Velocity (mm/min)	DAS (μm)
1	0.50	300	1500	2.5
2	0.50	550	1500	4.1
3	0.50	300	3000	1.8
4	0.50	550	3000	2.6
5	0.30	300	2250	2.7
6	0.30	550	2250	3.0
7	0.70	300	2250	2.9
8	0.70	550	2250	4.0
9	0.30	425	1500	n/a
10	0.30	425	3000	3.6
11	0.70	425	1500	3.0
12	0.70	425	3000	2.4
13	0.50	425	2250	3.5
14	0.50	425	1500	4.4
15	0.50	425	3000	3.0

From Table 5.4 experiments: 1, 6, 7, 12, and 15 were selected to ensure that each level of each factor were included at least once to ensure a wide spread of experiments in the design space. The training data set was used to tune the model hyper-parameters for the GP model by optimising the negative log likelihood in Equation 2.10 using K-Fold cross validation.

As can be seen in Table 5.4, despite the wide spread of data in the design space the traditional DoE scheme was unable to identify which process parameter settings would result in a desired DAS, although the experimenters did approach the optimum in experiment 14 with DAS = 4.4μm. Another shortcoming of the preliminary experiments was the lack of exploration of the $4.5 - 9.0\mu$m region which could not be located within the 15 preliminary experiments.

Table 5.5 details the experiment selection results from the BB-DoE algorithm applied according to the specification set out in Section 5.4.4 and optimiser settings in Table 5.3. From these results it can be seen that the BB-DoE algorithm significantly outperformed the preliminary DoE scheme, as within an identical 15 experiments (or 3 batches), the BB-DoE algorithm

Table 5.5: DAS obtained from DED manufacturing experiments guided by BB-DoE algorithm for an experimentation budget of 10 batches of 5 experiments, terminated early by decision maker at 6 batches.

DED Manufacturing DAS, BB-DoE selected						
Batch Number	Experiment Number	Hatch Spacing (mm)	Laser Power (W)	Nozzle Velocity (mm/min)	DAS (μm)	DAS-D (μm)
1	1	0.51	331	2163	2.3	2.2
1	2	0.40	411	2850	3.0	1.5
1	3	0.38	558	1922	3.6	0.9
1	4	0.54	505	2577	3.2	1.3
1	5	0.70	567	1439	3.6	0.9
2	1	0.48	461	530	5.6	1.1
2	2	0.40	570	2869	3.2	1.3
2	3	0.43	568	1149	4.7	0.2
2	4	0.67	583	2642	2.7	1.8
2	5	0.44	504	2495	2.7	1.8
3	1	0.32	278	2925	2.0	2.5
3	2	0.53	559	1186	4.0	0.5
3	3	0.46	314	591	4.5	0.0
3	4	0.35	379	512	6.9	2.4
3	5	0.45	491	1201	3.7	0.8
4	1	0.43	399	1583	2.8	1.7
4	2	0.58	597	882	4.8	0.3
4	3	0.49	550	583	6.4	1.9
4	4	0.65	407	1189	3.9	0.6
4	5	0.63	486	589	5.2	0.7
5	1	0.41	228	1117	3.3	1.2
5	2	0.31	265	1544	3.8	0.7
5	3	0.62	205	901	2.8	1.7
5	4	0.66	582	812	5.8	1.3
5	5	0.69	394	710	6.1	1.6
6	1	0.41	368	884	4.9	0.4
6	2	0.39	485	1179	5.5	1.0
6	3	0.33	589	1654	4.2	0.3
6	4	0.35	339	1362	3.4	1.1
6	5	0.61	565	2948	3.7	0.8

identified the optimal process parameter settings required to achieve a DAS of 4.5 μm highlighted in Table 5.5.

Also in the 2$^{\text{nd}}$ batch of experiments at experiment 1 and 3 the BB-DoE algorithm began to select experiments that were exploring the 4.5–9.0μm DAS region with DAS = 5.6 and 4.7μm respectively. Subsequently in each batch following the 2nd batch, at least two candidate experiments of each chosen batch set had experiments which explored the $4.5 - 9.0\mu$m DAS region.

Therefore, the results shown in Table 5.5 have shown significant improvements with regard to placing experiments in areas of the design space to determine the optimum and improve the exploration of the design space. However, as set out in Chapter 3 in order to determine exploration and exploitation of the DoE algorithms, Regret and NRMSD were to be utilised. However, NRMSD requires the true output y at a set of testing experiments to be known a priori in order to provide an estimate of the exploration performance.

Consequently, whilst a preliminary data set was available as shown in Table 5.4 which could have been utilised as the testing data set in the calculation of NRMSD it was not suitable for use. This is due to the fact that the experiments available di not provide a uniformly distributed sample set of the design space with regards to the outputs. Namely, no experiments were placed in the DAS region 4.5–9.0μm which would inhibit predictive performance evaluation if used as a testing set and was thus deemed unsuitable for use. Therefore, an alternative performance measure was required to assess the exploration performance of the BB-DoE algorithm. In order to assess the variation in model performance from one batch to the next a visual representation of the changes in predictive uncertainty and mean were generated using heat maps.

Heat Map Results

A heat map is a data visualisation approach that shows the magnitude and change of the performance measure throughout the design space as colour in 2-Dimensions. Therefore in order to be implemented, a 2-D grid is required as the DED manufacturing case would need for one of its three input variables to be fixed to a constant value. In order to decide which input variable was most suitable for setting as a constant for analysis, the DED machine's degree of variability was assessed using Table 5.2. Whereby, a DED machine only allows for 41 different variations in Hatch Spacing and so setting it to a constant value may have the lowest impact on the heat map analysis.

$$\text{Hatch Spacing}\,(\text{mm}) = [0.30, 0.40, 0.45, 0.50, 0.60]$$

The hatch spacing was set at five different settings with a finer grid closer to the optimal setting that located the desired DAS of 4.5μm in batch 3. The heat maps considered analysed the change in magnitude of the performance measures with each batch selected during 2 stages of the BB-DoE:

- Stage 1: Batches 0 to 3,

  Predictive Uncertainty.

  Predictive Mean with contours.

- Stage 2: Batches 4 to 6,

  Predictive Uncertainty.

  Predictive Mean with contours.

The heatmap colour scaling presented in Figures 5.17 and 5.19 for the predictive uncertainty is set in the interval $[0, 1]$, due to the tuning of the GP surrogate model. A low predictive uncertainty is illustrated as blue and high predictive uncertainty as red. Whilst in Figures 5.18 and 5.20 the heatmap colour scaling is set in the interval $[-3, 3]$, due to the standardisation of the DAS output which is centred around the mean of 3.8886μm. Therefore, the 4.5μm optimum would be placed in the white to red region of the contour plots in the hatch spacing column of 0.45mm.

Firstly, Figures 5.17 and 5.18 illustrate the changes in predictive uncertainty and mean respectively for batches 0 (training data) to 3 (optimum located). In batches 0 to 3 of the BB-DoE optimisation the predictive uncertainty is high due to a lack of exploration of the design space, but gradually reduces with each batch selected switching from red to white to blue. However, in Figure 5.17 the 3rd batch's predictive uncertainty seems to increase again, likely due to the re-tuning of the GP model hyper-parameters in response to new unexplored regions of the design space. Wherein, the 2nd and 3rd batch experiments at a lower Laser Power (200 - 400) range with a low Nozzle velocity (< 1500) began to explore the $4.5 - 9.0\mu$m DAS region.

The predictive uncertainty observed in 2nd batch of Figure 5.17 with a hatch spacing of 0.30 and 0.60 also seems to have higher level's of uncertainty towards the extremes of fixed hatch spacing settings. This suggests that these sections of design space which have a high degree of uncertainty are due to either a lack of exploration, or a high variability which may lead to increased exploration in later batches. Where from the 3rd to 6th batch 60% of experiments lay in the hatch spacing ranges $(0.3 > x > 0.4)$ and $(0.6 > x > 0.7)$ despite only representing 50% of the design space as shown in Table 5.5.

Correspondingly, a similar pattern emerges in Figure 5.18 wherein initially the predicted mean or response surface is undefined with only a few experiments, but with each batch of experiments the response surface begins to take

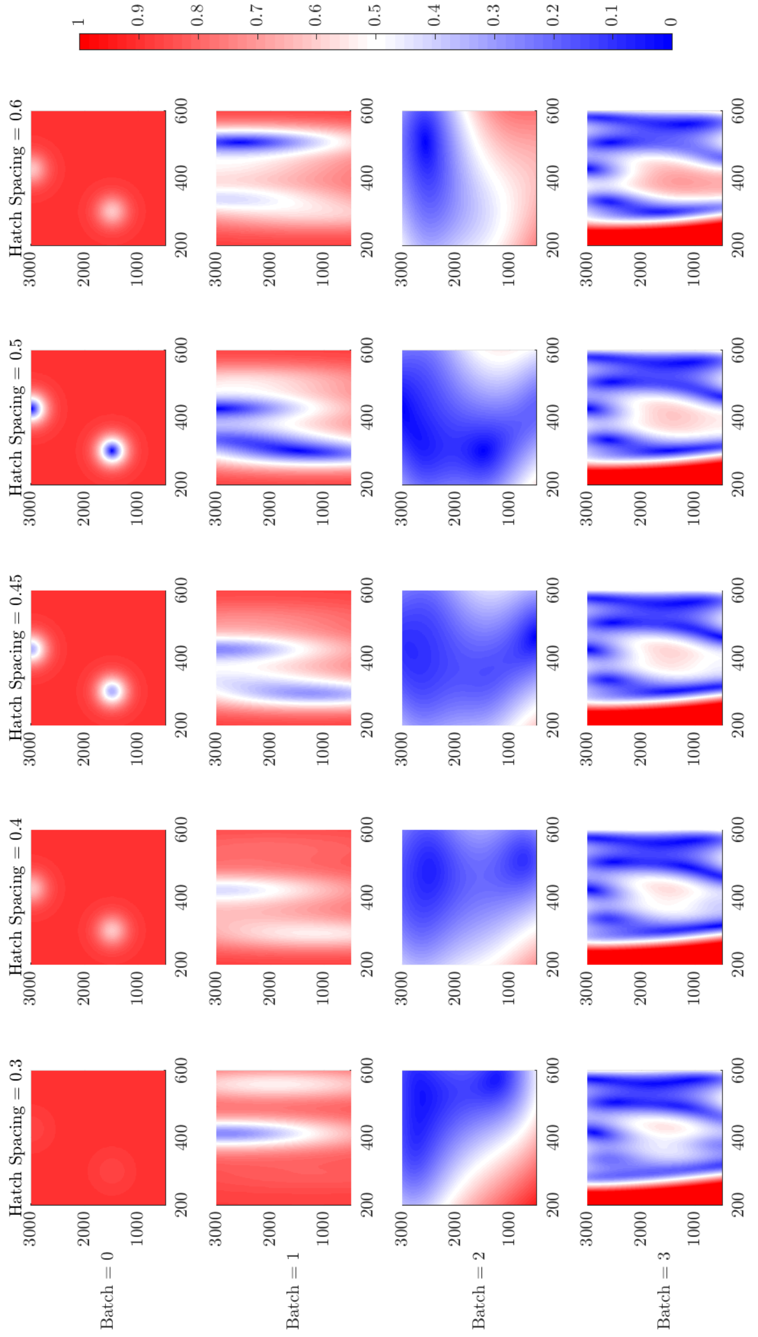


Figure 5.17: Evolution of predictive uncertainty in a 2-D grid representation of the design space at 5 fixed hatch spacing settings, $[0.30, 0.40, 0.45, 0.50, 0.60]$. The x-axis is the Laser Power (W) and the y-axis is the Nozzle Velocity (mm/min). The figure above assesses the early changes in predictive uncertainty between the training batch data to the 3$^{\text{rd}}$ batch in which the optimal process parameter settings for the DAS are determined.

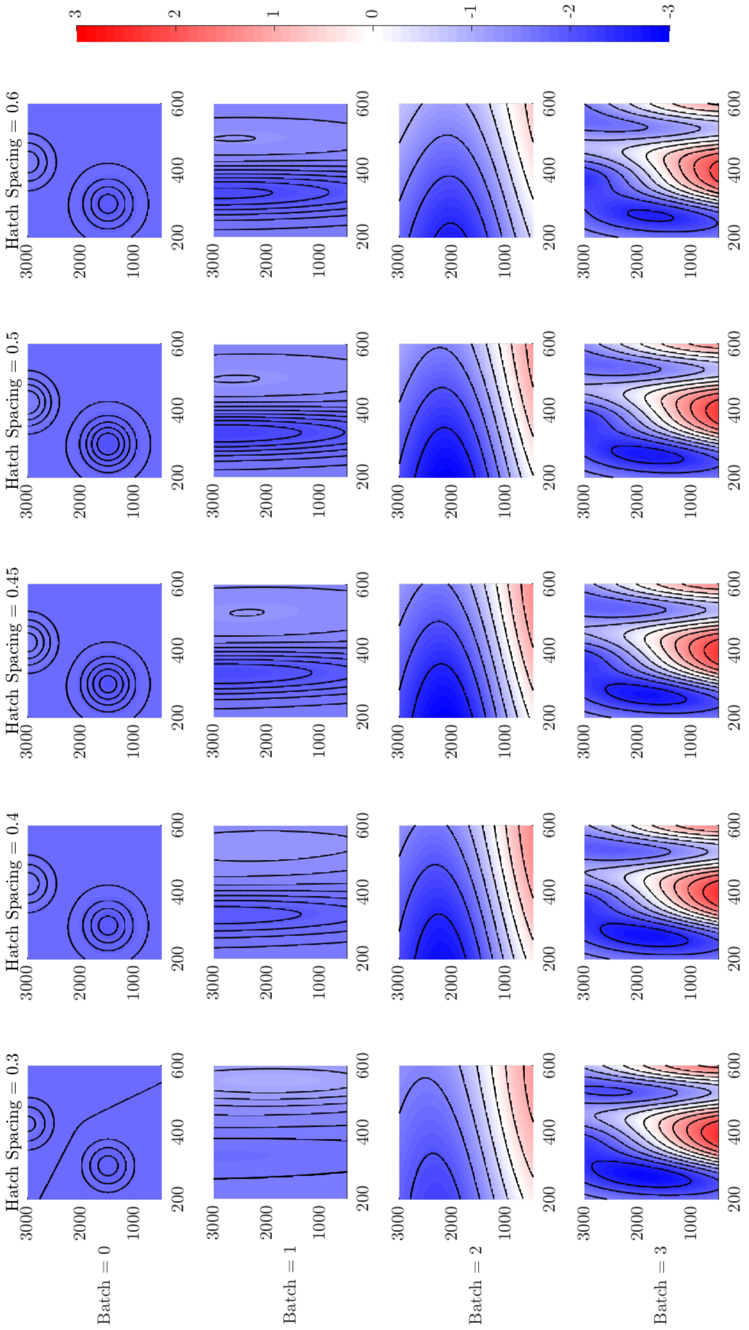


Figure 5.18: Evolution of predictive mean with contours in a 2-D grid representation of the design space at 5 fixed hatch spacing settings, $[0.30, 0.40, 0.45, 0.50, 0.60]$. The x-axis is the Laser Power (W) and the y-axis is the Nozzle Velocity (mm/min). The figure above assesses the early changes in predictive mean (with contours) between the training batch data to the 3rd batch in which the optimal process parameter settings for the DAS are determined.

shape. Whereby, in batches 0 to 2 of Figure 5.18 the response surface remains relatively smooth with little variation in shape or degree of change as shown with the smooth contours.

Change occurs in the 3rd batch in which the optimum is located on experiment 3 and the $4.5-9.0\mu$m DAS region becomes increasingly more explored. As shown in Figure 5.18, the design space became increasingly defined in the red regions of the heat map corresponding to the $4.5-9.0\mu$m DAS region. Also in the 3rd batch as the hatch spacing increases at a high Laser Power (> 500) the proportion of design space representing the ($> 4.5\mu$m) DAS region (red) gradually reduces for higher nozzle velocities.

Secondly, Figure 5.19 and Figure 5.20 illustrate the changes in predictive uncertainty and mean respectively from the 4th batch to the 6th batch. Unlike the initial batches shown in Figure 5.17 the predictive uncertainty in Figure 5.19 seems to have greatly reduced suggesting convergence in the prediction accuracy across the whole design space. However, in both the 5th and 6th batches the predictive uncertainty seems to be increasing, especially at hatch spacing > 0.5. This could suggest divergence in the model accuracy but is most likely due to the re-tuning of the GP hyper-parameters that occurs between each iteration step t. Whereby, the length scale hyper-parameter (L_d) corresponding to the hatch spacing is likely getting smaller suggesting greater variability in DAS over smaller changes in hatch spacing which could result in the increase in predictive uncertainty observed in Figure 5.19 at the 6th batch.

On the other hand in Figure 5.20 the overall shape and change in the response surface for the DAS from the 4th batch to the 6th batch is much smoother suggesting convergence in predictive mean. Whilst the $2.0-4.5\mu$m DAS (blue) region predominately occupies a similar area of the design space at nozzle velocities (> 1000 mm/min) for the entire laser power range ($200-600$ W). As the hatch spacing increases the $2.0-4.5\mu$m DAS (blue) region shifts into area's of the design space with a lower nozzle velocity for at least low laser power. As a result it can also be observed that the formation of DAS $> 4.5\mu$m (red) requires a higher laser power as the hatch spacing increases at low nozzle velocities.

Finally, in order to access whether the BB-DoE algorithm had converged to an optimal stopping point by the 6th batch the stopping criterion developed in Section 5.2.6 is assessed and shown in Figure 5.21. As discussed previously, from the initialisation of the BB-DoE algorithm to the 4th batch there were large variations in the predictive mean between batches. However, despite the predictive uncertainty increasing in some areas of the design beyond the

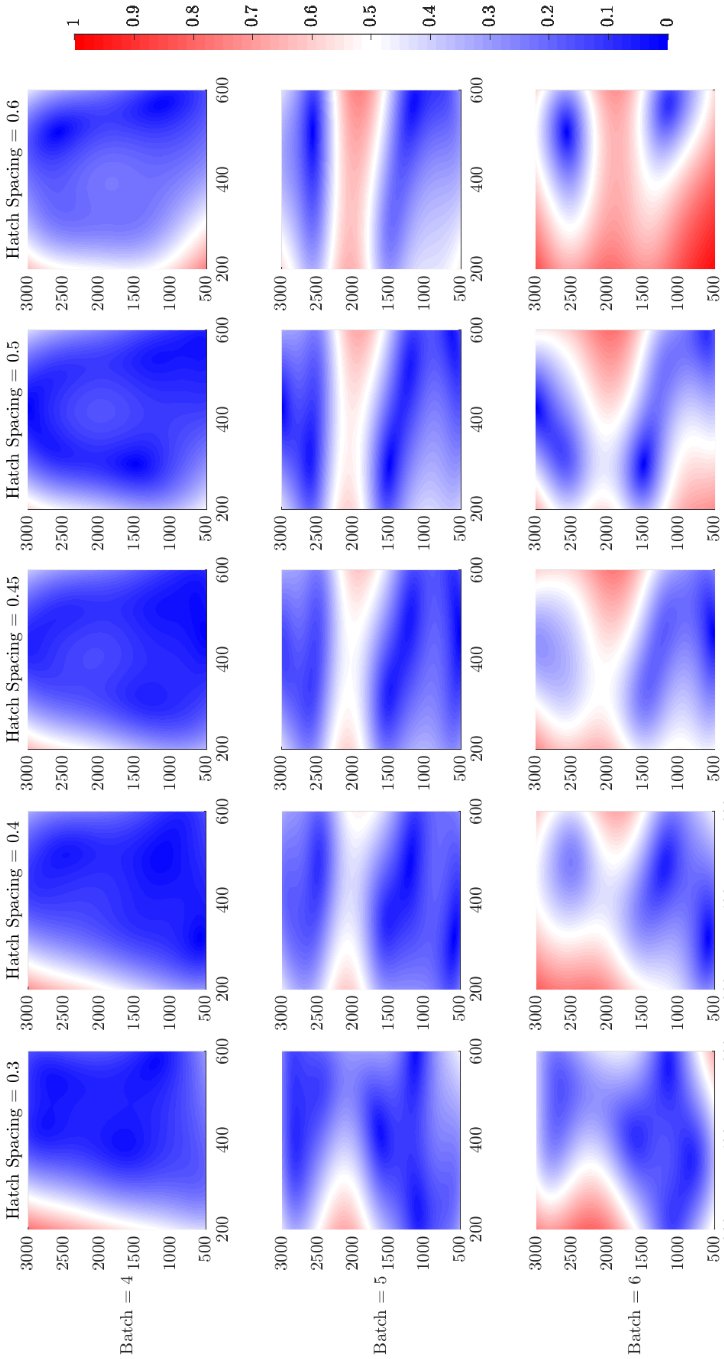


Figure 5.19: Evolution of predictive uncertainty in a 2-D grid representation of the design space at 5 fixed hatch spacing settings, $[0.30, 0.40, 0.45, 0.50, 0.60]$. The x-axis is the Laser Power (W) and the y-axis is the Nozzle Velocity (mm/min). The figure above assesses the changes in predictive uncertainty between the 4$^{\text{th}}$ batch to the 6$^{\text{th}}$ batch after the optimal process parameter settings for the DAS have been determined.

