An exploration of building design and optimisation methods using Kriging meta-modelling

Submitted by

Michael James Wood

to the

University of Exeter

as a thesis for the degree of

Doctor of Philosophy

in

Engineering

February 2016

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

Signed …….

Abstract

This thesis investigates the application of Kriging meta-modelling techniques in the field of building design and optimisation.

In conducting this research, there were two key motivational factors. The first is the need for building designers to have tools that allow low energy buildings to be designed in a fast and efficient manner. The second motivating factor is the need for optimisation tools that account, or help account, for the wide variety of uses that a building might have; so-called Robust Optimisation (RO).

This thesis therefore includes an analysis of Kriging meta-modelling and first applies this to simple building problems. I then use this simple building model to determine the effect of the updated UK Test Reference Years (TRYs) on energy consumption. Second, I examine Kriging-based optimisation techniques for a single objective. I then revisit the single-building meta-model to examine the effect of uncertainty on a neighbourhood of buildings and compare the results to the output of a brute-force analysis of a full building simulator. The results show that the Kriging emulation is an effective tool for creating a meta-model of a building. The subsequent use in the analysis of the effect of TRYs on building shows that UK buildings are likely to use less heating in the future but are likely to overheat more.

In the final two chapters I use the techniques developed to create a robust building optimisation algorithm as well as using Kriging to improve the optimisation efficiency of the well-known NSGA-II algorithm. I show that the Kriging-based robust optimiser effectively finds more robust solutions than

traditional global optimisation. I also show that Kriging techniques can be used to augment NSGA-II so that it finds more diverse solutions to some types of multi-objective optimisation problems. The results show that Kriging has significant potential in this field and I reveal many potential areas of future research.

This thesis shows how a Kriging-enhanced NSGA-II multi-objective optimisation algorithm can be used to improve the performance of NSGA-II. This new algorithm has been shown to speed up the convergence of some multi-objective optimisation algorithms significantly. Although further work is required to verify the results for a wider variety of building applications, the initial results are promising. An overview of the main research chapters is given in Figure 0-1.

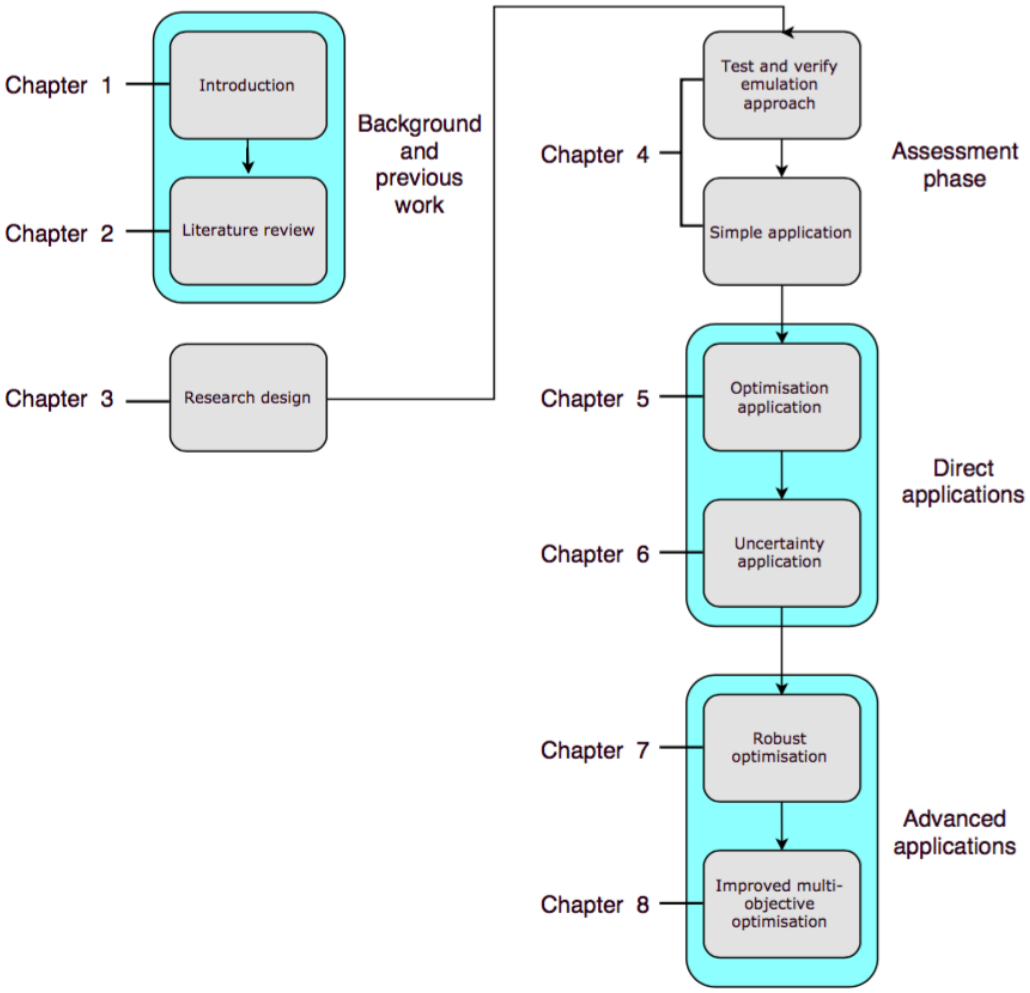


Figure 0-1: Phases of the research and their associated chapters.