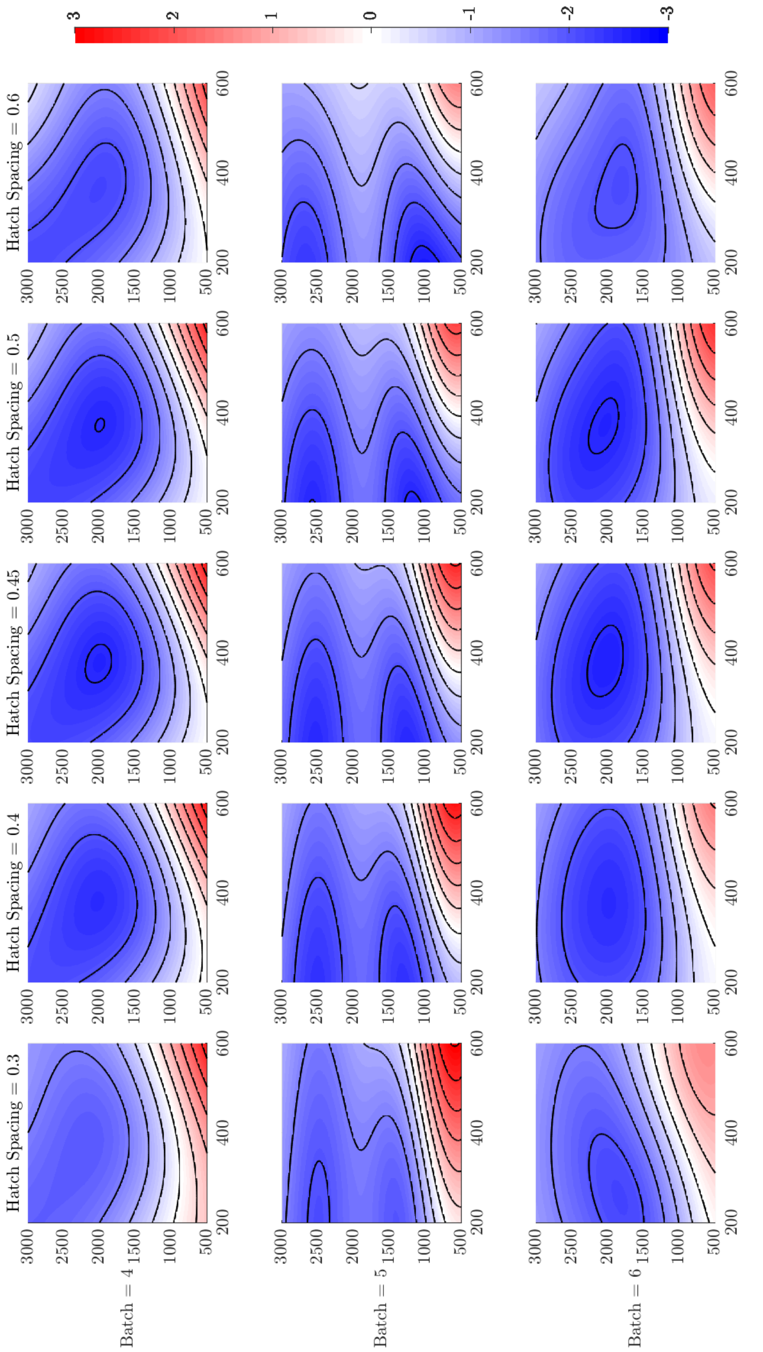


Figure 5.20: Evolution of predictive mean with contours in a 2-D grid representation of the design space at 5 fixed hatch spacing settings, $[0.30, 0.40, 0.45, 0.50, 0.60]$. The x-axis is the Laser Power (W) and the y-axis is the Nozzle Velocity (mm/min). The figure above assesses the changes in predictive mean (with contours) between the 4th batch to the 6th batch after the optimal process parameter settings for the DAS have been determined.

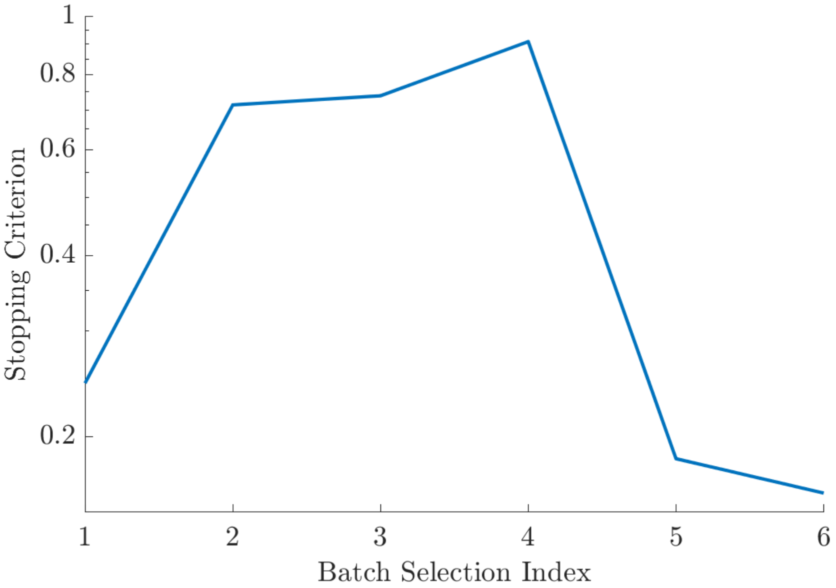


Figure 5.21: Evaluation of the stopping criterion for the BB-DoE algorithm on the DED manufacturing case study.

4th batch, the variation in predictive mean from batch to the next begins to reduce. This suggests that the predicted response surface of the GP surrogate model converging and hence the BB-DoE algorithm can be stopped after the 6th batch of 10 potential batches.

This work has demonstrated that the BB-DoE algorithm can satisfy the aims and objectives set out in Section 5.4.3 for a real AM problem specifically, identifying the optimal process parameter settings and their interactions on their effect on DAS in a DED processes in as few experiments as possible.

5.5 Conclusion

A BB-DoE algorithmic framework using a newly developed Non-Greedy NGB-GP-CB acquisition cost function was designed and evaluated in this Chapter. Developed alongside the Non-Greedy BB-DoE algorithm was a stopping criterion designed to suggest early stopping, which occurs when the output predictions from batch to batch become consistent as the surrogate model prediction accuracy converges. Both of these newly designed algorithmic concepts were assessed on the same synthetic benchmark functions previously applied in Chapter 4, as well as a blind AM case study.

The performance of the Non-Greedy NGB-GP-CB acquisition cost function on the synthetic benchmark functions was shown to have improved exploration performance as desired during its design process by including conditioning, constraints and β_t modifications. This performance was exemplified through its comparison against a current state-of-the-art Greedy BB-DoE algorithm using GP-BUCB which also utilises the conditioning property discussed in Section 5.2.1. Whilst the exploration performance improved, it was expected to have a decrease in exploitative performance when assessing the Regret on the benchmark functions. However, the exploitative performance of the NGB-GP-CB was only slightly worse with decreased performance in early batches but still locating the global optimum by the end of the experimentation budget (T).

The NGB-GP-CB was also briefly compared against one of the few current state-of-the-art Non-Greedy BB-DoE algorithm, PPES with the results demonstrated in Shah and Ghahramani [192] which assessed the Regret on Branin-Hoo and Mixture of Cosines. The comparison could only be assessed with regard to their exploitation performance in terms of Regret in which NGB-GP-CB was found to have comparable exploitative performance.

The results concerning NGB-GP-CB application onto the 4-Dimensional Hartmann benchmark function illustrated decreased exploration performance, achieving a NRMSD of 13 %. However, the stopping criterion value did not surpass $1x10^{-2}$, in contrast to both the Branin-Hoo and Mixture of Cosines $< 1x10^{-3}$. This suggests the NGB-GP-CB may have not converged and may be suitable for implementation on a larger experimentation budget.

Whilst NGB-GP-CB did outperform the GP-BUCB on NRMSD by 2.7 %, this also showed a decrease in model prediction accuracy when compared to the 2-Dimensional Branin-Hoo function (< 2 %) and Mixture of Cosines (< 7 %). This suggests poor exploration performance with increasingly large input dimensions. Initially, the β_t was modified to include an input dimension scaling parameter to alleviate the impact of increasingly complex optimisation landscapes for high dimensional problems. However, despite the improved performance shown between BB-DoE and B-DoE on identical benchmark problems another factor may be required to improve performance on high-dimensional problems.

The stopping criterion formulated was based on stopping in the event of output predictions from one batch to the next having minimal variation. Whereby, when minimal variations occur an accurate surrogate model is likely to have converged. However, the performance achieved on the benchmark functions suggest that convergence in the stopping criterion can occur when the surrogate model ceases to improve. Thus, this could occur even in scenarios in which the derived surrogate model is in-accurate such as for the GP-BUCB on the Mixture of Cosines function in Figure 5.10. Wherein the NRMSD converged to 12.8 %, whilst the stopping criterion plateaus at $5\text{x}10^{-3}$ which is almost an order of magnitude greater than NGB-GP-CB at $6\text{x}10^{-4}$. Therefore, a modification to the previously derived stopping criterion may be required to account for scenarios involving premature convergence. Although an additional parameter would likely be required, but in doing so a more robust stopping criterion can be achieved.

Lastly, the BB-DoE using NGB-GP-CB was assessed on optimising a DED process in order to investigate the process interactions between three input parameters and the formation of the DAS within a nickel super-alloy. It was shown that within the first 3 batches the process parameter settings which could identify the optimal DAS of 4.5 μm were determined. It was also shown within the study that the previously un-explored region for the DAS 4.5 – 9.0 μm was explored in later batches prior to initial convergence of the surrogate model prediction accuracy. This was validated when assessing the stopping criterion which from the 4th to 6th batch, the stopping criterion began to converge towards values demonstrated in the benchmark functions ($< 1\text{x}10^{-3}$). Although from the 4th batch, the predictive uncertainty across the design space began to increase despite apparent convergence in output prediction. This was most likely due to model hyper-parameter updates that began to fine-tune with each successive batch, which resulted in areas of increased exploration interest which could lead to further convergence in the stopping criterion. For discussion and reflection of research decisions, limitations and impacts refer to Section 7.2.

Chapter 6

Multi-Objective Batch DOE Optimisation

In Chapters 4 and 5, an investigation has been conducted into the viability of using the BO algorithm onto expensive AM DoE problems and extending the aforementioned BO algorithm for Batch DoE selection. As a result of the investigations and simulation analysis it was found that the GP-CB acquisition function was most suitable for extension into a Non-Greedy Batch acquisition function, for which NGB-GP-CB was derived. NGB-GP-CB was subsequently assessed for performance on a series of synthetic benchmark functions as well as an expensive AM case. The exploratory performance was improved at the expensive of early exploitative performance which recovered by the end of the experimentation budget (T) as well as locating the global optimum.

The final novelty of the thesis is to extend the previous novel BB-DoE algorithm for dealing with multi-objective issues which are of concern in additive manufacturing DoE problems, as aforementioned in Section 2.1.6. Specifically, the BB-DoE algorithm will be modified and integrated with a PD-MOEA which in the MOBO literature methods primarily introduced methods using EI [113] or kriging variance [140] [139]. Additionally a secondary selection mechanism to non-interactively select a batch of experiments from the POS at each iteration was modified from [23].

Whilst the developed MOBB-DoE algorithm is a novel formulation in the context of the BO DoE literature (see Section 2.4), although due to unforeseen circumstances (see Section 7.2.1 an AM DoE application was not performed for this algorithm. However, the novel adapted MOBB-DoE algorithm will be assessed upon a synthetic benchmark function from the multi-objective optimisation literature to assess performance.

6.1 Introduction

In this chapter, the aim is to further extend the previously developed NGB-GP-CB acquisition function for further additive manufacturing problems, namely dealing with multiple conflicting outputs. In additive manufacturing it is common to find an optimisation process in which there is more than one output of interest such as: microstructure properties, defect formation, manufacturing cost, etc. However, there is no guarantee that an experiment would provide an optimal solution for each output of interest as some outputs are in conflict with one another.

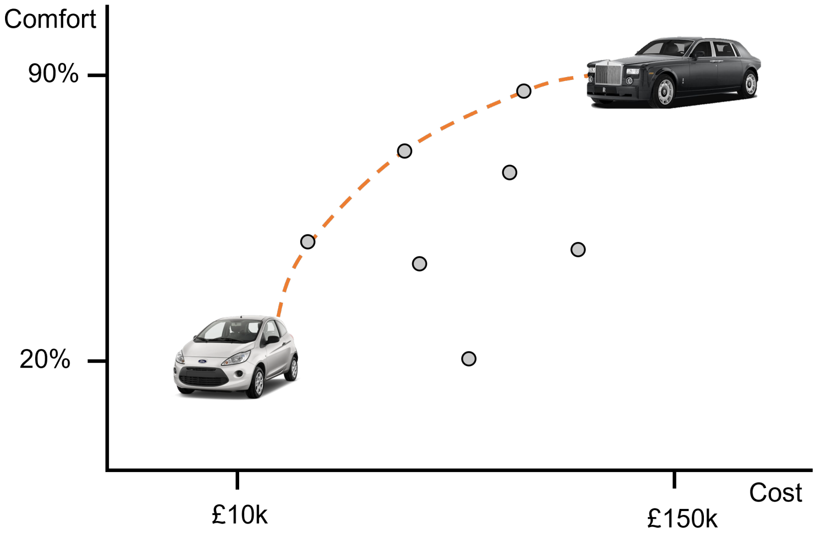


Figure 6.1: An illustration depicting a simple example of trading off between two conflicting output objectives, cost and comfort of a car.

In Figure 6.1, a simple example of conflicting outputs of cost vs comfort when comparing cars for purchase is used to illustrate the issue of solving multi-objective problems. Whereby, as the comfort of a car increases the corresponding cost to manufacture the car also increases. This leads to a set of optimal choices where no one candidate can improve over another in one output without a corresponding detriment to another output value, this is known as a non-dominated set or a Pareto Optimal Set (POS). As a result, there no longer exists a single candidate that is optimal but a set of candidates to be chosen from, therefore multiple conflicting outputs add an increased layer of complexity to the manufacturing DoE problems.

6.1.1 Multi-Objective Bayesian Optimisation in Literature

Within the literature Multi-Objective Bayesian Optimisation (MOBO) is a field of recent interest by a plethora of authors investigating the potential of MOBO as well as innovating improvements of other authors. The MOBO literature can be broadly classified into three main sections:

- Conversion of Multi-Objective Problems (MOP) into Single-Objective Problems (SOP) using scalarisation/aggregation functions.
- Adaptation of Single-Objective Bayesian Optimisation (SOBO) acquisition functions for use on MOPs.
- Innovative Algorithmic approach.

MOBO: Scalarisation

The conversion of MOPs into several SOPs was initially developed in Decomposition Based Multi-Objective Evolutionary Algorithm (D-MOEA) literature in early works such as MSOPS in [106] [107], ParEGO in [127] and was popularised as a research topic in MOEA/D by Zhang and Li [247]. However, these approaches have been adapted for use in a selection of MOBO approaches: ParEGO [127], TS-TCH [173], and their variants [17] [48] [2] [96] [50].

In each iteration of the ParEGO, the scalarisation occurs using the Tchebycheff function (See Equation 2.47) with a weight vector drawn randomly from a simplex lattice design [127]. The data set outputs are then scalarised using the current weight vector, after which the scalarised data is utilised to fit the DACE model [127]. Once the DACE surrogate model has been fitted, the EI acquisition function is optimised using a GA to determine the next optimal experiment to implement. The setup of the ParEGO methodology allows for simple and fast optimisations of a MOP using EI, despite these benefits it is often outperformed by other advanced approaches [179].

Similarly, TS-TCH employs the Tchebycheff function (See Equation 2.47) to scalarise its outputs into a SOP for sequential optimisation of Thompson Sampling (TS) acquisition function. In contrast, TS-TCH uses a prior on the scalarisation weights to encode decision maker preference to search areas of the PF that are of interest. Unlike ParEGO which fits the DACE model with the scalarised data, in TS-TCH the GP surrogate models are maintained for each output objective but scalarised during optimisation of the TS acquisition function. TS-TCH provides a flexible MOBO methodology, but the increased flexibility comes at the cost of requiring the selection and tuning of the scalarisation prior. Whereby, the scalarisation prior varies by domain and is dependent upon the decision maker's choice but advantageously can be interactively adapted during optimisation [173].

MOBO: Acquisition Functions

Another method for tackling MOBO is to instead adapt the acquisition functions to tackle MOPs rather than their conversion into several sub-SOPs using scalarisation functions. This is most frequently implemented using a Performance Indicator (P-I) known as, Pareto Hyper-Volume (HV) or adapted for use with a BO acquisition function.

Pareto Hyper-volume as discussed in Section 2.3.2, is a P-I that captures the volume of space above the PF that is occupied by the candidate experiment with respect to a given reference point [64]. By maximising the HV the optimiser will also locate the true PF, the HV is an ideal measure of performance for MOBO. As such a large proportion of MOBO algorithms implement a Pareto Hyper-Volume (HV) approach: SMS-EGO [179], EHVI [64], SUR [176], and their variants [50] [51] [1] [239].

In SMS-EGO a set of non-dominated candidates or the POS is determined by optimising a GP-CB acquisition function for each of the output variables. Once a POS is discerned, a GP surrogate model is fitted for each output variable prior to optimisation of the SMS-EGO acquisition function. Finally, each candidate experiment in the POS has their expected gain in HV assessed, to select the next optimal experiment to perform [179].

Expected Hyper-volume Improvement (EHVI) works in a similar manner to SMS-EGO in which firstly a POS must be determined. However, the computation of the EHVI can be very expensive to calculate, therefore a piecewise integration of the partitioned output space is utilised [64]. The output space is split into cells (see Figure 6.2) which are either active (non-dominated) or inactive (dominated) cells [65]. Inactive cells are said to have a contribution of 0 and thus by taking the sum of all cells, the piecewise integration is taking the sum of contributions from active cells [65]. Therefore, the probability of improvement is simply reduced to the probability that a candidate experiment lies within a non-dominated cell [102].

$$\text{EHVI Cells} = (|\text{POS}| + 1)^{\text{M}}$$

Subsequently, the speed of computation is directly related to the total number of cells required, which is dependent upon the size of the POS and the number of output variables. This relationship causes the total number of cells to grow exponentially with the number of outputs, limiting the feasibility of implementation to two or three outputs [102].

Stepwise Uncertainty Reduction (SUR) methodology is often equated to that of EHVI's as in both approaches the output space is separated into cells, see Figure 6.2. However, in SUR the acquisition function utilises the Probability of Improvement (PI) of the uncertainty or hypervolume at a candidate experiment [102]. Whereby the SUR criterion will add the candidate experiment that minimises the expected hypervolume of the POS [176].

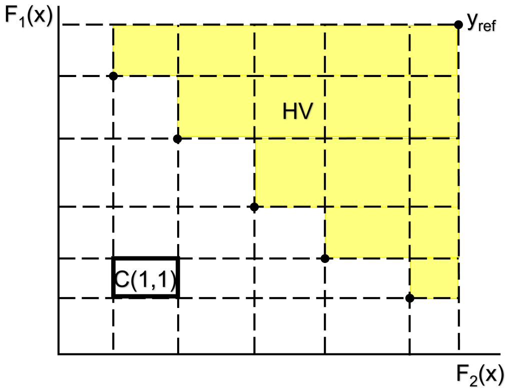


Figure 6.2: A representation of how the output of a 2-D search space is split in order to calculate the hyper-volume of a candidate experiment. The black points are points in the population, except the point in the top-right corner which is the reference point. The yellow region defines the measured hypervolume (HV) and an example cell is illustrated as the solid black boundary.

As previously described a primary issue of SUR and EHVI is the total number of cells required for the computation [102]. This is hindered as the optimisation progresses as undoubtedly the size of the POS will likely increase, which will raise the computational load of the SUR/EHVI [176]. This is a large contributor as to why SUR/EHVI are incapable of use for tackling MOPs of greater than three objectives [102].

Whilst HV adaptations make up a large proportion of MOBO algorithms that are adapted from SOBO algorithms, there are also alternatives available within the literature. Similarly, information-based acquisition functions could be extended for use in a multi-objective setting. In most information theoretic algorithms, the aim of the acquisition function was to determine the next optimal experiment to reduce the entropy or gain the most information about the location of the true optimum. Therefore, in MOBO the goal was to identify the experiment which would result in the largest reduction in entropy or gain in information about the Pareto Optimal Set (POS) rather than the global optimum. Of which the main approaches within literature include: PESMO [102], MESMO [20], and their variants [78] [22] [68].

Predictive Entropy Search Multi-Objective Optimisation (PESMO) follows the same implementation methodology as PES detailed in Section 2.2.2, modified from seeking the global optimum (x^*) to seeking the POS (x_{pos}) as shown in Equation 6.1 [102]:

$$u_{PESMO}(X) = H\left(P(y \mid D, x)\right) - \mathbb{E}_{P\left(x_{pos} \mid D\right)} H\left[P\left(y \mid D, X, x_{pos}\right)\right] \tag{6.1}$$

Where, u_{PESMO} is the Predictive Entropy Multi-Objective Search (PESMO) Acquisition Function, $H\left(P(y \mid D, x)\right)$ is the differential entropy of the posterior predictive distribution, $P\left(x_{pos} \mid D\right)$ is the posterior distribution over the Pareto optimal set, and $H\left[P\left(y \mid D, X, x_{pos}\right)\right]$ is the differential entropy of the posterior predictive distribution of y given the observed data (D) and POS (x_{pos}) are known.

Similarly to PES, the second term must be approximated by sampling the objective functions from their respective posterior distribution several times [102]. This is achieved using random kernel features and linear models from which the POS is approximated using a grid search of $1000 * d$ points [102]. However, in high dimensional problems a $1000 * d$ point grid is too expensive to optimise such that instead NSGA-II is used instead [55].