Acknowledgements

The work in this thesis was supported by the Engineering and Physical Sciences Research Council [EPSRC grant number EP/J002380/1] and the Centre for Energy and the Environment at the University of Exeter.

I would like to thank my supervisors, Dr Matthew Eames and Professor Peter Challenor, for their encouragement and valuable technical advice throughout my research.

Publications

Wood, M. J., M. E. Eames, and P. G. Challenor. 2015. “A Comparison between Gaussian Process Emulation and Genetic Algorithms for Optimising Energy Use of Buildings.” In Proceedings of the Building Simulation Conference 2015.

Wood, M., Eames, M., & Challenor, P. (2014). Proof of concept for the Bayesian analysis of computer code output in building energy modelling. In *Building Simulation and Optimisation*.

Wood, Michael, and Matthew Eames. 2014. “A Probabilistic Approach to Estimating the Savings from Voltage Reduction.” In Proceedings of the Building Simulation and Optimisation Conference 2014.

Eames, M. E., A. P. Ramallo-Gonzalez, and **M. J. Wood.** 2015. “An Update of the UK’s Test Reference Year: The Implications of a Revised Climate on Building Design.” Building Services Engineering Research and Technology.

Symbols and Abbreviations

Symbols

Symbol	*Description*
n	Number of design points
p	Number of active inputs
q	Number of basis functions
r	Number of outputs
s	Number of hyperparameter sets in an emulator
$\mathbb{R}$	Real numbers

Symbol	*Description*
x	Point in the simulator's input space
y	Reality - the actual system value
D	Design, comprising an ordered set of points in an input space
$\mathrm{f}(.)$	The output(s) of a simulator
$\mathrm{h}(.)$	Vector of basis functions

Symbol	*Description*
β	Hyperparameters of a mean function
δ	Hyperparameters of a correlation function
σ^2	Scale hyperparameter for a covariance function
θ	Collection of hyperparameters on which the emulator is conditioned
π	Distribution of hyperparameters

Function	*Description*
$\mathrm{m}(.)$	Mean function
$\mathrm{v}(.,.)$	Covariance function
$\mathrm{m}^*(.)$	Posterior mean function
$\mathrm{v}^*(.,.)$	Posterior covariance function
$\mathrm{c}(.,.)$	Correlation function

Abbreviations

Term	Meaning
ACH	Air changes per hour
ANN	Artificial neural network
ASHRAE	American Society of Heating, Refrigeration, and Air Conditioning Engineers
BES	Building energy simulation
BIM	Building information modelling
CIBSE	Chartered institution of building services engineers
CREST	Centre for Renewable Energy Systems Technology
DICE	Deep inside computer experiments
DSY	Design summer year
EA	Evolutionary algorithm
EGO	Efficient global optimisation
EGOrobust	Robust efficient global optimization
EI	Expected improvement
EPFL	Ecole Polytechnique Federale de Lausanne
GA	Genetic algorithm
GP	Gaussian process
HVAC	Heating, ventilation and air conditioning
IES VE	Integrated Environment Solution Virtual Environment (building energy modelling software)
MC	Monte Carlo (sensitivity analysis technique)
MSE	Mean-squared error
NSGA-II	Non-dominated sorting genetic algorithm II
NSGA-II-KRS	Non-dominated sorting genetic algorithm II with Kriging Random Sampling
NSGA-II-KLHS	Non-dominated sorting genetic algorithm II with Kriging Latin Hypercube Sampling
RC	Resistor-capacitor (a method for modelling heat flow through building models)
SVM	Support vector machine
TRY	Test reference year
U-value	Measure of heat insulation
UA	Uncertainty analysis
UHI	Urban heat island
UKCP09	United Kingdom climate predictions 2009
ZDT	Zitzler-Deb-Thiele (developers of the ZDT functions)

Contents

Symbols and Abbreviations 7

Symbols 7

Abbreviations 8

Chapter 1 Introduction 24

1.1 Motivation 24

1.2 Global context 24

1.3 Human factors 25

1.4 Philosophy of building modelling 26

1.5 The challenge in building efficient building optimisation tools 27

1.6 Research Questions 30

1.7 Thesis Structure 32

Chapter 2 Literature Review 35

2.1 Introduction 35

2.1.1 Efficiency and effectiveness 36

2.2 Commercial software 37

2.2.1 Building information modelling and Cloud computing 38

2.3 Academic Research 39

2.3.1 Introduction 39

2.3.2 Engineering models 40

2.3.3 Statistical and meta-modelling techniques 42

2.3.4 Genetic Algorithms 44

2.3.5 Multi-objective optimisation using NSGA-II in buildings 47

2.3.6 Sources of uncertainty 50

2.4 Other sources of uncertainty 54

2.4.1 Overheating and future climate 55

2.4.2 Uncertainty analysis 57

2.4.3 Robust optimisation 61

2.5 Summary 62

Chapter 3 Research Design 64

3.1 Rationale for choice of experiments 65

Chapter 4 Emulating the building: An introduction to Kriging 67

4.1 Background 67

4.1.1 Basic concept and advantages 67

4.1.2 Mathematical view of Kriging 69

4.1.3 Gaussian Processes and their Bayes Linear Counterparts 69

4.1.4 Emulator / simulator terminology 70

4.1.5 Building an Emulator 71

4.1.6 Prior forms of the mean functions 72

4.1.7 Prior covariance function 74

4.1.8 Correlation functions 74

4.1.9 Posterior mean and covariance functions and the hyperparameters 75

4.2 Proof of concept for the creation of a Kriging emulator for a building design problem 78

4.2.1 The building simulator 79

4.2.2 About the building 80

4.2.3 Emulator variables 82

4.2.4 Selection of the training data 82

4.2.5 Validating the emulator 83

4.2.6 Individual standardised errors 83

4.2.7 Calculating the Mahalanobis distance 85

4.2.8 Brute force validation 86

4.2.9 Discussion of the results 88

4.3 Testing the effect of new test reference years using Kriging 90

4.3.1 Test references years in the UK 90

4.3.2 Choosing the ‘most average’ weather files 91

4.3.3 Implications for the new TRYs on buildings 92

4.3.4 Kriging meta-model 94
4.3.5 The building model and emulator 95
4.3.6 Results 97
4.4 Discussion 101
4.5 Summary 103
Chapter 5 Kriging optimisation methods 105
5.1 Introduction 105
5.2 Background 105
5.2.1 A simple strategy for improving the emulator 106
5.2.2 Kriging optimisation 107
5.2.3 Efficient Global Optimisation using Expected Improvement 108
5.3 Aims 110
5.4 Method 110
5.4.1 The GA optimisation settings and software 112
5.4.2 Kriging with Efficient Global Optimisation 113
5.4.3 Test setup 113
5.5 Results 115
5.6 Discussion 117
5.7 Summary 119

Chapter 6 Uncertainty management 120
6.1 Introduction 120
6.2 Aims 121
6.3 Method 121
6.3.1 Overview of the process 122
6.3.2 Assumed improvements to the buildings 124
6.3.3 Other variables 124
6.3.4 Uncertainty Bounds 126
6.3.5 The occupancy model 127
6.3.6 Emulator setup 128
6.3.7 Monte Carlo Analysis 129
6.4 Results 129
6.5 Discussion 135
6.6 Conclusions 137
6.7 Summary 137
Chapter 7 Robust Optimisation Methods 139
7.1 Introduction 139
7.2 Background 139
7.3 Aims 140

7.4 Method 141
7.4.1 Robust expected improvement using Kriging 142
7.4.2 Rehman windowing algorithm. 144
7.4.3 Adapted windowing algorithm 146
7.4.4 Experimental setup 147
7.4.5 Validation approach 148
7.4.6 The building model 149
7.5 Results 150
7.5.1 10 optimisation steps 150
7.5.2 20 optimisation steps 154
7.5.3 Building solutions 156
7.6 Discussion 158
7.7 Conclusions 160
7.8 Summary 162
Chapter 8 Multi-objective optimisation using Kriging 163
8.1 Introduction 163
8.2 Background 163
8.2.1 Non-domination and dominated zones 164
8.2.2 Diversity 165

8.2.3 Dominated hypervolume 165

8.2.4 NSGA-II 166

8.3 Aims 170

8.4 Method 171

8.4.1 Augmenting NSGA-II with Kriging 171

8.4.2 Algorithms tested 173

8.4.3 Approach to testing the NSGA-II algorithms 173

8.4.4 Test functions 174

8.4.5 Analysis of the test functions 177

8.4.6 Analysis of the building model 178

8.5 Results 179

8.5.1 Test functions results 179

8.5.2 Building model Results 190

8.6 Discussion 192

8.6.1 Test function results 192

8.6.2 Building model results 194

8.7 Conclusions 196

8.8 Summary 196

Chapter 9 Discussion and conclusions 198

9.1 Limitations to the work 199

9.2 Further work 200

9.3 Recommendations to future researchers 200

References 202

Figures

Figure 0-1: Phases of the research and their associated chapters. 4

Figure 1-1: Number of publications on building optimisation by year [7]. (These papers were indexed by SciVerse Scopus of Elsevier) 28