Furthermore, Max-Value Entropy Search Multi-Objective Optimisation (MESMO) as with PESMO is a MOBO adaptation of its SOBO counterpart MES. MESMO methodology follows an identical implementation approach as detailed in Section 2.2.2 that has been modified to maximise the information gain about the output Pareto Front [20]. Whereby, Equation 2.34 has been reformulated for maximising information gain about the location of the PF in the output search space in Equation 6.2 [20]:

$$u_{MESMO}(X) = H\left(P(y \mid D, x)\right) - \mathbb{E}_{P\left(y_{pos} \mid D\right)} H\left[P\left(y \mid D, X, y_{pos}\right)\right] \tag{6.2}$$

Where, u_{MESMO} is the Max-Value Entropy Multi-Objective Search (MESMO) Acquisition Function, $H\left(P(y \mid D, x)\right)$ is the differential entropy of the posterior predictive distribution, $P\left(y_{pos} \mid D\right)$ is the posterior distribution over the Pareto optimal set in the output space, and $\mathrm{H}\left[\mathrm{P}\left(\mathrm{y} \middle| \mathrm{D}, \mathrm{X}, \mathrm{y}_{\mathrm{pos}}\right)\right]$ is the differential entropy of the posterior predictive distribution of y given the observed data (D) and Pareto optimal set in the output space ($\mathrm{y}_{\mathrm{pos}}$) are known.

In the same manner in which PESMO must approximate its second term, MESMO must approximate the second term as well. This is achieved in the same manner as PESMO using random kernel features, linear models, and NSGA-II to approximate the PF. As demonstrated in Equation 2.35, the MES has an analytical form using Monte Carlo estimation of the expectation in Equation 2.34. This analytical form can also be extended for MOP to be reformulated into Equation 6.3 [20]:

$$\mathrm{u}_{\mathrm{MESMO}}(\mathrm{X}) = \frac{1}{\omega}\sum_{\mathrm{i}=1}^{\omega}\sum_{\mathrm{j}=1}^{\mathrm{M}}\left[\frac{\gamma_{\mathrm{i}}^{\mathrm{j}}(\mathrm{X})\,\varphi\left(\gamma_{\mathrm{i}}^{\mathrm{j}}\right)}{2 * \Phi\left(\gamma_{\mathrm{i}}^{\mathrm{j}}\right)} - \log\Phi\left(\gamma_{\mathrm{i}}^{\mathrm{j}}\right)\right] \tag{6.3}$$

and

$$\gamma_{\mathrm{i}}^{\mathrm{j}} = \frac{\mathrm{y}_{\mathrm{i}}^{\mathrm{j}*} - \mu_{\mathrm{j}}(\mathrm{X})}{\sigma_{\mathrm{j}}(\mathrm{X})} \qquad \text{and} \qquad \mathrm{y}_{\mathrm{i}}^{\mathrm{j}*} = \max\left(\mathrm{y}_{1}^{\mathrm{j}}, \ldots, \mathrm{y}_{\mathrm{R}}^{\mathrm{j}}\right)$$

Where, ω is the total number of Monte Carlo estimations of the global optimum y^{*} for output j. M is the total number of output objectives, $\mathrm{y}_{\mathrm{i}}^{\mathrm{j}*}$ is the optimum sample for output objective j, φ is the probability density function, and Φ is the cumulative density function.

PESMO and MESMO offer structured extensions for information-based BO algorithms for implementation on MOPs, but they are also not without their own limitations. Much alike their SOBO counterparts they require a series of approximations to calculate their entropies, which can sub-optimal [21]. Despite these approximations the optimisation may also be expensive to implement and is highly dependent upon the number of Monte Carlo samples [21].

Whilst adapted BO acquisition functions using the hyper-volume indicator are the pre-dominantly implemented in MOBO literature, there are MOO extensions that utilise SOBO acquisition functions. These extensions focus on using the PD-MOEA optimiser with BO acquisition functions to generate a POS that cover the PF such as: Multi-EGO [113], K-MOGA/KD-MOGA [140] [139], and UseMO [23].

Multi-EGO was one of the first algorithms to extend the use of BO for use on MOP which employed a PD-MOEA using a kriging model with EI [113].

Whereby, the EI was applied onto the kriging predictors in each iteration for every output objectives simultaneously and utilised non-dominated sorting to maximise EI simultaneously. Therefore, the experiments that were chosen were not the ones that maximised all the EI, but rather the trade-off between them all [185].

K-MOGA is a kriging based MOGA which utilised the kriging variance as the BO acquisition function. In each iteration of the MOGA, a kriging model is fit to each output objective which is used to evaluate the predictive variance of the entire population [140]. In order to ascertain which individuals in the population to select the predictive variance is assessed to measure if a threshold is passed (for which the threshold has a closed-form) and if they do they are selected. By using this approach, the K-MOGA ensures that only in areas of high uncertainty is the experiment performed. In order to select the parents for the next generation of the MOGA a non-dominated sort is used [185]. KD-MOGA is an extension of the K-MOGA with the inclusion of a space-filling design [139]. The space-filling design is used each generation in order to sample better points during reproduction of the MOGA [185].

MOBO: Innovative Algorithms

The final style of MOBO algorithms are methods which offer an innovative algorithmic approach for optimisation to identify a POS. These methods are often the most unique as they are not adaptations of methods or acquisition functions previously introduced in BO literature. The most notable algorithms introduced into the literature are: PAL [256], USeMO [23], and their variants [18] [257] [22].

Firstly, Pareto Active Learning (PAL) is a classification MOBO algorithm which aims to identify the POS iteratively in an efficient scheme. At each iteration PAL selects a candidate experiment from a finite discrete set known as the *Uncertain Set* in the search space to classify it as either Pareto Optimal or Non-Pareto Optimal [256]. At each iteration a GP surrogate model is built for each output which is used to infer their predictive means and standard deviations for all outputs for every candidate experiment. After which each candidate is assigned an uncertainty region ($\Omega_{t}(x)$), as given in Equation 6.4 [256]:

$$\Omega_{t}(x) = \Omega_{t-1}(x) \cap U_{\mu_{t},\sigma_{t},\beta_{t+1}}(x) \tag{6.4}$$

and,

$$U_{\mu_t,\sigma_t,\beta_{t+1}}(x) = \left[y : \mu(x) - \beta^{\frac{1}{2}}\sigma(x) \leq y \leq \mu(x) + \beta^{\frac{1}{2}}\sigma(x)\right]$$

where, $\Omega_t(x)$ is the uncertainty region for iteration t. $U_{\mu_t,\sigma_t,\beta_{t+1}}$ is a hyper-rectangle defined by μ_t, σ_t, and β_{t+1} which are the predictive mean, standard deviation and scaling parameters respectively.

During PAL using the uncertain set candidate experiments are assigned to one of the two regions: Pareto Optimal or Non-Pareto Optimal, from which their classification remains unchanged [256]. An experiment is deemed to be Pareto Optimal if the worst value in its $\Omega(x)$ is not dominated by the best value in other experiments $\Omega(x)$. Whilst on the other hand it is classified as Non-Pareto Optimal if an experiment's best $\Omega(x)$ is dominated by any other experiments worst $\Omega(x)$. All other points are to remain uncertain, after which PAL will select an experiment to perform from the uncertainty set with the largest $\Omega(x)$ [102].

Secondly, Uncertainty-aware Search Multi-objective Optimisation (USeMO) operates equivalently to PD-MOEA algorithms (Section 2.3.2) by employing SOBO acquisition functions, with an uncertainty maximisation selection mechanism for the selection of the next experiment candidate. At each iteration, GP surrogate models are built for each output variable independently; these are subsequently used to cheaply optimise a chosen SOBO acquisition function from the available library [23]. Thus, for each candidate experiment there will be a combination of multiple acquisition values corresponding to each output variable, combining to form a cheap MOP to optimise using a MOO solver. USeMO utilises NSGA-II to cheaply generate a PF of potential optimal candidate experiments, from which the uncertainty maximisation mechanism determines the next optimal candidate experiment to select [23].

In order to determine which candidate experiment from the POS to select and evaluate, a mechanism is required to differentiate value between each candidate in the PF. Previously in literature this has been achieved using two approaches: Pareto Hyper-Volume (HV) [64] and the Uncertainty region [176]. USeMO uses the uncertainty region introduced in PAL [176] in Equation 6.4, in which the volume of the uncertainty surrounding the candidate experiment forms a hyper-rectangle. After the NSGA-II has optimised a PF of POS each candidate's uncertainty region volume are determined and maximised to select the candidate with the largest uncertainty as the next experiment for evaluation [23].

6.1.2 Multi-Objective Batch Bayesian Optimisation in Literature

In Section 6.1.1, a review of MOBO literature was conducted to examine the currently available literature, which was primarily focused upon sequential selection. However, the goal of this work is to develop a Multi-Objective Batch Bayesian Experimental Design Optimisation (MOBB-DoE). Thus, building upon the previous literature review a survey of the current Multi-Objective Batch Bayesian Optimisation (MOBBO) algorithms is performed: MOEA/D-EGO [248], qParEGO [50], qEHVI [50], qNEHVI [51], and TSEMO [37].

Firstly, MOEA/D-EGO is an extension of the D-MOEA algorithm which selects a batch of candidate experiments by simultaneously optimising several EIs, of different aggregates of a decomposed MOP using varied weight vectors [243]. The EI of each sub-SOP scalarised by its weight vector is then optimised simultaneously using D-MOEA detailed in Section 2.3.2, using Equation 6.5 [243].

$$\text{maximise}\left[\text{EI}_{\text{w}^1}(\text{x}), \text{EI}_{\text{w}^2}(\text{x}), \ldots, \text{EI}_{\text{w}^W}(\text{x})\right] \tag{6.5}$$

where, w^i is the i^{th} weight vector, W is the total number of weight vectors and $\text{EI}_{w^i}(x)$ is the EI function of the aggregation function with the i^{th} weight vector.

Secondly, qParEGO extends ParEGO to the greedy batch selection setting by reversing the order in which the surrogate models are built to before the scalarisation functions are applied, the opposite of the order described in Section 6.1.1. For each point added to the batch set in a sequential greedy manner a different weight vector is applied [50]. Whilst MOEA/D-EGO and qParEGO are similar in methodological application of utilising scalarisation of the MOP into several sub-SOPs with EI applied onto each, they utilise different algorithmic approaches [243].

Thirdly, qEHVI is the parallel extension of EHVI detailed in Section 6.1.1, Wada and Hino [225] initially derived an exact formulation for the qEHVI approach but after finding it difficult to optimise jointly without gradients, an alternative was used [50]. Later [50] introduced a greedy batch selection scheme for qEHVI using the inclusion-exclusion principle (IEP) and optimised using gradients of the Monte Carlo (MC) estimator. IEP computed the joint Hyper-Volume Improvement (HVI) over the previously selected batch points and the new candidate point for each Monte Carlo sample. This involved computing the volume jointly dominated by each $2^q - 1$ unsampled point in the greedy batch candidate set [50]. Despite Daulton et al. [50] providing a suitable qEHVI formulation and optimisation strategy, the computations were expensive. Whereby, the computation scaled exponentially with batch size and multiplicatively with the number of hyper-rectangles in the box decomposition [51].

Fourthly, qNEHVI is an extension to qEHVI for tackling noisy MOPs but, it also introduced a different algorithmic approach towards calculating the joint HVI using Cached Box Decomposition (CBD) instead of IEP to reduce computational complexity [51]. CBD improves upon IEP as the joint HVI computed using box decompositions are performed once per MOBO iteration and stored. By doing so the qNEHVI reduced the computational scaling with batch size from exponential scaling to polynomial.

Finally, TSEMO employs Thompson sampling alongside NSGA-II to generate an approximate POS for the MOP, from which the next individual or batch set of candidate experiments are selected using the HV indicator [37]. Thompson sampling simply refers to choosing an action that is most likely to result in the optimum reward. In TSEMO, this is achieved by building surrogate GP models for each output variable, from which spectral sampling is used to draw a set of distinct functions for each output variable. Whereby, NSGA-II utilises the sampled functions to cheaply optimise and generate an approximate POS. The next candidate experiment is then selected from the POS by utilising the HV indicator, in which the candidate that maximises the HVI is chosen. In the batch setting TSEMO selects batches of candidates using a greedy selection strategy [37].

6.2 Multi-Objective Batch Extension of NGB-GP-CB Function

The goal of Multi-Objective Batch Bayesian Experimental Design Optimisation (MOBB-DoE) is to optimise a series of objectives simultaneously, which are unknown but can be evaluated pointwise exactly or in the presence of noise at multiple locations in parallel, as defined in Equation 6.6.

$$\mathbf{B}_{t+1} = \arg\max \mathrm{f}\left(\mathbf{B}_t\right) = \left[\mathrm{f}^1\left(\mathbf{B}_t\right), \ldots, \mathrm{f}^M\left(\mathbf{B}_t\right)\right] \tag{6.6}$$

Where, $\mathbf{B}_{t+1}$ is the batch vector set of Q candidates to be evaluated, M is the total number of output objectives to optimised simultaneously. At each iteration, a batch set of $\mathbf{B}_{t+1}$ points must be selected that seek to provide a diverse and set of non-dominated solutions known as the Pareto Optimal Set

(POS) that together form a Pareto Front (PF) that optimise a set of M output objectives.

6.2.1 MOBB-DOE Optimiser Analysis

As discussed in Section 2.3.2, in order to develop a MOBO algorithm suitable for implementation on DoE problems a non-interaction multi-objective optimisation (MOO) methodology is the most suitable. Non-interactive MOO methods are often optimised using a GA, after which the literature can be broadly classified into three algorithmic approaches:

- Pareto-Dominance Multi-Objective Evolutionary Algorithm (PD-MOEA)
- Performance Indicator Multi-Objective Evolutionary Algorithm (PI-MOEA)
- Decomposition Based Multi-Objective Evolutionary Algorithm (D-MOEA)

In Section 6.1.1, a variety of literature algorithms were integrated with BO acquisition functions to develop state-of-the-art MOBO algorithms such as: ParEGO [127], EHVI [64], SMS-EGO [179], PESMO [20], MESMO [20], UseMO [23] as well as their MOBBO variants [248] [50]. Each of these state-of-the-art algorithms which have previously integrated BO acquisition functions had to consider how to integrate these into MOO solver's whilst accounting for: non-interactive experiment selection, diversity/spread maintenance, exploitation-exploration trade-off and desired computational complexity. These characteristics must be assessed and weighted against the NGB-GP-CB acquisition function to decide which MOO solver to choose for integration with a GA to design a state-of-the-art MOBB-DoE algorithm.

In BO algorithms one core mechanism that is often in conflict with MOO is the requirement to select a single or batch of candidate experiment(s) at each iteration. In traditional MOO a POS is generated which trades off performance between all output variables considered, as to improve performance in one output often leads to degradation of other solutions. Once a PF had been generated where each candidate experiment is as optimal as the next, a decision maker would select from the POS. Whereas, in BO this selection must occur non-interactively and thus a secondary mechanism is required. Historically, in MOBO literature, the adaptation of non-interactive MOO approaches for BO algorithms has been tackled predominately using two approaches: Scalarisation and Hyper-volume.

Firstly, scalarisation approaches utilise weight vectors to convert MOPs into several sub-SOPs for optimisation on specialised algorithms such as MOEA/D-EGO [248] or ParEGO [127]. The advantages of such algorithms are that by scalarising the MOP into one or many SOP to be optimised, the original BO acquisition could be applied directly. Thus, allowing for a simple and fast MOBO algorithm to be developed. However, a key aspect that isn't currently explored in MOBO algorithms is maintaining the diversity and coverage of candidate experiments on the PF, which is determined by the selection of the weight vector set [237].

On the other hand this is topic is explored extensively in the D-MOEA literature with schemes that have sought to improve on the original weight vectors designs [82] [83] [250], introduce adaptive weight vectors schemes [202] [181] and design schemes to introduce user preferences [150] [178]. Although as one property of interest is non-interactive implementation, the user preference-based designs would not be suitable.

Generalised Decomposition (gD) [82] [83] is one D-MOEA literature method which aimed to generate an optimal set of weight vectors under the assumption that a reference PF exists for a particular scalarisation function. It was shown to outperform the original weight vector designs using simplex lattice [247] and uniform random sampling [112]. Although a limitation of this approach was the requirement of knowing the PF geometry to determine a reference PF but the authors argued a good performance could be achieved by using a linear PF. This would be unsuitable for application in AM manufacturing problems in which the PF geometry would pre-dominantly unknown.

In both ParEGO and qParEGO the weight vectors are randomly drawn from the initial simplex lattice designs, to be applied in each iteration and for each batch set candidate for the MOBO and MOBBO variants respectively. Whilst MOEA/D-EGO utilised the uniform random sampling design. In both MOBB-DoE algorithms the original weight vector generation methods were used suggesting there would be no guarantees the weight vectors would lead to efficient selection of candidate experiments that are well distributed along a PF [82] [83].

Secondly, HV was initially developed in the PI-MOEA literature as it acted as a performance indicator measuring the volume of space in which a non-dominated candidate occupied on the PF. HV provided a quantifiable measure of improvement that a particular candidate provided through its addition to the PF, which attracted research interest into the development and integration with various BO acquisition functions: GP-CB [179] and EI [176] [64] [50]

[51].

On the other hand, HV is computationally expensive to calculate as demonstrated in [179] [176] [64] [50] [51], whereby the output space is partitioned into cells. The number of cells required to compute the HV scales with the number of output objectives exponentially [102], which prevents HV-based methods from performing well on MOPs with greater than three objectives known as Many-Objective Optimisation Problem (MaOP).

Whilst not as popular as scalarisation or hyper-volume approaches, there is one class of acquisition function that could be directly scaled up into MOOs, Information-Theoretic acquisition functions. This was demonstrated by [102] and [20], where the predictive posterior distributions sought to gain information about a POS rather than the global optimum in PES/MES [101]/[230]. However, much alike their SOBO counterparts the MOBO information-theoretic acquisitions are expensive to calculate and require even more approximations to perform despite their theoretical basis.

MOBB-DoE: MOO Optimiser Summary

In summary, the MOBO algorithms from literature predominantly were based upon two types of non-interactive MOO: D-MOEA and PI-MOEA. Whilst D-MOEA using scalarisation functions allow ease of extension into MOBO algorithms through the conversion of MOP into several SOPs, if the weight vectors are not chosen carefully there is no guarantee of selecting candidate experiments that produce a diverse POS of experiments [82]. Although there have been vast literature contributions into the investigation of improving weight vector generation. Furthermore, by reducing the MOP into a SOP it reduces the capability of D-MOEA algorithms to capture the trade-off between multiple objectives, which can lead to more exploitative candidate selections [23].

On the other hand, PI-MOEA using performance indicators such as HV allow for a quantifiable measure of a candidate experiments improvement as an addition to the PF. Therefore, using HV as a base the derivation of multi-objective acquisition functions such as SMSego and EHVI was possible, but not without limitations as the HV acquisition functions are often computationally expensive to utilise beyond three output objectives [102] [20].

Another essential point to consider are the traits of the NGB-GP-CB acquisition function as they will contribute to the design of the MOBB-DoE algorithm to be produced and assessed throughout this chapter. NGB-GP-CB is a batch acquisition function which consolidates the acquisition value of each candidate within the batch set, based upon the SOBO acquisition function GP-CB. The BB-DoE algorithm developed using NGB-GP-CB also utilised constraints to prevent over-exploitation and promote exploration to further refine GP models and search for alternative optimums.

Considering the shortcomings of D-MOEA and PI-MOEA algorithms, whilst accounting for the traits of the NGB-GP-CB acquisition function PD-MOEA is the most suitable non-interactive MOO algorithm to implement. PD-MOEA detailed in Section 2.3.2, optimise under the principle of Pareto-dominance (Definition 2.3.2 - 2.3.4) where a candidate is optimal if it is not dominated by any of candidate. Thus, the goal of a PD-MOEA is to generate a POS that represent trade-offs between output objectives on a PF. However, an additional mechanism is required once a POS is found to discern which candidate from the POS will guide the MOBB-DoE towards to the true PF.

6.2.2 Multi-Objecitve Modelling

In order to optimise Equation 6.6 a suitable MOBB-DoE algorithm is required, within this thesis a combination of the NGB-GP-CB acquisition function with a PD-MOEA solver and finally a secondary POS selection mechanism will be used. In order to select a batch which optimises the multi-objective batch cost shown in Equation 6.7, the NGB-GP-CB acquisition function will be determined for each output objective of interest shown in Equation 6.8.

$$\mathbf{B}_{t+1} = \arg\max_{\mathbf{B}_t} u_{\mathrm{MOO-NGB-GP-CB}}\left(\mathbf{B}_t \,\middle|\, D_t\right) \tag{6.7}$$

Where,

$$u_{\mathrm{MOO-NGB-GP-CB}}\left(\mathbf{B}_t \,\middle|\, D_t\right) = \left[u^{i}_{\mathrm{NGB-GP-CB}}\left(\mathbf{B}_t\right), \ldots, u^{M}_{\mathrm{NGB-GP-CB}}\left(\mathbf{B}_t\right)\right] \tag{6.8}$$

where, $u_{\mathrm{MOO-NGB-GP-CB}}$ is the multi-objective non-greedy batch acquisition function, $\mathbf{B}_t$ is a batch vector set of Q candidates to be evaluated, and $u^{i}_{\mathrm{NGB-GP-CB}}$ is the non-greedy batch acquisition value for the i^{th} output objective of M total output objectives for the batch vector set $\mathbf{B}_t$.