Figure 2-1: The ever decreasing returns against computational effort 35

Figure 2-2: Efficiency vs. effectiveness 37

Figure 2-3: Popularity of search algorithms in research 44

Figure 2-4: Algorithm used by the R package GA 46

Figure 2-5: Main effects on building energy use 58

Figure 4-1: Example of Kriging regression (**x** is the variable input and **z** is the response of the simulator) 68

Figure 4-2: Form of the building used in the simulator 79

Figure 4-3: Plot of the standardised errors of the validation points 85

Figure 4-4: Plot of the Mahalanobis distance vs. its expected probability distribution 86

Figure 4-5: Histogram of percentage errors of the simulator 87

Figure 4-6: Location of points that are greater than 2 standard deviations from the predicted mean function (as predicted by the covariance of the emulator) 88

Figure 4-7: A comparison between the heating degree days at each location for the current and updated TRYs 93

Figure 4-8: A comparison between the cooling degree days at each location for the current and updated TRYs 94

Figure 4-9: Histograms of the difference between the heating and cooling energies of the buildings as well as the change in total exergy (Edinburgh and London) 98

Figure 4-10: Histograms of the difference between the heating and cooling energies of the buildings as well as the change in total exergy (Manchester and Plymouth) 99

Figure 4-11: Energy use vs. heat transfer coefficient - London 101

Figure 5-1: An example Kriging model for a toy function 106

Figure 5-2: An improved emulator for the toy function using an additional simulation point 107

Figure 5-3: A redundant additional simulation point 108

Figure 5-4: Building model to be optimised 110

Figure 5-5: Overview of the algorithm used by the R package GA (this figure is reproduced from earlier for convenience) 112

Figure 5-6: Results of the GA for different population sizes 116

Figure 5-7: Range of the results from Kriging 117

Figure 5-8: Best solution by Kriging 118

Figure 5-9: Best solution by GA 119

Figure 6-1: Range of buildings used in the neighbourhood model 122

Figure 6-2: MC results for the baseline case 131

Figure 6-3: MC results for the improved case 132

Figure 6-4: MC results for the difference of the emulators 133

Figure 6-5: MC results for the 'emulated difference' 134

Figure 7-1: Example of a robust optimisation problem 142

Figure 7-2: Algorithm for creating the confidence level emulator 147

Figure 7-3: Histogram of MC analysis on the robust solution (10 optimisation steps, run 1) 151

Figure 7-4: Histogram of MC analysis on the global solution (10 optimisation steps, run 1) 151

Figure 7-5: Histogram of MC analysis on the robust solution (10 optimisation steps, run 2) 152

Figure 7-6: Histogram of MC analysis on the global solution (10 optimisation steps, run 2) 152

Figure 7-7: Histogram of MC analysis on the robust solution (10 optimisation steps, run 3) 153

Figure 7-8: Histogram of MC analysis on the global solution (10 optimisation steps, run 3) 153

Figure 7-9: Histogram of MC analysis on the robust solution (20 optimisation steps, run 1) 154

Figure 7-10: Histogram of MC analysis on the global solution (20 optimisation steps, run 1) 154

Figure 7-11: Histogram of MC analysis on the robust solution (20 optimisation steps, run 2) 155

Figure 7-12: Histogram of MC analysis on the global solution (20 optimisation steps, run 2) 155

Figure 7-13: Histogram of MC analysis on the robust solution (20 optimisation steps, run 3) 156

Figure 7-14: Histogram of MC analysis on the global solution (20 optimisation steps, run 3) 156

Figure 8-1: Two non-Pareto dominated points 164

Figure 8-2: Demonstration of diversity in Pareto optimal frontiers 165

Figure 8-3: Example of the dominated hypervolume of Pareto optimal set..... 166

Figure 8-4: Crossover operation..... 167

Figure 8-5: Mutation operation 168

Figure 8-6: Steps in the NSGA-II algorithm..... 169

Figure 8-7: NSGA-II procedure (from Deb et al. [164]) 170

Figure 8-8: Building to be optimised in the NSGA-II-based algorithms 178

Figure 8-9: Progressions of the hypervolume of each evolution (Deb 3) 180

Figure 8-10: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for Deb3 181

Figure 8-11: Progressions of the hypervolume of each evolution (Fonseca 2)182

Figure 8-12: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for Fonseca2 . 183

Figure 8-13: Progressions of the hypervolume of each evolution (zdt1) 184

Figure 8-14: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for ZDT1 185

Figure 8-15: Progressions of the hypervolume of each evolution (zdt2) 186

Figure 8-16: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for ZDT2 187

Figure 8-17: Progressions of the hypervolume of each evolution (zdt3) 188

Figure 8-18: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for ZDT3 189

Figure 8-19: Progressions of the hypervolume of each evolution (Building model) 190