In order to optimise M NGB-GP-CB acquisition functions as shown in Equation 6.8, corresponding predictive means are required for all M output objectives. Only the predictive mean changes with each output objective as mentioned in Section 5.2.1 the predictive standard deviation is dependent on only the input variables. In order to alter the predictive mean for each output objective a corresponding GP surrogate model is required for inference, leading to the need for tuning multiple GP surrogate models.

Hence the observed experiment sets of data are replicated and partitioned for their corresponding output variables ready for use as training data sets for each GP surrogate model. Once partitioned each representative data set can then be used to tune each GP surrogate model using the framework set out in Section 3.4 to tune each respective model hyper-parameters.

6.2.3 Multi Acquisition Function Ranking

After each GP surrogate model is tuned for their respective output, they can be utilised in the NGB-GP-CB acquisition function with the methodology set out in Section 5.2. Therefore, for each batch set of candidate experiments there will be *M* acquisition values or to be referred to in this chapter as the output objectives as depicted in Equation 6.7.

As noted in Section 2.3.2, there are a variety of PD-MOEA solvers available within the literature that optimise MOPs using the basis of Pareto-Dominance set out using Definitions 2.3.2 - 2.3.4. In this thesis, Fonseca and Fleming [73] MOGA will be used to cheaply optimise the output objectives.

MOGA used a Pareto Ranking strategy (See Figure 2.14 for 2-D example illustration), which ranks all candidates according to their level of domination. Whereby a non-dominated candidate refers to having a Pareto rank of one with all subsequent ranks being dominated candidates.

Prior to Pareto ranking, each candidate's output objectives ($\mathrm{u}^{\mathrm{i}}_{\mathrm{NGB-GP-CB}}$) are multiplied by negative one to convert the output objectives acquisition value from a maximisation problem into a minimisation problem. Once transformed into a minimisation problem to aid in the determination of Pareto ranks the output objectives are normalised in the range of $[0, 1]^{\mathrm{d}}$ using Equation 6.9.

$$\mathrm{v}^{\mathrm{norm}} = \frac{\mathrm{v} - \mathrm{v}^{\mathrm{min}}}{\mathrm{v}^{\mathrm{max}} - \mathrm{v}^{\mathrm{min}}} \tag{6.9}$$

where, $\mathrm{v}^{\mathrm{norm}}$ is the variable v normalised to the range of $[0, 1]$, $\mathrm{v}^{\mathrm{min}}$ and $\mathrm{v}^{\mathrm{max}}$ are the minimum and maximum variable values respectively.

Once the output objectives are transformed through minimisation and normalisation of output objectives, the Pareto ranks of each candidate will be determined by iteratively setting a single candidate as a reference to compare with the remaining GA population candidates.

$$C_{i=1}^{P} = \begin{cases} C_i & \text{Reference Candidate} \\ C_{\neq i} & \text{Comparison Candidates} \end{cases}$$

In each iteration, a single reference candidate is set to be compared with the remaining candidates, which are compared along each output objective. Each comparison candidates output objective value is subtracted from the corresponding reference candidate output objective value to find its difference value.

$$F_{i=1}^{P} = u_{NGB}^{norm}\left(C_{ref}\right) - u_{NGB}^{norm}\left(C_{comp,i}\right)$$

The difference values (F) between the reference candidate and each comparison candidate for all output objectives are stored to determine the ranking of the reference candidate. The purpose of converting the output objectives to a minimisation problem then normalising their values is to aid in the determination of reference candidates: Superiority (See Definition 2.3.3) or Inferiority (See Definition 2.3.2) with respect to the comparison candidates.

$$r_{i=1}^{P} = \begin{cases} 1 & \text{if } F_i > 0 \\ 0 & \text{if } F_i < 0 \end{cases}$$

By normalising the output objectives into the range $[0, 1]^d$, for a minimisation problem the PF cannot exist lower than zero. Therefore, for each output objective if the difference value is positive with respect to that output objective, then the reference candidate is inferior to the comparison candidate. Whereas, if the difference value is negative, it is superior with regard to that output objective. Hence, the reference candidate is dominated by a comparison candidate in the event of when the difference values for all output objectives are positive. Whereas, in the event that a single outputs difference value is negative, then the reference candidate in regard to that comparison candidate are classified as non-dominate to each other.

As specified in Fonseca and Fleming [73] a non-dominated candidate has a Pareto rank of one, which is determined by totalling the number of comparison candidates that dominate each reference candidate using Equation 6.10.

$$P_r\left(C_i\right) = 1 + \sum_{i=1}^{P} r_i \tag{6.10}$$

The Pareto ranking's range from one to the size of the GA population (P) being evaluated, with rankings of candidates sorting in ascending order. In order to implement the chosen mating selection strategy (See Section 3.3.2) a single assigned fitness value is required. In Fonseca and Fleming [73] this procedure was referred to as 'Average Fitness' which assigned an identical fitness to members within the same Pareto rank as shown in Equation 6.11.

$$\Upsilon\left(C_i\right) = P - \sum_{j=1}^{P_r(C_i)-1} u\left(r_j\right) - \frac{1}{2}\left(u\left(r_i\right) - 1\right) \tag{6.11}$$

where, $\Upsilon\left(C_i\right)$ is the average fitness for GA individual C_i, P is the population size of the GA, $u\left(r_i\right)$ is the total number of individual classified in $r = i$, and $u\left(r_j\right)$ is the total number of individuals classified from $r = 1$ to $r = i - 1$.

Whereby, if there are 10 individuals in $P_r = 1$ then the $u\left(r_1\right) = 10$ or 3 individuals of $P_r = 2$ then the $u\left(r_2\right) = 3$. Thus, for each individual in the next lower Pareto rank, the total number of indivudals up until that rank are summed $u\left(r_j\right) \quad \forall j = 1, \ldots, r_i - 1$ and subtracted from the assignable Average Fitness of that next rank.

This is repeated until an average fitness ($\Upsilon(.)$) is assigned for individual (C_i) in the GA population. This ensured individuals in the same Pareto ranking would have an equal opportunity of being selected during the mating selection step. Following this updated MOBBO procedure, the standard GA methodology can proceed according to Section 3.3, with the exception of a modification required for the Elitism operator.

6.2.4 MOBB-DoE Modified Elitism Operator

In Chapters 4 and 5, the elitism operator followed Algorithm 3.1. In order to facilitate multiple output objectives, the elite set update approach requires further modification for use in conjunction with MOGA.

Previously, new elite set candidates were selected from the current GA population by identifying any individuals whose acquisition value was greater than the current elite set. However, in MOBO each candidate has an acquisition value for each output objective which prevents the identification of new elite set candidates in MOPs using the same methodology.

To alleviate this issue the Pareto ranks of individuals in the GA population will be utilised. Whereby, once the Pareto ranks are determined, individuals with $P_r = 1$ or non-dominated individuals, are identified as new elite set candidates. These new elite set candidates are then placed with the current elite set, which undergo a second Pareto Ranking. The secondary Pareto ranking is performed as the elite population are held as an external archive set and therefore, there is no guarantee of all elite set candidates being non-dominant with each other.

Following the secondary Pareto ranking, the combined elite set is reduced to only $P_r = 1$ or non-dominated elite set candidates, which can proceed in the same methodology as Algorithm 3.1. Another modification is required when the size of the new elite set is greater than pre-specified elite population size defined in Equation 3.13.

In Algorithm 3.1, , when the new elite set size was greater than the pre-specified elite set size, the elite set was re-ordered into descending order of acquisition value prior to removal of the worst performing individuals. Whereas, in MOBO as $P_r = 1$ or non-dominated individuals are non-inferior to each other (See Definition 2.3.4), wherein each individual is as good as any other within the elite set. Thus, the new elite set cannot be sorted into descending order for the removal the worst performing individuals, as each individual is non-inferior to any other.

As the potential acquisition value of members of the new elite set cannot be used to order the set from best to worst candidate in terms of greater value. A different metric shall be utilised to improve the quality of the elite set through removal of excess candidates, one such measure is the diversity of candidates. This is achieved by utilising the similarity measure demonstrated in the K Nearest Neighbour (KNN) Algorithm [31] which utilises Euclidean Distance to determine clusters of individuals in search spaces.

Traditionally, KNN is a clustering algorithm whose primary purpose is to classify new data based upon the previously obtained data classifications using a similarity measure [31]. In this case, the similarity measure was used for the purposes of thinning the new elite set to the pre-defined elite set size, whilst ensuring the diversity within the set remained high by removing the individuals in closest proximity in the output objective space.

The similarity measure utilises Euclidean distance between two candidates to determine the distance between two points in space as shown in Equation 6.12.

$$\mathrm{Euc}_D(p,q) = \sqrt{\sum_{i=1}^{d} \left(p_i - q_i\right)^2} \tag{6.12}$$

where, $\mathrm{Euc}_D(.)$ is the Euclidean distance between two points in space, d is the number of dimensions, and p/q are two arbitrary points in this d-dimensional space.

Therefore, a systematic approach is required to iteratively determine which candidates of the new elite set are in closest proximity in the output space. Upon which, one candidate from the closest proximity pair should be removed from the new elite set. This approach will subsequently be repeated until the elite set has been thinned down to the pre-specified elite size described in Section 3.3.5.

Algorithm 6.1 MOBB-DoE Elite Set Update

```
Inputs:
    $P$ : GA Population                $u$ : Acquisition value
    $E$ : Elite Population             $P_r$ : Pareto Ranks
    $E_{size}$ : Elite Set Size        $\mathbf{o}$ : Optimisation options

$P \leftarrow$ Constraint Penalty (P)                                  // See Equation 3.14
$E \leftarrow$ Constraint Penalty (E)                                  // See Equation 3.14
procedure MOO Elite Set Updater($P, E, E_{size}, u, P_r, \mathbf{o}$) do
    $E_{potential} \leftarrow \text{find}\big(P\big(P_r = 1\big)\big)$
    $E_{new} = \big[E_{old}, E_{potential}\big]$
    $P_{r,new} \leftarrow$ Pareto Ranking$\big(E_{new}, \text{Size}(E_{new})\big)$      // See Section 6.2.3
    $E_{new} \leftarrow$ Removal$\big(E_{new}\big(P_{r,new} \neq 1\big)\big)$          // Remove non $P_r = 1$ elites
    $E_{new} \leftarrow$ Unique Sorting$\big(E_{new}\big)$                   // Identical candidate removal
    if Size($E_{new}$) > $E_{size}$ then
        repeat
            $E_{close} \leftarrow \text{KNN}\big(E_{new}, u\big)$              // Find elites in closest proximity
            $Dist_{close} \leftarrow Euc_D\big(E_{close}, E_{new}\big)$   // Identify $E_{close}$ with shortest $Euc_D$
            $E_{new} \leftarrow$ Removal$\big(E_{close}\big(Dist_{close}\big)\big)$
        until Size($E_{new}$) = $E_{size}$
    end if
    return New external elite population, $E_{new}$
end procedure
```

This methodology is demonstrated in Algorithm 6.1, which systematically iterates through new elite set candidates to determine which combination of candidates have the closest proximity. After which the short-listed candidates are compared with the remaining candidates to determine which of the two would be in close proximity in the output space with the remaining elite set members. This is to ensure that the new elite set determined will have the greatest diversity in order to improve the diversity through the generations of the GA. The systematic assessment of the new elite set candidates and the excess removal is iterated through until the new elite set size matches the pre-specified elite size described in Section 3.3.5.

6.2.5 Uncertainty Maximisation

As discussed in Section 6.2.1 a PD-MOEA was selected as the MOO solver to generate a POS that represents a trade-off between output objectives on a PF. Upon the completion of the PD-MOEA within its population a POS will have been generated from which a single candidate must be chosen, so as to remain a non-interactive methodology. In order to select the best candidate from the POS, a secondary selection mechanism is needed.

Traditionally, in the MOBO literature as shown in Section 6.1.1 and 6.1.2, the primary mechanism for selecting a single best candidate from the POS is either implemented through the incorporation of the metric into the MOO solver or as a stand-alone mechanism. Previously, as discussed in Section 6.2.1, the Hyper-volume metric was shown to be a popular choice. However, this metric was formerly discounted for use within the MOO olver due to scaling poorly beyond three output objectives as well as being computationally expensive [102] [20]. For these disadvantages it will also be a poor choice as a secondary selection mechanism for choosing between members of the POS for the best candidate for experimentation.

An alternative measure was introduced in Belakaria et al. [23] which utilised the volume of a hyper-rectangle which is constituted of the upper and lower confidence bounds surrounding a candidate, as defined in Equation 6.13.

$$\mathrm{U}(\mathbf{x}) = \prod_{\mathrm{i}=1}^{\mathrm{M}} \mathrm{CI}\left(\mathrm{f}_{\mathrm{i}}(\mathbf{x}), \mathbf{x}, \beta_{\mathrm{t}}\right) \tag{6.13}$$

where, $\mathrm{U}(\mathbf{x})$ is the uncertainty hyper-rectangle for candidate $\mathbf{x}$, and $\mathrm{CI}(.)$ is the confidence interval function. $\mathrm{f}_{\mathrm{i}}(\mathbf{x})$ is Gaussian Process model for output i, $\mathbf{x}$ is a GA candidate, and β_{t} is the time varying exploration/exploitation trade-off parameter.

The volume of a hyper-rectangle is determined by taking the product of all the edge lengths, which in this context refers to the size of each confidence bound for each output objective as shown in Equation 6.14

$$\mathrm{CI}\left(\mathrm{f_i}\left(\mathbf{x}\right),\mathbf{x},\beta_\mathrm{t}\right)=\mathrm{u_{GP-UCB}}\left(\mu_\mathrm{i}\left(\mathbf{x}\right),\sigma_\mathrm{i}\left(\mathbf{x}\right),\beta_\mathrm{t}\right)-\mathrm{u_{GP-LCB}}\left(\mu_\mathrm{i}\left(\mathbf{x}\right),\sigma_\mathrm{i}\left(\mathbf{x}\right),\beta_\mathrm{t}\right) \quad (6.14)$$

where, $\mathrm{u_{GP-UCB}}(.)$ is the Gaussian Process Upper Confidence Bound (GP-UCB) and $\mathrm{u_{GP-LCB}}(.)$ is the Gaussian Process Lower Confidence Bound (GP-LCB). $\mu_\mathrm{i}(\mathbf{x})$ is the predictive GP mean of output i and $\sigma_\mathrm{i}(\mathbf{x})$ is the predictive GP standard deviation of output i.

In order to incorporate the uncertainty maximisation metric for use within MOBB-DoE, the methodological approach requires modification for the determination of the volume of the hyper-rectangle for a batch set. Therefore, in order to determine the volume, the upper and lower confidence bounds must first be calculated. As discussed in Section 5.2.1 the predictive variance of each batch set member should be conditioned upon the assumption, this is that each remaining batch set member has already been observed.

By doing so the predictive variance will reduce for the batch set members that are in close proximity of each other, which similarly will affect the size of their confidence bounds. The updated predictive variance will then be used to determine for each batch set member their corresponding upper and lower confidence bounds in Equation 2.29. Hence each batch set member will have a confidence bound for each output objective determined by using Equation 6.14.

In order to determine the total volume of the uncertainty for the batch set, each individuals' batch set hyper-rectangle is determined according to Equation 6.13. This is achieved by taking the product of each confidence bound or "edge" of each output objective as illustrated in Figure 6.3. This produces a single volume for each member of the batch set which represents their uncertainty volume. By summing each members' volume, the total volume of the batch set can be identified which can subsequently be used to select which batch candidate is best from the POS.

$$\mathrm{U}\left(\mathrm{B_t}\right)=\sum_{\mathrm{i}=1}^{\mathrm{Q}}\mathrm{U}\left(\mathbf{x}_{\mathbf{t+i}}\right)$$

In summary, Algorithm 6.2 is a novel approach which utilises the novel NGB-GP-CB acquisition function using a PD-MOEA optimiser and secondary selection for optimal candidates. Whilst PD-MOEA based MOBO have been implemented previously in [113] [140] [139], these methods were focused on EI or kriging variance and not GP-CB or batch selection methodologies. The

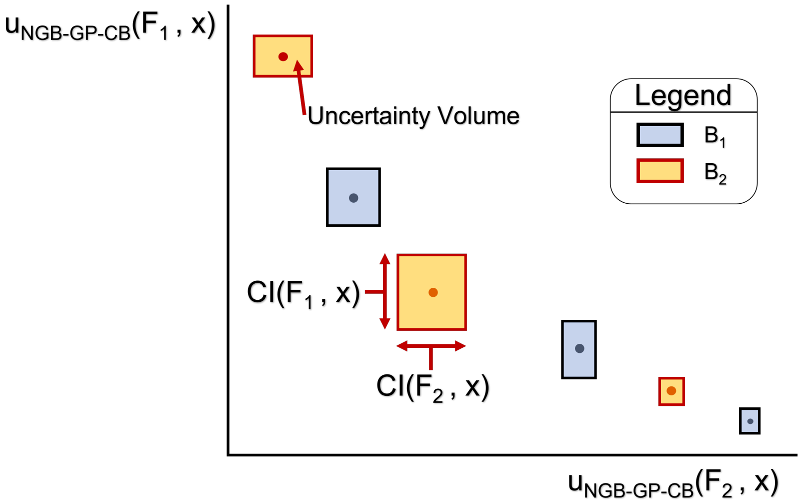

$u_{NGB\text{-}GP\text{-}CB}(F_1, x)$
Uncertainty Volume
Legend
B_1
B_2
$CI(F_1, x)$
$CI(F_2, x)$
$u_{NGB\text{-}GP\text{-}CB}(F_2, x)$

Figure 6.3: Illustration of the secondary selection step in MOBB-DoE algorithm: Uncertainty Maximisation. Wherein the confidence intervals for each output objective form the edges of a hyper-rectangle which when multiplied together determines their volume for each batch set member which are summed to find the total uncertainty volume of a batch candidate set.

modifications required to achieve this were developed and detailed in Sections 6.2.1 through 6.2.5.

Algorithm 6.2 MOBB-DoE Algorithm

Inputs:

n_s	:	LHS samples	$\mathbf{o}$	:	Optimisation settings
d	:	Number of inputs	Q	:	Batch Size
M	:	Number of outputs	N_B	:	Number of Batches
λ	:	Model Hyper-priors			

```
 1: $x_t \leftarrow$ Latin Hypercube Sampling$(n_s, d)$
 2: for $i = 1, 2, \ldots, M$ do
 3:     $y_t^i \leftarrow$ Evaluate$(x_t, i)$
 4:     $D_t^i \leftarrow \left[x_t, y_t^i\right]$
 5:     $\theta_t^i \leftarrow$ GPR$\left(\lambda, D_t^i\right)$                                  // See Algorithm 3.2
 6: end for
 7: for $t = 1, 2, \ldots, N_B$ do
 8:     $B_{t+Q} \leftarrow$ MOBB-DoE GA Optimiser$\left(D_t^{1:M}, GP(m, k|\theta_t^{1:M}), \mathbf{o}\right)$   // See line 15
 9:     for $i = 1, 2, \ldots, M$ do
10:         $y_{t+1:t+Q}^i \leftarrow$ Evaluate$\left(B_{t+Q}, i\right)$
11:         $D_{t+1:t+Q}^i \leftarrow D_t \cup \left[x_{t+1:t+Q}, y_{t+1:t+Q}^i\right]$
12:         $\theta_{t+1}^i \leftarrow$ GPR$\left(\lambda, D_{t+1:t+Q}^i\right)$                   // See Algorithm 3.2
13:     end for
14: end for

15: procedure MOBB-DoE GA Optimiser$\left(D_t, GP(m, k|\theta_t), \mathbf{o}\right)$ do
16:     Initialise GA Population $\left(P_1 = [C]\right)$
17:     for $G = 1, 2, \ldots, N_{Gen}$ do
18:         $u_{new} \leftarrow u_{NGB-GP-CB}\left(C_G, D_t, GP(m, k|\theta_t), \mathbf{o}\right)$       // See Algorithm 5.1
19:         $P_G \leftarrow$ Update Population $\left[C_G, u_{new}\right]$
20:         if $G = 1$ then
21:             $E \leftarrow \max_{i=1}^{E_{size}} P_G$
22:         else
23:             $E_{New} \leftarrow$ Elite Set Updater$(P_G, E, E_{size})$                  // See Algorithm 6.1
24:         end if
25:         $P_G \leftarrow$ Constraint Evaluation$\left(P_G\right)$                       // See Section 5.2.4
26:         $\left[P_r, u\left(r\right)\right] \leftarrow$ Pareto Ranking$\left(P_G\right)$                     // See Section 6.2.3
27:         $\Upsilon\left(P_G\right) \leftarrow$ Average Fitness$\left(P_r, u\left(r\right)\right)$                   // See Section 6.2.3
28:         $Par_G \leftarrow$ Tournament Selection$\left(\Upsilon\left(P_G\right)\right)$                 // See Section 3.3.2
29:         $O_G \leftarrow$ Blending Crossover$\left(Par_G\right)$                       // See Section 3.3.3
30:         $M_G \leftarrow$ Random Mutation$\left(O_G\right)$                          // See Section 3.3.4
31:         $C_{G+1} \leftarrow O_G \cup M_G$
32:     end for
33:     return $B_{t+1} \leftarrow \arg\max_C u\left(P, E\right)$
34: end procedure
```

6.3 Simulation Study: Benchmarks

6.3.1 Introduction

In this chapter, a novel MOBB-DoE has been developed by incorporating the NGB-GP-CB developed in Chapter 5 with a PD-MOEA chosen according to the requirements set out previously, with a secondary optimisation step: Uncertainty maximisation developed by Belakaria et al. [23] to select each batch of candidates to add to the Pareto Front (PF). In Chapters 4 and 5, the assessment of performance was carried out on a series of synthetic benchmark functions selected for their varied optimisation landscape characteristics detailed in Section 4.3.