Figure 8-20: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for the building model 191

Figure 8-21: Illustration of the NSGA-II algorithm getting 'stuck' during an algorithm generation 193

Figure 8-22: Typical evolution of the NSGA-II algorithm for the building model 195

Tables

Table 4-1: Properties of constructions used in the building model 81

Table 4-2: Properties of the glazing used in the building model (these values are taken from typical argon filled 4-16-4 double glazing) 81

Table 4-3: Make-up of the current and updated TRYs for the UK 92

Table 4-4: Range of variation of the building parameters (note that each variables in continous) 95

Table 4-5: Properties of the building constructions 96

Table 5-1: Constructions used in the building model 111

Table 5-2: Inputs varied in the 9-input building model 111

Table 5-3: Number of training simulations and repetitions of the Kriging algorithm (these were chosen to roughly mirror the number of simulations in each GA routine in Table 5-4) 114

Table 5-4: Populations, iterations, simulations and total repetitions used for the GA routines 115

Table 5-5: Solutions offered by the Kriging and GA approaches 118

Table 6-1: The neighbourhood of building under analysis 123

Table 6-2: Properties of the glazing 125

Table 6-3: Construction properties used in the building 126

Table 6-4: Uncertainty ranges used in the analysis 126

Table 6-5: Example of a Markov chain probability table 127

Table 6-7: Emulator types used in the analysis 128

Table 7-1: Setup for analysing the robust optimisation and global optimisation algorithms 148

Table 7-2: Uncertainty of variables used in the analysis 149

Table 7-3: Solutions offered by each of the optimisation algorithms based on 10 steps 157

Table 7-4: Solutions offered by each of the optimisation algorithms based on 20 steps 157

Table 8-1: NSGA-II test setup 177

Table 8-2: GA setup for the NSGA-II analysis 177

Table 8-3: Test setup for the various NSGA-II test methods 179

Chapter 1 Introduction

1.1 Motivation

The work in this thesis has many motives. The biggest motive is the global need for a suite of building optimisation tools that a capable of delivering buildings that perform well in the field. It is well known that buildings rarely perform as expected. The difference between the expected performance and the actual performance is called which leads to a *performance gap* [1]. Given the impact that buildings have on global energy use, closing this gap will make a much-needed contribution to the reducing global energy requirements. The main objective of my research is to explore the potential uses for Kriging in building design and to evaluate its suitability as a tool for reducing energy use in buildings in the real world.

1.2 Global context

The impact of buildings on global energy use is very large with buildings accounting for around 40% of global energy demand. Given the large contribution of buildings to these emissions, the drive to reduce energy use within them is high.

Many countries have now issued legislation aimed at reducing the emissions from buildings. The EU has set 'legally binding' targets of a 20% reduction in Europe-wide greenhouse-gas emissions (GHG) by 2020[1] [2] and an 80% reduction by

[1] Based on 1990 levels

2050 [3]. In addition to these aims, the UK Government has further legislated for an 80% reduction in GHG emissions by 2050 [4].

Given their large contribution to GHG emissions, we need a strategy for reducing the energy use of buildings. However, developing a strategy difficult, since the energy used is affected by variables that are difficult to control, such as climate, heating technology and the efficiency of building services, and the way the occupants behave in the building. However, I have chosen to focus on energy use, because it is one of the primary factors in good building design.

1.3 Human factors

One of the biggest complicating factors in the design of low-energy buildings is that they have to keep their environments comfortable for their occupants. Factors that might negatively affect comfort include;

- being too hot (overheating);
- being too cold;
- air becoming stale (too much CO_2);
- not having enough natural light (daylighting).

Each of these elements can sometimes work against the desire to minimise energy use. If I did not need to design buildings for humans, I would simply design a very well insulated box with no windows, which would very likely overheat, quickly fill up with CO_2 (if occupied) and would have no natural light. This would clearly not make a good working or living environment. Getting the balance right between energy use and comfort is the major challenge of building design.

1.4 Philosophy of building modelling

Given the range of elements that need to be balanced in a building design, architects, designers and engineers are increasingly reliant on computation models to assist them in their design decisions. However, these models are only useful if they can give a realistic prediction of how a building might function in the real world.

With the advent of fast desktop computers, modelling software is being used more and more in the building design process [5], and is even used (albeit with generally less sophisticated software) as part of compliance processes for meeting various building regulations. These regulations vary depending on the geographic location because they depend on the specific requirements of the local climate.

Building simulators are not just used for predicting thermal performance. Other commonly simulated performance criteria include;

- heating and cooling energy consumption;
- annual lighting energy consumption;
- daylight levels; and
- the level of thermal comfort of occupants.

Unlike other design problems, there is no effective way to prototype a building. Because of this, each new building is effectively its own prototype. Building designers therefore rely heavily on computer modelling to check their designs.

Buildings also present further design challenges. Their uniqueness means that even though the *processes* in a building (i.e. the physics of heat flow through

building materials and air flow through rooms) are relatively well understood, the way those processes interact in their environment and the way in which the building is used is difficult to predict.

Since it is not possible to see how building designs perform 'in the field' *before* construction, building designers have the least real-world data upon which to base their decisions [1], [6]. This is because there are (generally) no requirements in the building regulations for buildings to be monitored post construction.

Furthermore, optimising different aspects of the building's performance will 'nudge' the design in different directions. For example, designing a building to minimise the lighting energy use and to maximise daylight, will most likely end up with a building that resembles a greenhouse. This is good for maximising daylight, but would likely overheat during the summer months.

Given the global implications of buildings that do not perform as expected, developing and improving building design techniques is an important area of research.

1.5 The challenge in building efficient building optimisation tools

The number of publications related to the field of building optimisation has increased rapidly over the previous decades [7]. This is because computers are faster and cheaper (allowing more research to be done at less costs) and because there is an increased drive for more efficient buildings.

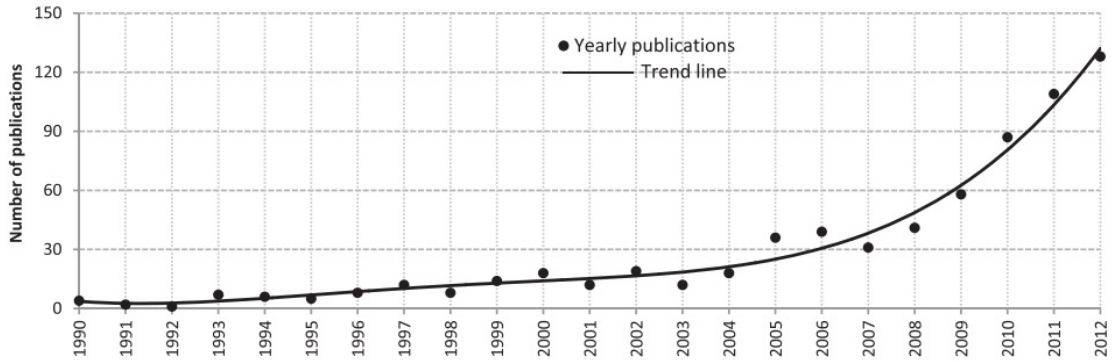


Figure 1-1: Number of publications on building optimisation by year [7]. (These papers were indexed by SciVerse Scopus of Elsevier)

There are two major barriers to the widespread uptake of simulation-based optimisation;

- to reduce the computational power required to search building solutions for buildings with a high number of input parameters; and
- to make the optimised designs more robust to changes in the input parameters.

Any building model with a numerical output can be thought of as a mathematical function. A simple formulation for a one-output simulation problem is,

$$y = f(\mathbf{x}) \qquad \text{Eq. 1}$$

Where the inputs to the model are in the vector $\boldsymbol{x}$, y is the output and $f(.)$ is the continuous simulator function.

For some modelling problems, I might be able to analyse the inputs and outputs so that I can solve the problem analytically and bypass the need for a computational simulator. However, the relationship between the building design and its performance is not analytically intractable. I.e. we cannot derive an

equation for the annual energy use based on the parameters of the building. I therefore need to develop alternative methods of analysis.

In assessing any building model, we should always bear in mind the minimising the energy use in a *simulator* may not always result in a low-building in the real world. However, although this difference is important, the scope of the research in this thesis addresses optimisation and analysis methods only. As future data on building performance becomes available, it is hoped that this can be fed in to the existing optimisation frameworks that have been developed.

Perhaps the most obvious place to start in finding the optimum design is to search all possible input combinations. However, for all but the simplest problems, this very quickly becomes impossible. If we have a discrete search space *I* with *d* dimensions, the number of simulations (n) required to search all possible combinations is,

$$n = l^d \qquad \text{Eq. 2}$$

(where l is the number of search levels and d is the number of dimensions.)

This means that for nine parameter inputs, each with 10 discrete levels in *I*, 9 billion simulations would be required. For a fast simulator taking only 0.001 seconds to run, the search would take three months. If I double the potential values to 20, the analysis time would increase to over 385 million years. The search space also exponentially increases according to the number of dimensions. This problem was identified by the mathematician Richard Bellman and is commonly known as Bellman's curse of dimensionality [8].