Although these benchmark functions are no longer suitable for use when assessing the performance of the MOBB-DoE algorithms which requires multiple conflicting output objectives. Therefore, in order to assess performance of the developed MOBB-DoE an alternative benchmark function will be used: Binh and Korn function [30].

Binh and Korn Test Function

The Binh and Korn function is a 2-Dimensional constrained synthetic benchmark function with two minimisation outputs, shown in Equation 6.15 [30]:

$$\text{Minimise} = \begin{cases} f^1\left(x_1, x_2\right) & = 4x_1^2 + 4x_2^2 \\ f^2\left(x_1, x_2\right) & = \left(x_1 - 5\right)^2 + \left(x_2 - 5\right)^2 \end{cases} \tag{6.15}$$

The Binh and Korn function is a constrained problem with the following inequality constraints:

$$\text{Constraints} = \begin{cases} g^1\left(x_1, x_2\right) & = \left(x_1 - 5\right)^2 + \left(x_2\right)^2 \leq 25 \\ g^2\left(x_1, x_2\right) & = \left(x_1 - 8\right)^2 + \left(x_2 + 3\right)^2 \geq 7.7 \end{cases}$$

The Binh and Korn functions design space is located in the domain $x_1 = [0, 5]$ and $x_2 = [0, 3]$. The experiments that satisfy the constraints and are $P_r = 1$ are contained in the POS which make up the true PF in the objective space shown in Figure 6.4.

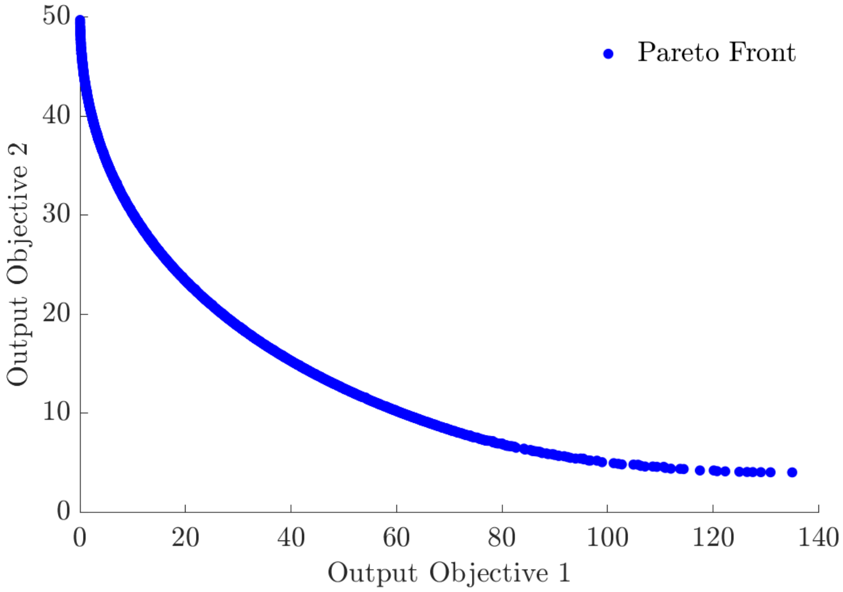


Figure 6.4: Representation of the Binh and Korn functions True Pareto Front.

6.3.2 Experiment Software

Algorithm 6.2 and the majority of functions have been self-coded in MATLAB 2017b in Windows 10 using their respective literature sources unless otherwise stated. The code was run on a personal PC with the following specifications: AMD Ryzen 5 1600 6 core processor with 12 threads and 24 GB of RAM of which all were used to run Algorithm 6.2 according to the details specified in Table 5.1. In this study no existing libraries or repositories were used.

6.3.3 Experiment Details

Similarly, in Chapters 4/5 the experimentation budget to assess the performance of the algorithms of interest were set to be a reduced total experimentation budget of 50 experiments to provide an accurate representation of an expensive to evaluate AM problem whilst also providing sufficient data to fit GP surrogate models and locate optimum solutions in this case a diverse and accurate POS. Therefore, as the core mechanisms of the MOBB-DoE framework utilise the BB-DoE framework but expanded for multiple objectives on the Binh and Korn test function, the simulation study settings will be identical to those in Table 5.1.

Previously, in order to assess exploitative performance, the Regret metric was utilised as detailed in Section 3.2.1. However, as shown in Figure 6.4

each experiment located in the POS on the PF is as optimal as any other member. This poses a problem in determining the exploitative performance of the developed MOBB-DoE as Regret requires the comparison against a single global optimum as shown in Equation 3.3.

Therefore, an alternative metric is required to assess exploitative performance. Typically, in the MOO literature the most popular performance metric is the Hyper-volume metric [183]. Whilst previously, in Section 6.2.1 he Hyper-volume indicator or metric have not been implemented due to being expensive to compute [179] [176] [64] [50] [51] and poor scalability [102]. Hence, an alternative metric that measures the accuracy of the optimised PF that is frequently applied within the MOO literature is the Generational Distance (GD) [183].

Generational Distance

The GD is a measure which calculates the distance between a reference set (selected experiments) and the true PF, as shown in Equation 6.16.

$$\mathrm{Gen_d} = \frac{1}{|\mathrm{POS}|}\sum_{i=1}^{|\mathrm{POS}|} \mathrm{Euc_d}\left(\mathrm{POS_i}\right) \tag{6.16}$$

Where, |POS| is the size of the reference POS comprising the experiments of $\mathrm{P_r} = 1$ selected by the MOBB-DoE algorithm.

$$\mathrm{POS} = \left[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{|\mathrm{POS}|}\right]$$

Where, $\mathbf{y}$ is the vector set of output objectives for each selected experiment in the reference set. In order to calculate each $\mathrm{Euc_d}$ the closest point on the true PF must be chosen for calculation. The experiment on the true PF closest to the reference point of interest is chosen using a K Nearest Neighbour (KNN) algorithm, after which the Euclidean Distance is calculated.

Although, GD is a frequently implemented metric within the MOO literature [183] this is primarily due to its simple formulation allowing for quick implementation. GD provides a good approximation of how far the current best solutions are from approaching the true PF but this characteristic also links to its limitation in it only considers the accuracy of the POS in locating the PF, but doesn't consider the diversity.

This is exemplified in a situation in which a single solution is quickly placed on the PF, as this would give a GD of 0 suggesting convergence to the true PF [11]. Therefore, by using GD the is a risk of poor algorithmic performance being mis-represented due to a single solution locating the PF leading to a small GD. It is due to these characteristics that Hyper-volume indicators or Inverted Generational Distance (IGD) which may be more difficult to calculate but represent both accuracy and diversity of the PF that they are so prevalent in literature.

6.3.4 Results and Discussion

The results of this simulation study are to present a novel implementation of a non-greedy MOBB-DoE algorithm on the performance metrics utilised throughout this thesis as well as the Generational Distance (GD) introduced in Section 6.3.3. The performance metrics are illustrated in Figures 6.5, 6.6, 6.7, 6.8 and 6.9.

Benchmark Results: PF Convergence

Figure 6.5 illustrates all of the experiments chosen via the MOBB-DoE algorithm in the output objective space. They occupy a broad range across the entire output objective space showing a wide diversity. The experiments chosen also seem to search both edges of the Binh and Korn benchmark function towards $y^1 < 50$ and $y^2 < 130$, suggesting the MOBB-DoE algorithm can produce a diverse POS.

Of the 50 experiments shown in Figure 6.5, if they were to be reduced to only their $P_r = 1$ experiments contained in the POS to show the experimental PF this is demonstrated in Figure 6.6. Over the experimentation budget (T) of 50 experiments, 27 experiments are $P_r = 1$ suggesting a 54% success rate of selecting non-dominated experiments using the non-greedy MOBB-DoE algorithm.

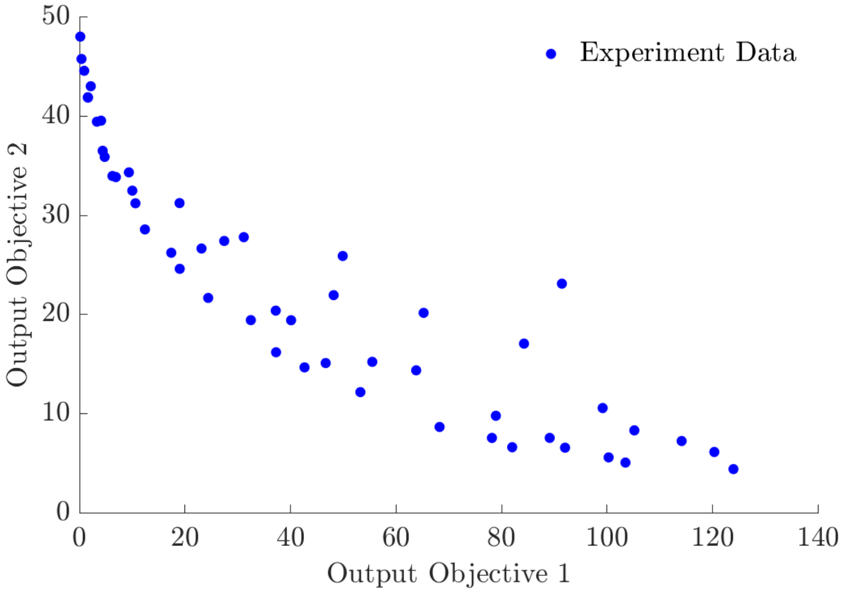


Figure 6.5: Visual representation of experiment data in the output objective space chosen using MOBB-DoE algorithm on the Binh and Korn test function and was run according to the settings in Table 5.1.

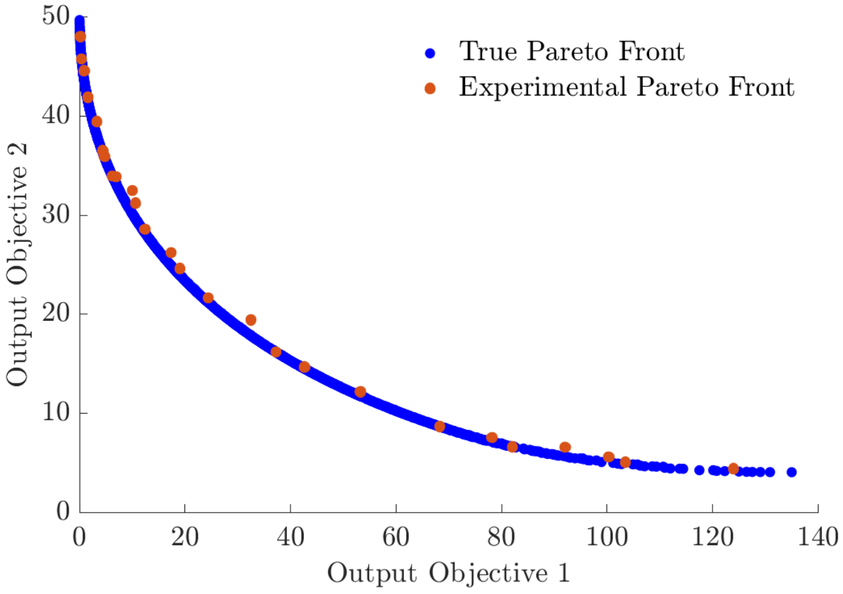


Figure 6.6: Visual comparison between the true PF and the $P_r = 1$ experimental data to illustrate distribution and diversity of experiment chosen using MOBB-DoE algorithm on the Binh and Korn test function and was run according to the settings in Table 5.1.

As can be seen, the experimental PF found by the MOBB-DoE algorithm is representative of the true PF. A diverse spread across the entire PF is achieved locating experiments at the boundaries of the output objective space as well as uniformly spaced along the PF. Although it can also be seen that more experiments are placed in areas where $y^1 < 60$ and $y^2 > 10$.

Benchmark Results: Exploration

Previously in Figures 6.5 and 6.6 a diverse POS was demonstrated which can be achieved by using MOBB-DoE, but this may not correspond to accurate surrogate models of the respective design spaces. Although as shown in Figure 6.7, for both output objectives an accurate and representative surrogate model is achieved with NRMSD < 0.1 % by the 8th batch.

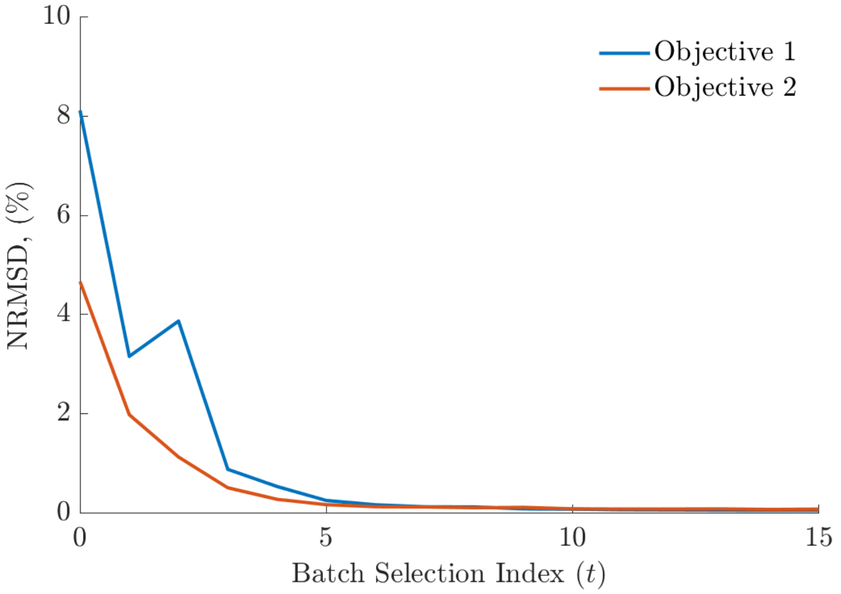


Figure 6.7: NRMSD performance metric using the MOBB-DoE algorithm on the Binh and Korn test function for both output objective 1 and 2. The MOBB-DoE algorithm was run according to the settings in Table 5.1.

This aligns with the performance seen on SOPs in Chapters 4/5 for 2-Dimensional DoE problems. This suggests that surrogate model accuracy is maintained even when multiple objectives are optimised simultaneously.

Benchmark Results: Exploitation

As previously mentioned in Section 6.3.3 the GD metric is a replacement for the Regret used throughout the thesis and the GD for the Binh and Korn function is shown in Figure 6.8. Initially the GD begins to increase over the first 3 batches, after which the GD slowly begins to reduce with each batch set of experiments performed, plateauing at 0.5.

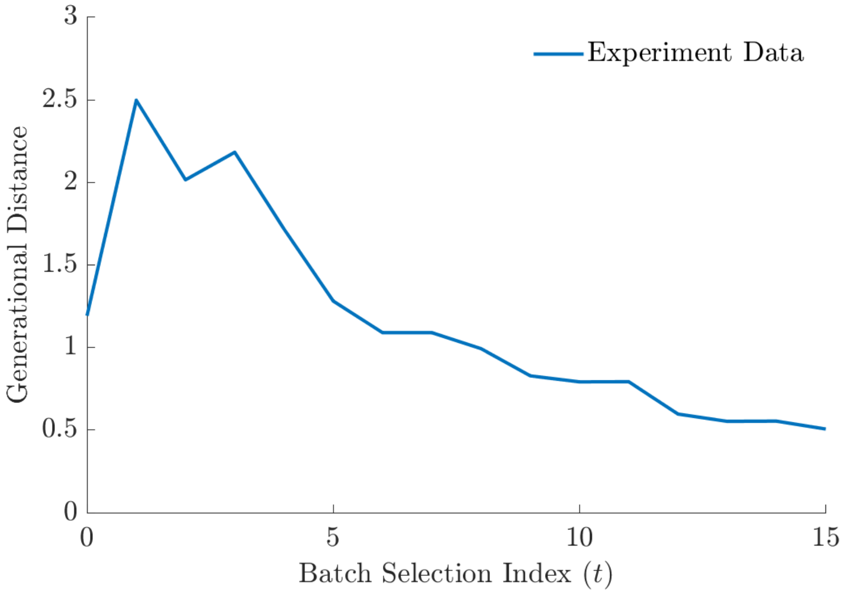


Figure 6.8: Generational Distance (GD) performance metric using the MOBB-DoE algorithm on the Binh and Korn test function and was run according to the settings in Table 5.1.

An increase in the GD suggests that the average distance between the performed experiments and the true PF is increasing, such as experiments appearing farther from the true PF. However, as the rise in GD occurs during the initial batches this is most likely caused by exploration of the design space. As the MOBB-DoE proceeds and more experiments are performed this initial divergence from the true PF is minimised by more experiments close to the true PF and lowering the impact of outlier experiments.

Benchmark Results: Stopping Criterion

Similarly to B-DoE and BB-DoE, the MOBB-DoE algorithm shows a steady decrease in the stopping criterion with each batch as shown in Figure 6.9. Based upon the degree of reduction demonstrated by the stopping criterion, the NRMSD and GD, an optimal stopping point could be in the 7/8th batch.

In Figures 6.10 and 6.11 the experimental data is split between the first 7 batches and last 8 batches to illustrate the differences in distribution of experiments in the objective space.

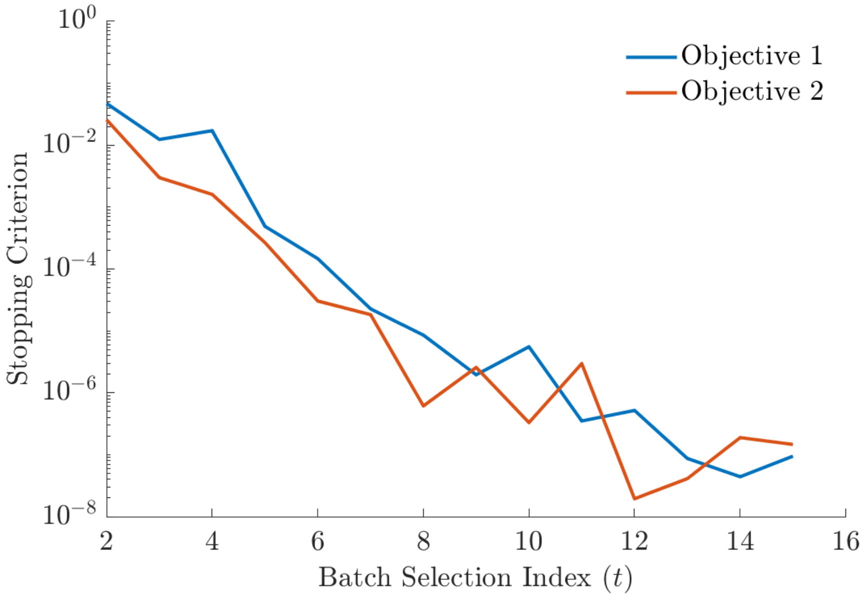


Figure 6.9: Stopping criterion determined for the MOBB-DoE algorithm on the Binh and Korn test function and was run according to the settings in Table 5.1.

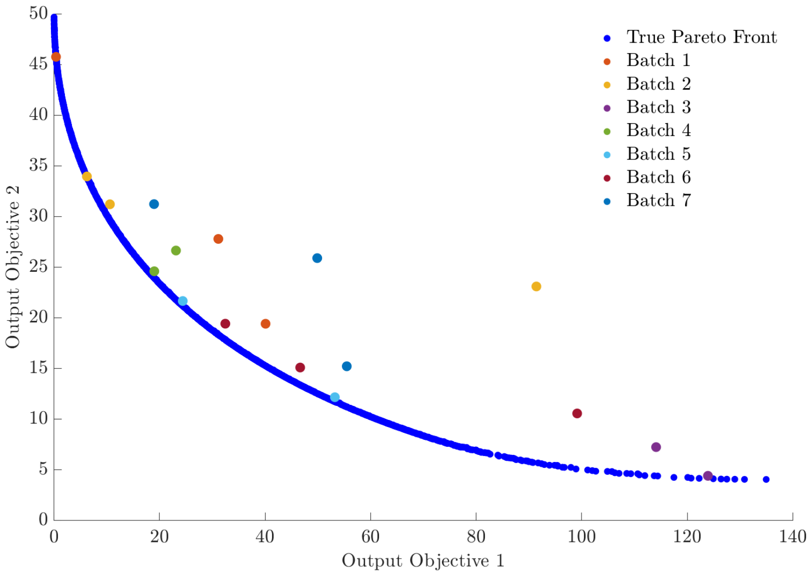


Figure 6.10: Visual comparison between the true PF and the first 7 batches of experimental data to illustrate distribution and diversity of experiment chosen using MOBB-DoE algorithm on the Binh and Korn test function and was run according to the settings in Table 5.1. Experiments selected are split into the batch order they were selected.

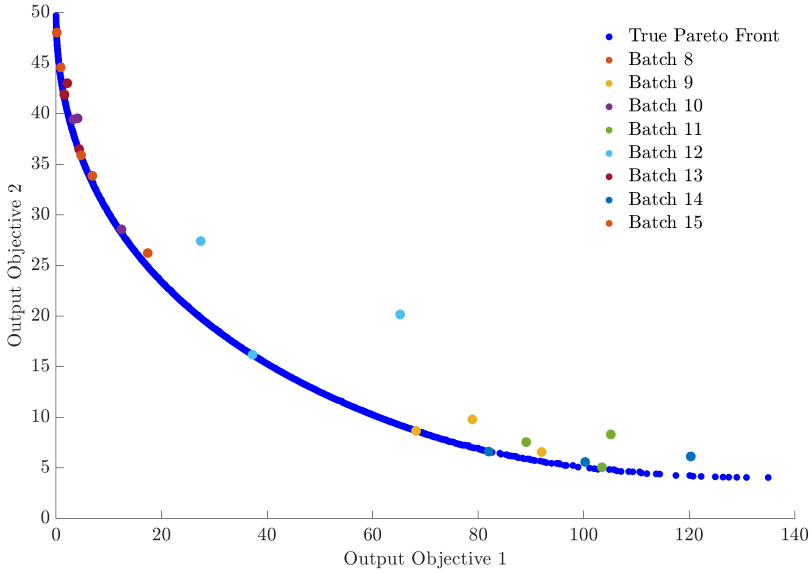


Figure 6.11: Visual comparison between the true PF and the last 8 batches of experimental data to illustrate the distribution and diversity of experiments chosen using MOBB-DoE algorithm on the Binh and Korn test function and was run according to the settings in Table 5.1. Experiments selected are split into the batch order they were selected.

As can be seen in Figure 6.10, there are a few experiments chosen within the batch sets are placed in the POS on the true PF. Although the majority of experiments performed are dominated solutions they are exploratory in nature and improve the surrogate models accuracy resulting in good performance for the stopping criterion observed in Figure 6.9 and in the NRMSD in Figure 6.7.

On the other hand as can be seen in Figure 6.11, a large proportion of the experiments performed are located on the true PF and explore the boundaries of the true PF. If the stopping criterion would end the MOBB-DoE algorithm early these POS experiments would not be performed and the boundaries of the true PF explored.

6.4 Conclusion

A MOBB-DoE algorithmic framework has been developed in this chapter. The MOBB-DoE algorithmic framework can be sub-divided into three main components: the non-greedy NGB-GP-CB acquisition cost function developed in Chapter 5, a PD-MOEA Multi-Objective Optimiser, and an uncertainty maximisation approach taken from Belakaria et al. [23] modified for batch selection. The developed MOBB-DoE algorithm was then applied onto a synthetic benchmark problem to assess its capabilities in locating a diverse, accurate, and spread POS in close proximity to the true PF.

The overall performance of the MOBB-DoE algorithm on the Binh and Korn synthetic test function suggested acceptable performance as a representative POS which was spread across the true PF and accurate GP surrogate models were developed within the experimentation budget. As the members of the POS were spread along the entire PF reaching both boundary sections of the objective space with POS members reasonably distributed along the entire PF. Also, of the 50 experiments selected via the MOBB-DoE algorithm, 54 % were non-dominated experiments, which as shown in Figure 6.6 were close to the true PF demonstrating an efficient experiment selection process.

In regard to exploratory performance, accurate surrogate models were found within 8 batches for both output objectives which is consistent with the performance of the BB-DoE algorithm on similar 2-Dimensional problems. Unlike in previous chapters which utilised Regret to determine exploitative performance it was no longer a suitable performance metric as on MOPs there is no single global optimum but rather a POS constituting the PF. Therefore, the assess the exploitative performance the Generational Distance (GD) metric was introduced and found slow convergence in performance with each successive batch added. Although initially the GD began to increase this was likely due to the low experimental data volume and exploratory experimentation leading to an initial greater GD for 2-3 batches. Thus, with each batch of experiments placed closer to the true PF this led to a gradual lowering of the GD with each successive batch after the 3$^{\text{rd}}$.

An auxiliary goal of this chapter was to further investigate the validity and performance of using the stopping criterion developed in Chapter 5 onto MOPs. It was shown that the stopping criterion had quick convergence towards the thresholds previously tested in Chapter 5, and far surpassing them as the experiments continued. However, upon further examination if the MOBB-DoE algorithm were to be stopped early under the recommendation of the stopping criterion, it would result in a much greater reduced POS as shown when comparing Figure 6.10 and 6.11.

Hence, the stopping criterion determines when a good model predictive performance is found but as previously demonstrated it can occur whenever the model improvement plateaus. Also, the stopping criterion does not indicate good exploitative performance as well as when a representative POS is found. Therefore, further modifications are required prior to implementation of the stopping criterion to make it suitable for inclusion in either a BB-DoE or a MOBB-DoE framework.

The MOBB-DoE algorithm presented a novel approach for a non-greedy batch DoE optimisation of MOPs that can efficiently select experiments in a batch format to find a representative PF with experiments spread evenly along the entire PF. However, in this work a limited set of synthetic benchmark functions were applied with no application onto a real-world AM application to demonstrate its performance onto a wider range of problems. Furthermore, the proposed stopping criterion assessed suggested good capabilities in determining accurate surrogate models during the optimisation but was shown to require further research. For discussion and reflection of research decisions, limitations and impacts refer to Section 7.2.

Chapter 7

Conclusion

7.1 Summary and Conclusions

The central theme of this thesis has been the development of suitable algorithms for Design of Experiments or Experimental Design (DoE) for implementation on various Additive Manufacturing (AM) processes. The DoE formulations included: sequential, batch, and multiple conflicting outputs. The requirements for the developed algorithms were to minimise the total number of experiments whilst simultaneously produce accurate surrogate models and optimise process parameter settings to locate the global optimum. Initially the work was exploratory in nature investigating the Bayesian Optimisation (BO) literature as a suitable DoE scheme for expensive AM DoE problems, prior to algorithm development onto desired problem types needed in the additive manufacturing field.

The main contributions of this work can be split into three sections: the implementation of a novel BO framework onto the an expensive AM DoE case study [246], and the development of two novel algorithms: a non-greedy Batch Bayesian Experimental Design Optimisation (BB-DoE), and a non-greedy Multi-Objective Batch Bayesian Experimental Design Optimisation (MOBB-DoE). Additionally, a useful stopping criterion was derived over the course of the thesis to assist in the assessment of the developed algorithms to evaluate their exploratory performance.

Bayesian Optimisation (BO) was suggested as an alternative data-efficient DoE methodology for expensive manufacturing in Chapter 4. It was shown for sequential experiment selection that on both synthetic benchmark functions as well as the novel application onto an expensive additive manufacturing problem, it significantly outperformed traditional DoE in both the generation of surrogate models and locating the global optimum. Further analysis of the results demonstrated differences in exploration and exploitation for each BO acquisition cost function coupled with the analysis of their theoretical properties, discerning differences in their potential as B-DoE approaches.

The analysis showed MES demonstrated greater exploration whilst EI exhibited improved exploitation and GP-CB showed a balance between the two. Although in conclusion it was shown that the MES would be an overly expensive acquisition cost function if extended to batch selection [212] and EI tended to overly exploit leading to poor exploration performance [26]. Subsequently, the GP-CB acquisition cost function was selected to form the basis for the novel extensions for expensive AM DoE, non-greedy batch selection.

The second contribution of this thesis was the development of the Non-Greedy Batch GP-CB (NGB-GP-CB) acquisition function: incorporated into a GA optimisation framework to produce a novel BB-DoE algorithm. Its performance was shown to outperform current state of the art greedy BB-DoE algorithm: GP-BUCB, whilst having comparable performance to a non-greedy BB-DoE algorithm: PPES.

The novel non-greedy BB-DoE algorithm using NGB-GP-CB was also implemented onto a blind AM case study which sought to determine the process parameter settings: laser power, nozzle velocity and hatch spacing which optimised the micro structural property Dendritic Arm Spacing (DAS) for a Directed Energy Deposition (DED) process. The optimal process parameter settings were found within three batches of a ten-batch experimentation budget. Whilst also selecting experiments in the $4.5 - 9.0\mu$ m DAS region during its subsequent batches after the optimum was identified, which had previously unexplored. The process interactions leading to part formation with DAS in this region have now been identified.

The third contribution to the thesis was the extension of the previously developed non-greedy BB-DoE algorithm onto Multi-Objective Problems (MOP) resulting in the Multi-Objective Batch Bayesian Experimental Design Optimisation (MOBB-DoE) algorithm. The algorithm developed utilised the previous NGB-GP-CB acquisition cost function with a PD-MOEA optimiser to generate batches of potential experiments that optimise multiple conflicting objectives simultaneously. The non-dominated batches of potential experiments then required a secondary selection mechanism to select a single batch during each iteration of the MOBB-DoE algorithm, a batch modified uncertainty maximisation operator which was inspired by [23] and adapted for multi-objective use. Whereby, each batch would select experiments with the purpose of generating an accurate, spread and diverse Pareto Optimal Set (POS) that lie on the true Pareto Front (PF) providing a decision maker a wide selection of optimal process parameter settings to choose from. The evaluation on the synthetic benchmark function demonstrated the MOBB-DoE algorithm was capable of generating the representative POS in a data-efficient scheme via a novel algorithmic approach.

This thesis also contributed towards to development of a stopping criterion as it is an important tool in assisting with the evaluation and early termination of expensive DoE problems. The stopping criterion was evaluated on both the BB-DoE algorithm and the MOBB-DoE algorithm, suggesting early termination in scenarios in which the model performance had plateaued.

The analysis and investigations carried out in the thesis has shown that BO offers a beneficial algorithmic approach to expensive DoE in manufacturing, particularly for AM. Whereby, BO is a data efficient approach that maximises the information gained from each experiment whilst minimising the total number of experiments required to define accurate surrogate models and optimise the process parameter settings. It also provided a stable foundation for extension onto manufacturing specific DoE problems to be tackled with batch and multi-objective batch selection. In which both BB-DoE and MOBB-DoE have resulted in improved performance over both traditional DoE schemes as well as against current state-of-the art BO approaches in both exploration and exploitation.

7.2 Critical Reflections

Throughout the thesis there have been many decisions at various stages that have been made due to circumstances outside of my control as well as limitations caused by choices in the research direction that require discussion and reflection.

7.2.1 Impact of COVID-19

One of the primary limiters on the research performed during the thesis was the impact of COVID-19 pandemic which led to the lockdown of facilities specifically in universities. This led to issues regarding access to research collaborators, access to research materials and apparatus which led to the scaling back of intended research goals.

Firstly, the main problem that occurred for my research was the loss of access to the high-performance computing (HPC) clusters in the university for a few months which led to the loss of implementation time and subsequently required some of the simulation studies to be reduced. In the original goals for the research an additional synthetic benchmark was planned which had 6 input variables and constraints applied called the Hesse function [209]. This was planned to be implemented on both the B-DoE and BB-DoE algorithms in Chapters 4/5.

Secondly, a 2$^{\text{nd}}$ multi-objective synthetic benchmark was also planned, this function also only optimised two output objectives but had a multi-modal and more complex PF geometry to increase the computational complexity for the MOBB-DoE algorithm. This was intended to contrast against the simpler PF geometry of the Binh and Korn test function assessed in Section 6.3. Which would have provided a more balanced analysis of the MOBB-DoE algorithms performance.

Thirdly, the manufacturing case studies investigated in this thesis had their experimentation budget delayed due to inability to access research apparatus for research collaborators or case studies such as the intended manufacturing case study for Chapter 6 removed. Finally, in order to implement all of the synthetic benchmarks within the reduced time frame for research the intended experimentation budget for investigations had to be reduced in two aspects: the total number of experiments performed in one repeat (75 to 50) and the total number of repeats per problem (100 to 50).

7.2.2 Suitability of test function suite and case studies

In Section 4.3.4 a discussion on the rationale behind the choice of synthetic benchmarks was conducted and to assess how they related potentially to a AM DoE problem. One of the key characteristics that were varied among the synthetic benchmark problems was the dimensionality of the input, which varied from 2 – 4 input variables. Originally 4 synthetic benchmarks were considered for both Chapters 4/5 which also included a 6-D input synthetic benchmark.

These were chosen due to the initial literature review of engineering and AM DoE review papers examined in the literature [109] [47] which stated on average the number of factors considered were 2-6. Although in both Bayesian Optimisation (BO) and Industry 4.0 an area of increasing literature interest is high dimensional optimisation due to ever increasing number of controllable inputs and sensor data available to researchers. Thus, there seems to be a conflict in what is typically applied in literature and the increasing needs of AM manufacturers.

Therefore, from the standpoint of what is typically included within the literature [109] [47], the choice of synthetic benchmarks was correct, although unfortunately due to circumstance (see Section ??) the higher dimensional problem was removed. However, considering the current/future needs of AM DoE as well as Industry 4.0 where higher dimensional problems are valuable assessments, the choice of limiting the synthetic benchmarks to 2-6 input parameters may be a rather short-sighted implementation strategy. Yet, in Section 1.2 the goal of the work was not to investigate high dimensional problems so whilst limiting in applicability to AM problems, for this thesis I believe it was the correct choice.

The assessment of MOBB-DoE in this thesis was the most limited implementation with only one synthetic benchmark considered with only 2 inputs, 2 outputs and a simple convex PF geometry. To further assess the capabilities of the MOBB-DoE algorithm it would need to be fully assessed in a combination of 3 further aspects for Multi-Objective Optimisation (MOO): higher dimensional inputs, higher number of output objectives (3 or 4) and a more complex PF geometry such as discontinuous or mixture of convex/concave. Much alike its sequential and batch counterparts assessed in this thesis a higher dimensionality of the number of input dimensions would be required to assess its scalability. This was exemplified as necessary in Section 5.3.4 wherein as the dimensionality of the inputs increased the modelling performance began to decrease slightly. Also, the purpose behind implementing the MOBB-DoE on a greater number of output objectives is due to the primary disadvantage of PD-MOEA's, in which as the number of output objectives increase the number of non-dominated solutions within a population also increase reducing the selection pressure in the GA. Therefore, an investigation is required to assess how these scalability concerns progress and effect the MOBB-DoE algorithm as well as how both the dimensionality of input vs output objectives interact affecting performance. Finally, the complexity of the MOO problems can be affected by the shape of the PF geometry, for which a simple convex geometry was used to illustrate the MOBB-DoE algorithm capabilities. However, in order to full ascertain its capabilities on a variety of problems and determine its weaknesses more complex and varied geometries are required.

The other method of assessment used throughout the thesis were the novel applications of AM case studies on defect formation in Selective Laser Melting (SLM), and micro-structural property analysis in Directed Energy Deposition (DED). The SLM case study used to assess the B-DoE algorithm was a rather limiting case study as the experimentation had already taken place and no more experiments could be performed. This created an unusual opportunity in order to ascertain whether a more efficient DoE scheme could be implemented to select the same experiments in such a manner that obtained more information with fewer experiments. However, due to its formulation it led to issues in regard to how the modelling error was assessed. This was due to the fact that as the size of data was so small there was not enough data that could be set aside to be used as a testing set and no more experiments could be run to generate a test set. Therefore, the testing set was set of the remaining data set from which experiments were being selected from leading the model error inevitably converging to zero regardless of modelling performance. In order to improve the assessment in the SLM case study an external modelling test set of 5-10 experiments perhaps chosen with a LHD to cover the entire space would allow for thorough assessment of modelling performance.

On the other hand the implementation of the micro-structural property analysis in Directed Energy Deposition (DED) tackled in Chapter 5 was well implemented and achieved desirable results. The only drawback would be the lack of successive experimentation to the full experimentation budget of 10 batches to fully assess how the modelling performance and stopping criterion would progress. As reasonable convergence had occurred by the 6th batch, as a joint decision the case study was stopped early but, in retrospect it would have been desirable to run for the full budget to assess how the continued placing of experiments would affect the results including:

1. Exploration into the DAS region of $4.5 - 9.0\mu$m.
2. Finding process parameters for DAS larger than 6.9μm.
3. Assessment of impact of exploration on convergence of predictive uncertainty in the design space.
4. Assessment of further experimentation on the convergence of the stopping criterion.

Finally, whilst there were initially a MOO AM DoE case study planned it had been removed due to time and facility constraints due to COVID-19 (see Section ??). However, over the course of the DED case study a more applicable case study was identified. In the micro-structural property analysis in Directed Energy Deposition (DED) case study additional data was recorded for the structural build quality of designed parts which seemed to be in conflict with the DAS property. This presents an opportunity in a similar fashion the SLM case study in which previous case study data could be utilised to investigate an alternative methodology. Although as illustrated in the issues discussed earlier for the SLM case study, it would be more appropriate to implement the case study as an additional case using new experiments but potentially using the previous data as a modelling test set.

7.2.3 Methodological issues

Throughout the thesis the main topic for investigation was focused primarily on the acquisition functions implemented. In Chapter 4, the literature acquisition functions were re-assessed to determine which of the three: EI, GP-CB and MES had the most desirable properties for extension to a Batch Bayesian Experimental Design Optimisation (BB-DoE) and Multi-Objective Batch Bayesian Experimental Design Optimisation (MOBB-DoE) algorithm in Chapters 4/5 whilst accounting for both their exploratory and exploitative performances. However, by focusing only of the algorithmic development and their ability in achieving a desirable exploration and exploitative performance, certain scalability within the methodology were not assessed including:

1. Introduction of noise.
2. Varying GA parameters.
3. Effect of batch sizes
4. Alternative β_t parameter designs.
5. Effect of constraint settings.

As exemplified by the amount of the scalability that were not investigated another concern/limitation of the work in the thesis was the large number of tune-able parameters implemented. In Section 1.2 one of the guiding research aims was to create a simple and easy to implement DoE methodology for use on AM problems. However, as the research progressed in order to create a more generalised approach that could be applied onto a wider range of applications and problems, more and more functions with tune-able parameters were designed into the methodologies. Whereby, in order to achieve one goal led to an ironic conclusion in which to achieve satisfactory performance I had set aside another which in the process created an alternate problem to be addressed.

7.2.4 Computational costs

The final limitation to be considered in the thesis is the computational complexity of the implemented methods which inhibited a larger proportion of implementation due to time constraints preventing further investigations. The methodology that led to the greatest computational load within the thesis was the design of the model hyper-parameter tuning algorithm using K-fold cross-validation with a GA optimiser as detailed in Section 3.4.

A significant proportion of the B-DoE, BB-DoE, and MOBB-DoE algorithms implementation time per call was spent in the model hyper-parameter tuning which occurred between each experiment selection optimisation. This was due to the required matrix inversion in order to calculate the predictive mean and standard deviations for each hyper-parameter configuration for every fold shown in lines 8-11 in Algorithm 3.2. This repeated evaluation compounded with the aspect that the matrix inversion in GP scale with the number of samples/experiments, meant a large computational load was ever increasing with each experiment selected. This issue was then multiplicative when concerning the design of the MOBB-DoE algorithm, in which multiple GP models were developed for each output objective, all of which required individual tuning.

Alternative design choices could have severely reduced the computational time by using a different optimiser for the model hyper-parameter optimisation such as gradient-based optimiser since the kernel choice for the GP model was a squared exponential kernel which is differentiable. This could have reduced the computational time significantly, whereby increasing the proportion of the research that could have been dedicated to alternative benchmarks or case studies as mentioned in Section 7.2.2 as well as alternative investigations as mentioned in Section 7.2.3. Also, model choice could have relieved some of the computational burden by using a sparse GP to limit the impact of large data sizes or a multiple-output GP to limit the number of surrogate models to 1 during MOBB-DoE instead of M.

7.3 Future Research Avenues

The research performed throughout this work has covered a variety of topics as well as raising several potential new avenues for future research to be performed:

7.3.1 Algorithmic Development

- In Chapter 2, one research direction of interest for the thesis was to develop a Manufacturing DoE approach for high dimensional input data, due to increasing interest in the concept of *Big Data*. This is also an increasingly popular area of interest in the Bayesian Optimisation literature due to scalability issues beyond 10-20 input parameters [165]. Therefore, extending the developed algorithms or providing modifications to existing algorithm components to improve performance in high dimensions, would allow for greater diversity in problem application.

- Similarly, whilst Bayesian Optimisation is a sample-efficient approach its non-parametric surrogate model, Gaussian Processes, also suffers from scalability wherein it has cubic complexity with data size O(N) [143]. Examining alternative models or scalable GPs which retain the prediction quality whilst improving the scalability to larger data sets could improve performance on higher dimensional problems as well.

- Chapters 5/ 6 introduced a stopping criterion which was focused primarily on the stability of model predictions to suggest early termination of the optimisation when a well-defined model was identified. Despite providing a good initial performance, it has limiting use in Bayesian Optimisation which seeks to efficiently define a tuned surrogate model whilst find the global optimum. Further research into the development of a stopping criterion that balances the trade-off of exploration and exploitation in Bayesian Optimisation can provide a balanced stopping criterion.

- In Chapter 6, a key process involved during the PD-MOEA optimisation was the selection of the next batch of candidates using a secondary optimisation. Whilst the approach implemented a batch extended uncertainty maximisation approach [23] showing good performance, there has been limited research on this topic. Exploring alternative architectures and methods to tackle this non-interactive selection to improve diversity, spread and approximation of the true PF is under researched.

7.3.2 Application in Additive Manufacturing

- From the data acquired during the DED manufacturing case in Chapter 5 additional build quality data was also attained illustrating build accuracy/inaccuracy. When compared to the DAS there seemed to be better quality builds in experiments with a lower DAS, suggesting a conflict between the two objectives. This suggests potential for further application of the MOBB-DoE approach developed in Chapter 6, this could provide an opportunity to further enhance and refine the developed approaches on a broader range of additive manufacturing problems encouraging industry uptake.

Whilst the identified regions of future work are based upon future stages of theoretical improvements building upon the algorithmic development and broader use cases of additive manufacturing problems. There is of course significant potential for development in cases beyond additive manufacturing. This may include the automatic setup and configuration of the various tuneable parameters with both the BB-DoE and MOBB-DoE algorithms based upon different problem classifications to allow for quick implementation for manufacturers without the need for extensive knowledge of the underlying mechanisms of the algorithms.