As discussed earlier in this chapter, it has been shown that the real-world performance of building is very difficult to predict. There are many uncertain parameters that can have a large effect on the performance of the building. To account for this, there has a been a resurgence of interest in *robust optimisation*[2] [9]–[13]. However, being able to effectively search high dimensional search space as well as developing effective robust optimisation algorithms is very challenging. In the following chapter, I discuss the recent research on this subject and previous efforts to address both of these problems.

1.6 Research Questions

In this thesis, I aim to answer a number of research questions rated to Kriging and building design.

First, I would like to find out whether the Kriging method is a suitable method for emulating the annual energy use of building. Since building models have many input parameters, the size and shape of the potential search space is huge. Previous research has shown that Kriging methods can create effective emulators with only a limited number of training samples [14]. I would like to find out whether or not this finding is applicable to buildings.

The second question that I would like to answer is whether or not Kriging methods can be effectively deployed as an optimisation tool. Tresidder has examined this in his thesis [15]. However, I would like to find out how Kriging compares to other optimisation methods currently used in building design.

---

[2] Robust optimisation is a general term. It refers to optimisation approaches that try to find solutions that are *insensitive* to small changes in the input parameters.

Thirdly I want to explore the potential applications for Kriging methods in examining the effect of uncertainty in building design. Uncertainty analysis is rarely conducted by building designers. This is primarily because this type of analysis takes too long. However, if I can show that Kriging methods can building effective emulators with limited training, then we can use these emulators to perform Monte Carlo analyses and therefore determine the effect of uncertainty. I would like to find out whether this is possible and compare the relative value of Kriging *vs* Brute Force methods for this type of analysis.

Recent research has shown that Kriging methods can be applied to robust optimisation problems. Robust optimisation (RO) is an emergent field in building design. The aim of RO is to provide designs that are resilient to changes in their environment. For example, we need building designs that are resilient to future changes in the climate. Since we don't know for certain what these changes might be, we need design methods that create buildings that are less sensitive to different weather patterns. RO is a practical way to potentially reduce uncertainty in building performance. I would therefore like to explore whether or not Kriging-based RO methods are an effective tool for this.

My final research question is to investigate the effectiveness of Kriging methods in the field of multi-objective optimisation. I want to find out if Kriging-based output approximation methods can be used to improve existing optimisation methods. More specifically, I want to find if Kriging can be used to improve the popular NSGA-II genetic algorithm, which is in common use in building optimisation.

1.7 Thesis Structure

This thesis is structured around the key research questions. Chapter 2 explores recent literature in the field of building design. The literature review explores the different representations of building design in modelling and how the Kriging approach fits in. It also explores areas that are related to the research questions. There is a review of the use of genetic algorithms in building design as well as an exploration of different sources of uncertainty. The review also looks at robust optimisation and how this has been applied (so far) to building design.

Chapter 3 gives an overview of how my research was structured and how I approached answering the research questions. It sets out the boundaries for the exploration of each of the questions and explains how the research was designed to maximise the knowledge gained from each of the experiments.

In Chapter 4 I explore the basics of Kriging and the mathematics on which is it based. I then detail my initial attempts to create a Kriging-based emulator of a building model. I show how I used this model (in conjunction with work contributed by my supervisor, Dr Matthew Eames), to analyse the effect of future weather on the heating and cooling requirements of buildings in the UK.

Chapter 5 introduces the concept of Kriging-based efficient global optimisation (EGO). I demonstrate the basic principles of this method and show how it can be used as an alternative means of optimising a building design. I then compare the performance of this optimisation method to a genetic algorithm. I apply both these methods to a building energy optimisation problem and compare their performance. Since both methods have an element of probability associated with

them, I repeat identical processes a number of times and then compare the variability in the results.

In Chapter 6 I examine how Kriging can be used as uncertainty analysis tool. This chapter details how a Kriging model was used to build a surrogate model of a building. The emulator is validated and then used to emulator the energy performance of a neighbourhood of buildings. Each building in the neighbourhood is assigned a particular set of construction parameters. The buildings also have their own uncertainty parameters associated with each construction element. I perform a Monte Carlo (MC) analysis on each building using two methods. The first method runs the MC analysis using brute-force methods (i.e. using a fully-fledged building simulator). The second method uses the Kriging emulator. I then compare the results of these two approaches are then of these two approaches.

In Chapter 7 I examine the use of Kriging methods for robust optimisation. This chapter introduces a method developed by Rehman et al [16] which uses a Kriging emulator to find the robust optimum of a single-objective design. In this chapter, I apply and develop this approach for use in the context of building modelling. I demonstrate the use of the methods on a building model as well as validating the method against a number of test functions.

Chapter 8 is the final research chapter. It looks at my efforts to improve the NSGA-II multi-objective optimisation function. In this chapter I show how Kriging approximation can be incorporated into the existing algorithm to improve its performance. The performance of the Kriging-assisted algorithm is compared to the performance of the original algorithm. The performance of both algorithms is

investigated for a number of test functions as well as a complex building optimisation problem.