Bibliography

[1] M. Abdolshah, A. Shilton, S. Rana, S. Gupta, and S. Venkatesh. Expected hypervolume improvement with constraints. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3238–3243, 2018.

[2] N.R. Aghamohammadi, S. Salomon, Y. Yan, and R.C. Purshouse. On the effect of scalarising norm choice in a parego implementation. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 1–15. Springer, 2017.

[3] D.G. Ahn. Directed energy deposition (ded) process: State of the art. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 8, 02 2021.

[4] K.A. Al-Ghamdi. *Improving the practice of experimental design in manufacturing engineering*. PhD thesis, University of Birmingham, 2011.

[5] A. Alaeddini and K. Yang. Adaptive sequential experiment methodology for response surface optimisation. *International Journal of Quality Engineering and Technology*, 1(1):40–61, 2009.

[6] B.S. Anderson, A.W. Moore, and D. Cohn. A nonparametric approach to noisy and costly optimization. In *ICML*, pages 17–24, 2000.

[7] R. Anitha, S. Arunachalam, and P. Radhakrishnan. Critical parameters influencing the quality of prototypes in fused deposition modelling. *Journal of Materials Processing Technology*, 118(1):385–388, 2001.

[8] J. Antony. *Design of Experiments for Engineers and Scientists*. Elsevier insights. Elsevier Science, 2014.

[9] S. Ashwin Renganathan, V. Rao, and I.M. Navon. Camera: A method for cost-aware, adaptive, multifidelity, efficient reliability analysis. *Journal of Computational Physics*, 472:111698, 2023.

[10] V.P. Astakhov. *Design of Experiment Methods in Manufacturing: Basics and Practical Applications*, pages 1–54. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[11] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon. Performance indicators in multiobjective optimization. *European Journal of Operational Research*, 292(2):397–422, 2021.

[12] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*, 425:75 – 103, 2012. Theoretical Foundations of Evolutionary Computation.

[13] J. Azimi, A. Jalali, and X. Fern. Hybrid batch bayesian optimization. *arXiv preprint arXiv:1202.5597*, 2012.

[14] J. Bader and E. Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, March 2011.

[15] D. Baeck, T. Fogel and Z. Michalewicz. *Evolutionary Computation 1: Basic Algorithms and Operators*. CRC Press, 2000.

[16] R. Bardenet and B. Kégl. Surrogating the surrogate: accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm. In T.J. J. Fürnkranz, editor, *27th International Conference on Machine Learning (ICML 2010)*, pages 55–62, Haifa, Israel, June 2010. http://www.machinelearning.org.

[17] B. Barracosa, J. Bect, H. Baraffe, J. Morin, G. Malarange, and E. Vazquez. Bayesian multi-objective optimization with noisy evaluations using the knowledge gradient. In *PGMO Days 2019*, 2019.

[18] B. Barracosa, J. Bect, H. Dutrieux Baraffe, J. Morin, J. Fournel, and E. Vazquez. Extension of the Pareto Active Learning Method to Multi-Objective Optimization for Stochastic Simulators. In *MASCOT-NUM Workshop on Stochastic Simulators*, Virtual conference, France, March 2021.

[19] T. Bayes. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s. *Philosophical Transactions*, 53: 370–418, 1763.

[20] S. Belakaria, A. Deshwal, and J.R. Doppa. Max-value entropy search for multi-objective bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. 2019.

[21] S. Belakaria, A. Deshwal, and J.R. Doppa. Information-theoretic multi-objective bayesian optimization with continuous approximations. *arXiv preprint arXiv:2009.05700*, 2020.

[22] S. Belakaria, A. Deshwal, and J.R. Doppa. Max-value entropy search for multi-objective bayesian optimization with constraints. *arXiv preprint arXiv:2009.01721*, 2020.

[23] S. Belakaria, A. Deshwal, N.K. Jayakodi, and J.R. Doppa. Uncertainty-aware search framework for multi-objective bayesian optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10044–10052, Apr. 2020.

[24] D.E. Bell. Regret in decision making under uncertainty. *Operations Research*, 30(5):961–981, 1982.

[25] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. 2011.

[26] J. Berk, V. Nguyen, S. Gupta, S. Rana, and S. Venkatesh. Exploration enhanced expected improvement for bayesian optimization. In M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, and G. Ifrim, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 621–637, Cham, 2019.

[27] J. Berk, S. Gupta, S. Rana, and S. Venkatesh. Randomised gaussian process upper confidence bound for bayesian optimisation. *arXiv preprint arXiv:2006.04296*, 2020.

[28] J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West. Optimization under unknown constraints. *Bayesian Statistics*, 9(9):229, 2011.

[29] N. Beume, B. Naujoks, and M. Emmerich. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653 – 1669, 2007.

[30] T.T. Binh and U. Korn. Mobes: A multiobjective evolution strategy for constrained optimization problems. In *IN PROCEEDINGS OF THE THIRD INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS (MENDEL97*, pages 176–182, 1997.

[31] C.M. Bishop and N.M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[32] T. Blickle and L. Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evol. Comput.*, 4(4):361–394, December 1996.

[33] G. Blondet, J. Le Duigou, N. Boudaoud, and B. Eynard. Simulation data management for adaptive design of experiments: A litterature review. *Mechanics & Industry*, 16(6):611, 2015.

[34] A. Botchkarev. A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management*, 14:045–076, 2019.

[35] I. Boussaïd, J. Lepagnot, and P. Siarry. A survey on optimization metaheuristics. *Information Sciences*, 237:82–117, 2013. Prediction, Control and Diagnosis using Advanced Neural Computations.

[36] G.E.P. Box and K.B. Wilson. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(1):1–45, 1951.

[37] E. Bradford, A.M. Schweidtmann, and A. Lapkin. Efficient multiobjective optimization employing gaussian processes, spectral sampling and a genetic algorithm. *Journal of global optimization*, 71(2):407–438, 2018.

[38] F.H. Branin. Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *IBM Journal of Research and Development*, 16(5):504–522, 1972.

[39] M. Brennan, J. Keist, and T. Palmer. Defects in Metal Additive Manufacturing Processes. In *Additive Manufacturing Processes*. ASM International, 06 2020.

[40] E. Brochu, V.M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR*, abs/1012.2599, 2010.

[41] D. Brockhoff and E. Zitzler. Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In *2007 IEEE Congress on Evolutionary Computation*, pages 2086–2093, Sept 2007.

[42] D. Brockhoff, T. Wagner, and H. Trautmann. 2 indicator-based multiobjective search. *Evolutionary Computation*, 23(3):369–395, Sept 2015.

[43] N. Celik, G. Pusat, and E. Turgut. Application of taguchi method and grey relational analysis on a turbulated heat exchanger. *INTERNATIONAL JOURNAL OF THERMAL SCIENCES*, 124:85–97, FEB 2018.

[44] W. Chen, S. Liu, and K. Tang. A new knowledge gradient-based method for constrained bayesian optimization. *arXiv preprint arXiv:2101.08743*, 2021.

[45] A. Chergui, K. Hadj-Hamou, and F. Vignat. Production scheduling and nesting in additive manufacturing. *Computers and Industrial Engineering*, 126:292–301, 2018.

[46] C. Chevalier, J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4):455–465, 2014.

[47] H.Y. Chia, J. Wu, X. Wang, and W. Yan. Process parameter optimization of metal additive manufacturing: a review and outlook. *Journal of Materials Informatics*, 2:16, 10 2022.

[48] C. Cristescu and J. Knowles. Surrogate-based multiobjective optimization: Parego update and test. In *Workshop on Computational Intelligence (UKCI)*, volume 770, 2015.

[49] A. Cully, J. Clune, D. Tarapore, and J.B. Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[50] S. Daulton, M. Balandat, and E. Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. *arXiv preprint arXiv:2006.05078*, 2020.

[51] S. Daulton, M. Balandat, and E. Bakshy. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *arXiv preprint arXiv:2105.08195*, 2021.

[52] A. Dave, J. Mitchell, K. Kandasamy, H. Wang, S. Burke, B. Paria, B. Póczos, J. Whitacre, and V. Viswanathan. Autonomous discovery of battery electrolytes with robotic experimentation and machine learning. *Cell Reports Physical Science*, 1(12):100264, 2020.

[53] A. De Palma, C. Mendler-Dünner, T. Parnell, A. Anghel, and H. Pozidis. Sampling acquisition functions for batch bayesian optimization. *arXiv preprint arXiv:1903.09434*, 2019.

[54] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley Interscience Series in S. Wiley, 2001.

[55] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, Apr 2002.

[56] R. Delanghe, T.V. Steenkiste, I. Couckuyt, D. Deschrijver, and T. Dhaene. A bayesian optimisation procedure for estimating optimal trajectories in electromagnetic compliance testing. *Engineering Proceedings*, 3(1):8, 2020.

[57] J. Deneault, J. Chang, J. Myung, D. Hooper, A. Armstrong, M. Pitt, and B. Maruyama. Toward autonomous additive manufacturing: Bayesian optimization on a 3d printer. *MRS Bulletin*, 46, 04 2021.

[58] B. Derbel, D. Brockhoff, A. Liefooghe, and S. Verel. On the Impact of Scalarizing Functions on Evolutionary Multiobjective Optimization. Research Report RR-8512, March 2014.

[59] T. Desautels, A. Krause, and J.W. Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research*, 15(119):4053–4103, 2014.

[60] U.M. Dilberoglu, B. Gharehpapagh, U. Yaman, and M. Dolen. The role of additive manufacturing in the era of industry 4.0. *Procedia Manufacturing*, 11:545–554, 2017. 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy.

[61] B. Duraković. Design of experiments application, concepts, examples: State of the art. *Periodicals of Engineering and Natural Sciences (PEN)*, 5, 2017.

[62] L.F.C.S. Durão, R. Barkoczy, E. Zancul, L. Lee Ho, and R. Bonnard. Optimizing additive manufacturing parameters for the fused deposition modeling technology using a design of experiments. *Progress in Additive Manufacturing*, 4(3):291–313, 2019.

[63] D. Duvenaud. Automatic model construction with gaussian processes. 2014.

[64] M. Emmerich. The computation of the expected improvement in dominated hypervolume of pareto front approximations. 01 2008.

[65] M.T.M. Emmerich, A.H. Deutz, and J.W. Klinkenberg. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2147–2154, 2011.

[66] A. Esmaeili and M. Gholami. Optimization and preparation of nanocapsules for food applications using two methodologies. *FOOD CHEMISTRY*, 179:26–34, JUL 15 2015.

[67] F.R. Espinoza-Quinones, M.M.T. Fornari, A.N. Modenes, S.M. Palacio, F.G. da Silva, Jr., N. Szymanski, A.D. Kroumov, and D.E.G. Trigueros. Pollutant removal from tannery effluent by electrocoagulation. *CHEMICAL ENGINEERING JOURNAL*, 151(1-3):59–65, AUG 15 2009.

[68] D. Fernández-Sánchez, E.C. Garrido-Merchán, and D. Hernández-Lobato. Improved max-value entropy search for multi-objective bayesian optimization with constraints. *arXiv preprint arXiv:2011.01150*, 2020.

[69] C. Finnsgard and C. Wanstrom. Factors impacting manual picking on assembly lines: an experiment in the automotive industry. *INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 51(6):1789–1798, MAR 15 2013.

[70] R. Fisher. *The Design of Experiments*. The Design of Experiments. Oliver and Boyd, 1935.

[71] C.M. Fonseca and P.J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 28(1):26–37, Jan 1998.

[72] C.M. Fonseca and P.J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. ii. application example. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 28(1):38–47, Jan 1998.

[73] C.M. Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423, San Francisco, CA, USA, 1993.

[74] A.I. Forrester and A.J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79, 2009.

[75] M.R. Galankashi, E. Fallahiarezoudar, A. Moazzami, N.M. Yusof, and S.A. Helmi. Performance evaluation of a petrol station queuing system: A simulation-based design of experiments study. *Advances in Engineering Software*, 92:15–26, 2016.

[76] J.R. Gardner, M.J. Kusner, Z.E. Xu, K.Q. Weinberger, and J.P. Cunningham. Bayesian optimization with inequality constraints. In *ICML*, volume 2014, pages 937–945, 2014.

[77] R. Garnett, M.A. Osborne, and S.J. Roberts. Bayesian optimization for sensor set selection. In *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*, pages 209–219, 2010.

[78] E.C. Garrido-Merchán and D. Hernández-Lobato. Predictive entropy search for multi-objective bayesian optimization with constraints. *Neurocomputing*, 361:50–68, 2019.

[79] S.S. Garud, I.A. Karimi, and M. Kraft. Design of computer experiments: A review. *Computers & Chemical Engineering*, 106:71–95, 2017. ESCAPE-26.

[80] M.A. Gelbart, J. Snoek, and R.P. Adams. Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*, 2014.

[81] I. Giagkiozis and P. Fleming. Methods for multi-objective optimization: An analysis. *Information Sciences*, 293:338 – 350, 2015.

[82] I. Giagkiozis, R.C. Purshouse, and P.J. Fleming. Towards understanding the cost of adaptation in decomposition-based optimization algorithms.

In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 615–620, Oct 2013.

[83] I. Giagkiozis, R. Purshouse, and P. Fleming. Generalized decomposition and cross entropy methods for many-objective optimization. *Information Sciences*, 282:363 – 387, 2014.

[84] D. Ginsbourger, J. Janusevskis, and R.L. Riche. Dealing with asynchronicity in parallel gaussian process based global optimization. 2010.

[85] D. Ginsbourger, R. Le Riche, and L. Carraro. A multi-points criterion for deterministic parallel global optimization based on kriging. 03 2008.

[86] D. Ginsbourger, R. Le Riche, and L. Carraro. *Kriging Is Well-Suited to Parallelize Optimization*, pages 131–162. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[87] T.N. Goh. A strategic assessment of six sigma. *Quality and Reliability Engineering International*, 18(5):403–410, 2002.

[88] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition, 1989.

[89] D.E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In G.J. RAWLINS, editor, *Foundations of Genetic Algorithms*, volume 1, pages 69–93. Elsevier, 1991.

[90] A.E. Gongora, B. Xu, W. Perry, C. Okoye, P. Riley, K.G. Reyes, E.F. Morgan, and K.A. Brown. A bayesian experimental autonomous researcher for mechanical design. *Science advances*, 6(15):eaaz1708, 2020.

[91] J. Gonzalez, Z. Dai, P. Hennig, and N. Lawrence. Batch bayesian optimization via local penalization. In A. Gretton and C.C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 648–657, Cadiz, Spain, 09–11 May 2016.

[92] S. Greenhill, S. Rana, S. Gupta, P. Vellanki, and S. Venkatesh. Bayesian optimization for adaptive experimental design: A review. *IEEE Access*, 8:13937–13948, 2020.

[93] R.R. Griffiths and J.M. Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chem. Sci.*, 11:577–586, 2020.

[94] H. Guo and A. Mettas. Design of experiments and data analysis. In *Annual Reliability and Maintainability Symposium*, 2012.

[95] G. Gutin, A. Yeo, and A. Zverovich. Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the tsp. *Discrete Applied Mathematics*, 117(1):81–86, 2002.

[96] J. Hakanen and J.D. Knowles. On using decision maker preferences with parego. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 282–297. Springer, 2017.

[97] J.H. Halton. Algorithm 247: Radical-inverse quasi-random point sequence. *Commun. ACM*, 7(12):701–702, dec 1964.

[98] J.M. Hammersley. Monte carlo methods for solving multivariable problems. *Annals of the New York Academy of Sciences*, 86(3):844–874, 1960.

[99] J.K. Hartman. Some experiments in global optimization. *Naval Research Logistics Quarterly*, 20(3):569–576, 1973.

[100] P. Hennig and C.J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6), 2012.

[101] J.M. Henrández-Lobato, M.W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, page 918–926, Cambridge, MA, USA, 2014.

[102] D. Hernández-Lobato, J.M. Hernández-Lobato, A. Shah, and R.P. Adams. Predictive entropy search for multi-objective bayesian optimization. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1492–1501. 2016.

[103] J.M. Hernández-Lobato, M. Gelbart, M. Hoffman, R. Adams, and Z. Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In *International conference on machine learning*, pages 1699–1707. PMLR, 2015.

[104] N. Hertlein, K. Vemaganti, and S. Anand. Bayesian optimization of energy-absorbing lattice structures for additive manufacturing. In *ASME International Mechanical Engineering Congress and Exposition*, volume 84539, page V006T06A028. American Society of Mechanical Engineers, 2020.

[105] D. Horváth, R. Noorani, and M.I. Mendelson. Improvement of surface roughness on abs 400 polymer using design of experiments (doe). *Materials Science Forum*, 561-565:2389 – 2392, 2007.

[106] E. Hughes. Multiple single objective pareto sampling. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, volume 4, pages 2678–2684 Vol.4, 2003.

[107] E. Hughes. Evolutionary many-objective optimisation: many once or one many? In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 222–227 Vol.1, 2005.

[108] F. Hutter, H.H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International conference on learning and intelligent optimization*, pages 507–523. Springer, 2011.

[109] L. Ilzarbe, M.J. Álvarez, E. Viles, and M. Tanco. Practical applications of design of experiments in the field of engineering: a bibliographical review. *Quality and Reliability Engineering International*, 24(4):417–428, 2008.

[110] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 2419–2426, June 2008.

[111] M. Jamil and X.S. Yang. A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194, 2013. PMID: 55204.

[112] A. Jaszkiewicz. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment. *IEEE Transactions on Evolutionary Computation*, 6(4):402–412, 2002.

[113] S. Jeong and S. Obayashi. Efficient global optimization (ego) for multi-objective problem and data mining. In *2005 IEEE Congress on Evolutionary Computation*, volume 3, pages 2138–2145 Vol. 3, 2005.

[114] S. Jiang, S. Yang, Y. Wang, and X. Liu. Scalarizing functions in decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 22(2):296–313, 2017.

[115] M. Johnson, L. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26(2):131–148, 1990.

[116] D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[117] D. Jones, S. Mirrazavi, and M. Tamiz. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1):1 – 9, 2002.

[118] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

[119] D.R. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[120] V.R. Joseph. Space-filling designs for computer experiments: A review. *Quality Engineering*, 28(1):28–35, 2016.

[121] K. Kandasamy, J. Schneider, and B. Poczos. High dimensional bayesian optimisation and bandits via additive models. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 295–304, Lille, France, 07–09 Jul 2015.

[122] D.S. Karna and R. Sahai. An overview on taguchi method. *International Journal of Engineering and Mathematical Sciences*, 1:1–7, 01 2012.

[123] R.S. Kenett and G. Vicario. Challenges and opportunities in simulations and computer experiments in industrial statistics: An industry 4.0 perspective. *Advanced Theory and Simulations*, 4(2):2000254, 2021.

[124] M. Khosravi, C. König, M. Maier, R.S. Smith, J. Lygeros, and A. Rupenyan. Safety-aware cascade controller tuning using constrained

bayesian optimization. *IEEE Transactions on Industrial Electronics*, 70: 2128–2138, 2020.

[125] A.I. Khuri and S. Mukhopadhyay. Response surface methodology. *WIREs Computational Statistics*, 2(2):128–149, 2010.

[126] J. Knowles and D. Corne. Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, April 2003.

[127] J. Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

[128] D. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand, by d.g. krige, published in the journal, december 1951 : introduction by the author. *Journal of The South African Institute of Mining and Metallurgy*, 52:201–203, 1951.

[129] A. Kumar, K.K. Pant, S. Upadhyayula, and H. Kodamana. Multiobjective bayesian optimization framework for the synthesis of methanol from syngas using interpretable gaussian process models. *ACS omega*, 2022.

[130] R. Kumar, M. Kumar, and J.S. Chohan. The role of additive manufacturing for biomedical applications: A critical review. *Journal of Manufacturing Processes*, 64:828–850, 2021.

[131] H.J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. In *Joint Automatic Control Conference*, number 1, pages 69–79, 1963.

[132] J. Laeng, A.Z. Khan, and . S.Y.Khu. Optimizing flexible behaviour of bow prototype using taguchi approach. *Journal of Applied Sciences*, 6: 622–630, 2006.

[133] T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[134] R. Lam and K. Willcox. Lookahead bayesian optimization with inequality constraints. *Advances in Neural Information Processing Systems*, 30, 2017.

[135] J. Lancaster, R. Lorenz, R. Leech, and J.H. Cole. Bayesian optimization for neuroimaging pre-processing in brain age classification and prediction. *Frontiers in aging neuroscience*, 10:28, 2018.

[136] A. Lara, S. Alvarado, S. Salomon, G. Avigad, C.A. Coello Coello, and O. Schütze. The gradient free directed search method as local search within multi-objective evolutionary algorithms. In O. Schütze, C.A. Coello Coello, A.A. Tantar, E. Tantar, P. Bouvry, P. Del Moral, and P. Legrand, editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, pages 153–168, Berlin, Heidelberg, 2013.

[137] A. Larroque, M. Fosas de Pando, and L. Lafuente. Cylinder drag minimization through wall actuation: A bayesian optimization approach. *Computers & Fluids*, 240:105370, 2022.

[138] M. Laumanns, E. Zitzler, and L. Thiele. A unified model for multi-objective evolutionary algorithms with elitism. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, volume 1, pages 46–53 vol.1, July 2000.

[139] G. Li, M. Li, S. Azarm, S.A. Hashimi, T.A.A. Ameri, and N.A. Qasas. Improving multi-objective genetic algorithms with adaptive design of experiments and online metamodeling. *Structural and Multidisciplinary Optimization*, 37:447–461, 2009.

[140] M. Li, G. Li, and S. Azarm. A kriging metamodel assisted multi-objective genetic algorithm for design optimization. 2008.

[141] Y. Li, Y. Huang, Z. Wei, L. Feng, and Y. Zhang. A case study of application of modern design of experiment methods in high speed wind tunnel test. In G. Ran, Z. Yun, Z. Jianming, Y. Yang, L. Ze, and G. Tao, editors, *ADVANCES IN COMPUTATIONAL MODELING AND SIMULATION, PTS 1 AND 2*, volume 444-445 of *Applied Mechanics and Materials*, pages 1229–1233. Kunming Univ Sci & Technol; Yunnan Soc Theoret & Appl Mech, 2014. 2nd International Conference on Advances in Computational Modeling and Simulation (ACMS 2013), Kunming, PEOPLES R CHINA, JUL 17-19, 2013.

[142] S. Lim, A. Garbo, P. Bekemeyer, C. Appel, R. Ewert, and J. Delfs. *High-fidelity Aerodynamic and Aeroacoustic Multi-Objective Bayesian Optimization.*

[143] H. Liu, Y. Ong, X. Shen, and J. Cai. When gaussian process meets big data: A review of scalable gps. *IEEE Transactions on Neural Networks and Learning Systems*, 31:4405–4423, 2020.

[144] D.J. Lizotte, T. Wang, M.H. Bowling, D. Schuurmans, et al. Automatic gait optimization with gaussian process regression. In *IJCAI*, volume 7, pages 944–949, 2007.

[145] D.J. Lizotte, R. Greiner, and D. Schuurmans. An experimental methodology for response surface optimization methods. *Journal of Global Optimization*, 53(4):699–736, 2012.

[146] D.J. Lizotte. *Practical Bayesian Optimization*. PhD thesis, CAN, 2008. AAINR46365.

[147] R. Lorenz, R.P. Monti, I. Violante, A. Faisal, C. Anagnostopoulos, R. Leech, and G. Montana. Stopping criteria for boosting automatic experimental design using real-time fmri with bayesian optimization. volume 1511, 11 2015.

[148] A.L. Lupatini, L.d.O. Bispo, L.M. Colla, J.A. Vieira Costa, C. Canan, and E. Colla. Protein and carbohydrate extraction from s-platensis biomass by ultrasound and mechanical agitation. *FOOD RESEARCH INTERNATIONAL*, 99(3, SI):1028–1035, SEP 2017.

[149] M.J. Álvarez, L. Ilzarbe, E. Viles, and M. Tanco. The use of genetic algorithms in response surface methodology. *Quality Technology & Quantitative Management*, 6(3):295–307, 2009.

[150] X. Ma, F. Liu, Y. Qi, L. Li, L. Jiao, X. Deng, X. Wang, B. Dong, Z. Hou, Y. Zhang, et al. Moea/d with biased weight adjustment inspired by user preference and its application on multi-objective reservoir flood control problem. *Soft Computing*, 20:4999–5023, 2016.

[151] P.C. Mahalanobis. A sample survey of the acreage under jute in bengal. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 4(4):511–530, 1940.

[152] A. Makarova, H. Shen, V. Perrone, A. Klein, J.B. Faddoul, A. Krause, M. Seeger, and C. Archambeau. Automatic termination for hyperparameter optimization. 2021.

[153] R. Marchant and F. Ramos. Bayesian optimisation for intelligent environmental monitoring. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2242–2249, 2012.

[154] G. Matheron. *The theory of regionalized variables and its applications*. Les cahiers du Centre de Morphologie Mathématique de Fontainebleau. École Nationale Supérieure des Mines, 1971.

[155] M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[156] M.C. Menkiti and M. Ejimofor, I. Experimental and artificial neural network application on the optimization of paint effluent (pe) coagulation using novel achatinoidea shell extract (ase). *JOURNAL OF WATER PROCESS ENGINEERING*, 10:172–187, APR 2016.

[157] E. Merrill, A. Fern, X. Fern, and N. Dolatnia. An empirical study of bayesian optimization: Acquisition versus partition. *Journal of Machine Learning Research*, 22(4):1–25, 2021.

[158] N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949. PMID: 18139350.

[159] Z. Michalewicz. Genetic algorithms + data structures = evolution programs. In *Artificial Intelligence*, 1992.

[160] J. Mockus. Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4): 347–365, Jun 1994.

[161] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

[162] O. Mohamed, S. Masood, and J. Bhowmik. Optimization of fused deposition modeling process parameters: a review of current research and future prospects. *Advances in Manufacturing*, 3:42–53, 2015.

[163] S. Mondal, D. Gwynn, A. Ray, and A. Basak. Investigation of melt pool geometry control in additive manufacturing using hybrid modeling. *Metals*, 10(5):683, 2020.

[164] D. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, Incorporated, 2017.

[165] R. Moriconi, M. Deisenroth, and S.K. K S. High-dimensional bayesian optimization using low-dimensional feature spaces. *Machine Learning*, 109:1925–1943, 09 2020.

[166] M.D. Morris and T.J. Mitchell. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43(3):381–402, 1995.

[167] R. Myers, D. Montgomery, G. Vining, C. Borror, and S. Kowalski. Response surface methodology: A retrospective and literature survey. *Journal of Quality Technology*, 36:53–78, 01 2004.

[168] R. Myers, D. Montgomery, and C. Anderson-Cook. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley Series in Probability and Statistics. Wiley, 2016.

[169] T. Nancharaiah, D. Raju, and V. Raju. An experimental investigation on surface quality and dimensional accuracy of fdm components. *Int J Emerg Technol*, 1, 01 2010.

[170] T.D. Ngo, A. Kashani, G. Imbalzano, K.T. Nguyen, and D. Hui. Additive manufacturing (3d printing): A review of materials, methods, applications and challenges. *Composites Part B: Engineering*, 143:172–196, 2018.

[171] V. Nguyen, S. Gupta, S. Rana, C. Li, and S. Venkatesh. Regret for expected improvement over the best-observed value and stopping condition. In M.L. Zhang and Y.K. Noh, editors, *Proceedings of the Ninth Asian Conference on Machine Learning*, volume 77 of *Proceedings of Machine Learning Research*, pages 279–294, Yonsei University, Seoul, Republic of Korea, 15–17 Nov 2017.

[172] H. Ohno and A. Suzumura. Electrolyte recommender system for batteries using ensemble bayesian optimization. *IFAC Journal of Systems and Control*, 16:100158, 2021.

[173] B. Paria, K. Kandasamy, and B. Póczos. A flexible framework for multi-objective bayesian optimization using random scalarizations. In R.P. Adams and V. Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 766–776. 22–25 Jul 2020.

[174] V. Perrone, I. Shcherbatyi, R. Jenatton, C. Archambeau, and M. Seeger. Constrained bayesian optimization with max-value entropy search. *arXiv preprint arXiv:1910.07003*, 2019.

[175] V. Picheny. A stepwise uncertainty reduction approach to constrained global optimization. In *Artificial intelligence and statistics*, pages 787–795. PMLR, 2014.

[176] V. Picheny. Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25(6):1265–1280, 2015.

[177] J. Pietraszek, N. Radek, and A.V. Goroshko. Challenges for the doe methodology related to the introduction of industry 4.0. *Production Engineering Archives*, 26(4):190–194, 2020.

[178] M. Pilat and R. Neruda. Incorporating user preferences in moea/d through the coevolution of weights. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 727–734, 2015.

[179] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$-metric selection. In G. Rudolph, T. Jansen, N. Beume, S. Lucas, and C. Poloni, editors, *Parallel Problem Solving from Nature – PPSN X*, pages 784–794, Berlin, Heidelberg, 2008.

[180] R. Priem, N. Bartoli, Y. Diouane, and A. Sgueglia. Upper trust bound feasibility criterion for mixed constrained bayesian optimization with application to aircraft design. *Aerospace Science and Technology*, 105:105980, 2020.

[181] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu. Moea/d with adaptive weight adjustment. *Evolutionary Computation*, 22(2):231–264, 2014.

[182] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. Adaptative computation and machine learning series. University Press Group Limited, 2006.

[183] N. Riquelme, C. Von Lücken, and B. Baran. Performance metrics in multi-objective optimization. In *2015 Latin American Computing Conference (CLEI)*, pages 1–11, 2015.

[184] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, pages 527–535, 1952.

[185] S. Rojas-Gonzalez and I. Van Nieuwenhuyse. A survey on kriging-based infill algorithms for multiobjective simulation optimization. *Computers & Operations Research*, 116:104869, 2020.

[186] A. Rollett, G. Rohrer, and R. Suter. Understanding materials microstructure and behavior at the mesoscale. *MRS Bulletin*, 40(11):951–960, 2015.

[187] K. Ruberu, M. Senadeera, S. Rana, S. Gupta, J. Chung, Z. Yue, S. Venkatesh, and G. Wallace. Coupling machine learning with 3d bioprinting to fast track optimisation of extrusion printing. *Applied Materials Today*, 22:100914, 2021.

[188] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989.

[189] M.S. Salwani, A. Ali, B.B. Sahari, A.A. Nuraini, A.A. Faieza, T.H.T. Ismail, J.M. Nursherida, A. Azizi, N.M. Safar, S.A. Satuan, S. Mohamad, and N. Muhamed. Analysis on impact performance of aluminum automotive side member. In S. Abdullah, R. Zulkifli, S. Haris, M. Nuawi, Z. Nopiah, A. Arifin, and W. Mahmood, editors, *TRENDS IN AUTOMOTIVE RESEARCH*, volume 165 of *Applied Mechanics and Materials*, pages 209+. Univ Kebangsaan Malaysia, Ctr Automot Res, Fac Engn & Built Environm, 2012. Regional Conference on Automotive Research (ReCAR), Kuala Lumpur, MALAYSIA, DEC 14-15, 2011.

[190] Y.C. San and M.D. Mashitah. Optimization of process parameters in bioreduction of silver nanoparticies by pycnoporus sanguineus. In M. Abdullah, L. Jamaludin, R. Razak, Z. Yahya, and K. Hussin, editors, *ADVANCED MATERIALS ENGINEERING AND TECHNOLOGY*, volume 626 of *Advanced Materials Research*, pages 95–98. Ctr Excellence Geopolymer & Green Technol; Univ Malaysia Perlis, Sch Mat Engn; King Abdul Aziz City Sci & Technol; Univ Cambridge; SMaRT@UNSW, Univ New S Wales; Norton Univ; Univ Malaysia Sarawak; Univ Putra Malaysia, Inst Adv Technol, 2012. International Conference on Advanced Materials Engineering and Technology (ICAMET 2012), Penang, MALAYSIA, NOV 28-30, 2012.

[191] O. Schutze, X. Esquivel, A. Lara, and C.A.C. Coello. Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16 (4):504–522, Aug 2012.

[192] A. Shah and Z. Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. 2015.

[193] A. Shah, A. Wilson, and Z. Ghahramani. Student-t Processes as Alternatives to Gaussian Processes. In S. Kaski and J. Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 877–885, Reykjavik, Iceland, 22–25 Apr 2014.

[194] A. Shahin, N. Janatyan, and N. Nasirzaheh. Service quality robust design – with a case study in airport services. *International Journal of Productivity and Quality Management*, 9(3):404–421, 2012. PMID: 46369.

[195] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, Jan 2016.

[196] R. Shalloo, S. Dann, J.N. Gruse, C. Underwood, A. Antoine, C. Arran, M. Backhouse, C. Baird, M. Balcazar, N. Bourgeois, et al. Automation and control of laser wakefield accelerators using bayesian optimization. *Nature communications*, 11(1):6355, 2020.

[197] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[198] C. Sharpe, C.C. Seepersad, S. Watts, and D. Tortorelli. Design of mechanical metamaterials via constrained bayesian optimization. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 51753, page V02AT03A029. American Society of Mechanical Engineers, 2018.

[199] B.J. Shields, J. Stevens, J. Li, M. Parasram, F. Damani, J.I.M. Alvarado, J.M. Janey, R.P. Adams, and A.G. Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, 2021.

[200] J. Simao, H. Lee, D. Aspinwall, R. Dewes, and E. Aspinwall. Workpiece surface modification using electrical discharge machining. *INTERNATIONAL JOURNAL OF MACHINE TOOLS & MANUFACTURE*, 43(2): 121–128, JAN 2003.

[201] V. Singh, R. Bhandari, and V.K. Yadav. An experimental investigation on machining parameters of aisi d2 steel using wedm. *INTERNATIONAL JOURNAL OF ADVANCED MANUFACTURING TECHNOLOGY*, 93(1-4):203–214, OCT 2017.

[202] J. Siwei, C. Zhihua, Z. Jie, and O. Yew-Soon. Multiobjective optimization by decomposition with pareto-adaptive weight vectors. In *2011 Seventh International Conference on Natural Computation*, volume 3, pages 1260–1264, 2011.

[203] J. Snoek, H. Larochelle, and R.P. Adams. Practical bayesian optimization of machine learning algorithms. In *NIPS*, 2012.

[204] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M.M.A. Patwary, P. Prabhat, and R.P. Adams. Scalable bayesian optimization using deep neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2171–2180, Lille, France, 2015.

[205] I. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967.

[206] J.T. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust bayesian neural networks. *Advances in neural information processing systems*, 29:4134–4142, 2016.

[207] N. Srinivas and K. Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, Sept 1994.

[208] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 1015–1022, Madison, WI, USA, 2010.

[209] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved January 28, 2023, from `http://www.sfu.ca/~ssurjano`.

[210] K. Swersky, J. Snoek, and R.P. Adams. Multi-task bayesian optimization. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger,

editors, *Advances in Neural Information Processing Systems*, volume 26. 2013.

[211] G. Taguchi. *System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs*. Number v. 2 in System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs. UNIPUB/Kraus International Publications, 1987.

[212] S. Takeno, H. Fukuoka, Y. Tsukada, T. Koyama, M. Shiga, I. Takeuchi, and M. Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In H.D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9334–9345. 13–18 Jul 2020.

[213] M. Tanco, E. Viles, L. Ilzarbe, and M. Alvarez. Manufacturing industries need design of experiments (doe). *Lecture Notes in Engineering and Computer Science*, 2, 07 2007.

[214] W.R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25 (3/4):285–294, 1933.

[215] W.R. Thompson. On the theory of apportionment. *American Journal of Mathematics*, 57(2):450–456, 1935.

[216] R. Trangucci, M. Betancourt, and A. Vehtari. Prior formulation for gaussian process hyperparameters. In *Practical Bayesian Nonparametrics workshop, NIPS*, 2016.

[217] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3):440–462, 2017.

[218] T. Ueno, T.D. Rhone, Z. Hou, T. Mizoguchi, and K. Tsuda. Combo: An efficient bayesian optimization library for materials science. *Materials discovery*, 4:18–21, 2016.

[219] J. Ungredda and J. Branke. Bayesian optimisation for constrained problems. *arXiv preprint arXiv:2105.13245*, 2021.

[220] A. Uriondo, M. Esperon-Miguez, and S. Perinpanayagam. The present and future of additive manufacturing in the aerospace sector: A review

of important aspects. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 229(11):2132–2147, 2015.

[221] A. Vafadar, F. Guzzomi, A. Rassau, and K. Hayward. Advances in metal additive manufacturing: A review of common processes, industrial applications, and current challenges. *Applied Sciences*, 11(3), 2021.

[222] S. Vaidya, P. Ambad, and S. Bhosle. Industry 4.0 – a glimpse. *Procedia Manufacturing*, 20:233–238, 2018. 2nd International Conference on Materials, Manufacturing and Design Engineering (iCMMD2017), 11-12 December 2017, MIT Aurangabad, Maharashtra, INDIA.

[223] C. van der Westhuizen, J. Du Toit, N. Neyt, D. Riley, and J.L. Panayides. Use of open-source software platform to develop dashboards for control and automation of flow chemistry equipment. *Digital Discovery*, 1, 01 2022.

[224] V.B. Veljkovic, A.V. Velickovic, J.M. Avramovic, and O.S. Stamenkovic. Modeling of biodiesel production: Performance comparison of box-behnken, face central composite and full factorial design. *CHINESE JOURNAL OF CHEMICAL ENGINEERING*, 27(7):1690–1698, JUL 2019.

[225] T. Wada and H. Hino. Bayesian optimization for multi-objective optimization and multi-point search. *arXiv preprint arXiv:1905.02370*, 2019.

[226] C. Wang, X. Tan, S. Tor, and C. Lim. Machine learning in additive manufacturing: State-of-the-art and perspectives. *Additive Manufacturing*, 36:101538, 2020.

[227] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao. Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 20(3):475–480, June 2016.

[228] R. Wang, Q. Zhang, and T. Zhang. Decomposition-based algorithms using pareto adaptive scalarizing methods. *IEEE Transactions on Evolutionary Computation*, 20(6):821–837, 2016.

[229] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer. Recent advances in bayesian optimization. *arXiv preprint arXiv:2206.03301*, 2022.

[230] Z. Wang and S. Jegelka. Max-value entropy search for efficient bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3627–3635, Sydney, NSW, Australia, 2017.

[231] Z. Wang, C. Gehring, P. Kohli, and S. Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754. PMLR, 2018.

[232] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *Trans. Evol. Comp*, 1(1):67–82, April 1997.

[233] K.C. Wong, C.H. Wu, R.K. Mok, C. Peng, and Z. Zhang. Evolutionary multimodal optimization using the principle of locality. *Information Sciences*, 194:138–170, 2012. Intelligent Knowledge-Based Models and Methodologies for Complex Information Systems.

[234] C. Wu and M. Hamada. *Experiments: Planning, Analysis, and Optimization*. Wiley Series in Probability and Statistics. Wiley, 2011.

[235] P. Wu, J. Wang, and X. Wang. A critical review of the use of 3-d printing in the construction industry. *Automation in Construction*, 68:21–31, 2016.

[236] Y. Xiong, P.L.T. Duong, D. Wang, S.I. Park, Q. Ge, N. Raghavan, and D.W. Rosen. Data-driven design space exploration and exploitation for design for additive manufacturing. *Journal of Mechanical Design*, 141(10), 2019.

[237] Q. Xu, Z. Xu, and T. Ma. A survey of multiobjective evolutionary algorithms based on decomposition: Variants, challenges and future directions. *IEEE Access*, 8:41588–41614, 2020.

[238] T. Xue, T.J. Wallin, Y. Menguc, S. Adriaenssens, and M. Chiaramonte. Machine learning generative models for automatic design of multi-material 3d printed composite solids. *Extreme Mechanics Letters*, 41: 100992, 2020.

[239] K. Yang, P.S. Palar, M. Emmerich, K. Shimoyama, and T. Bäck. A multi-point mechanism of expected hypervolume improvement for parallel multi-objective bayesian global optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, page 656–663, New York, NY, USA, 2019.

[240] K.Q. Ye. Orthogonal column latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association*, 93 (444):1430–1439, 1998.

[241] M. Yolmeh and S.M. Jafari. Applications of response surface methodology in the food industry processes. *Food and Bioprocess Technology*, 10 (3):413–433, MAR 2017.

[242] G.R. Zavala, A.J. Nebro, F. Luna, and C.A. Coello Coello. A survey of multi-objective metaheuristics applied to structural optimization. *Structural and Multidisciplinary Optimization*, 49(4):537–558, Apr 2014.

[243] D. Zhan and H. Xing. Expected improvement for expensive optimization: a review. *Journal of Global Optimization*, 78(3):507–544, 2020.

[244] J. Zhang and L. Xing. A survey of multiobjective evolutionary algorithms. In *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, volume 1, pages 93–100, July 2017.

[245] J.W. Zhang and A.H. Peng. Process-parameter optimization for fused deposition modeling based on taguchi method. In *Materials Processing Technology II*, volume 538 of *Advanced Materials Research*, pages 444–447. 9 2012.

[246] M. Zhang, A. Parnell, D. Brabazon, and A. Benavoli. Bayesian optimisation for sequential experimental design with applications in additive manufacturing. *arXiv preprint arXiv:2107.12809*, 2021.

[247] Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, Dec 2007.

[248] Q. Zhang, W. Liu, E. Tsang, and B. Virginas. Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474, 2009.

[249] S. Zhang, F. Yang, C. Yan, D. Zhou, and X. Zeng. An efficient batch-constrained bayesian optimization approach for analog circuit synthesis via multiobjective acquisition ensemble. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(1):1–14, 2022.

[250] Y. Zhang, R. Yang, J. Zuo, and X. Jing. Enhancing moea/d with uniform population initialization, weight vector design and adjustment using uniform design. *Journal of Systems Engineering and Electronics*, 26(5): 1010–1022, 2015.

[251] Y. Zhang, X. Zhang, and P. Frazier. Constrained two-step look-ahead bayesian optimization. *Advances in Neural Information Processing Systems*, 34:12563–12575, 2021.

[252] J. Zhou, D. Wu, and D. Guo. Optimization of the production of thiocarbohydrazide using the taguchi method. *JOURNAL OF CHEMICAL TECHNOLOGY AND BIOTECHNOLOGY*, 85(10):1402–1406, OCT 2010.

[253] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, Nov 1999.

[254] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In X. Yao, E.K. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J.E. Rowe, P. Tiňo, A. Kabán, and H.P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, pages 832–842, Berlin, Heidelberg, 2004.

[255] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, 2002.

[256] M. Zuluaga, G. Sergent, A. Krause, and M. Püschel. Active learning for multi-objective optimization. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 462–470, Atlanta, Georgia, USA, 17–19 Jun 2013.

[257] M. Zuluaga, A. Krause, and M. Püschel. ε-pal: an active learning approach to the multi-objective optimization problem. *The Journal of Machine Learning Research*, 17(1):3619–3650, 2016.