The final chapter reviews the contribution of my research to the large body of knowledge of building optimisation and design and reviews potential options for further work.

Chapter 2 Literature Review

2.1 Introduction

Effective building design tools should be able to successfully uncover suitable designs in large decision spaces, but they must also be capable of finding designs that are robust to uncertain building parameters. There is also another issue, which I explore in this section, that of *computational efficiency* and *computational effort*.

The idea of *computational efficiency* can be applied to both the optimisation algorithm and the underlying simulation tools. The efficiency is greater if the optimisation algorithm or simulator takes less time to obtain a result. It is therefore desirable that both the optimisation algorithm and the simulator are as efficient as possible.

I also consider the idea of *computational effort.* One measure of computational effort is the number of floating point operations (FLOPs) carried out – the more FLOPs used, the greater the effort. It is often the case that optimisation algorithms offer ever decreasing returns for additional effort.

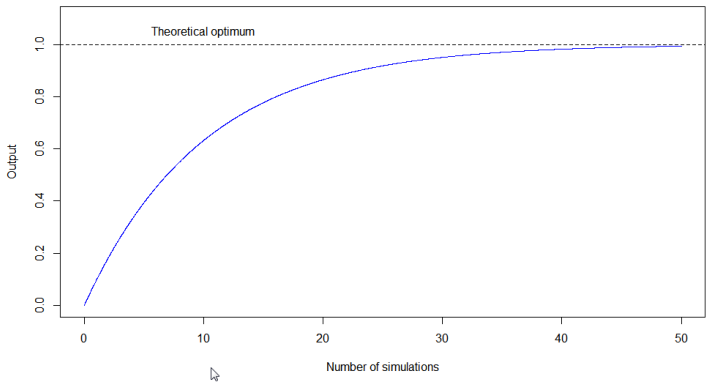


Figure 2-1: The ever decreasing returns against computational effort

Optimisation routines often makes good progress early on, but improves less over time.

The process of most optimisation algorithms tend to be asymptotic; as the algorithm progresses towards the 'optimum' result, the improvements become less and less. If I see computing power as a proxy for capital, then I am getting less and less for my money. The challenge therefore, is working out how I can best spend a computational budget to ensure that I get the most possible improvement.

2.1.1 Efficiency and effectiveness

When considering how best to spend computational power, I need to consider both efficiency and effectiveness. It is, therefore, important to make the distinction between *effectiveness* and *efficiency*:

- *Effectiveness* is the description of how well an algorithm covers the parameter space to find the global optimum; and
- *Efficiency* characterises the quickness with which the algorithm converges.

When considering the results obtained by the simulation model and the optimisation algorithm, both the efficiency and effectiveness need to be considered. In Figure 2-2, algorithm 2 can be considered more efficient than algorithm 1, even though algorithm 1 tends towards a more 'optimum' result overall. In other words, algorithm 1 is more *effective* and algorithm 2 is more *efficient*.

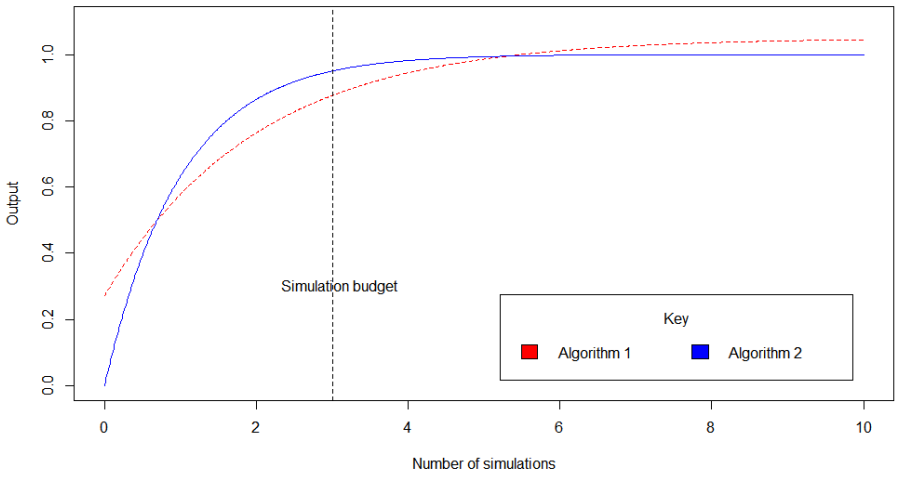


Figure 2-2: Efficiency vs. effectiveness

The ideal scenario is where an algorithm is both *effective* and *efficient*, but in most cases, there is a trade-off between these two characteristics.

2.2 Commercial software

The needs of architects and engineering are different. Architects need tools that can quickly evaluate early stage design options, whereas building services engineers will require more detail in order to specify heating, ventilation and air conditioning equipment that can serve the building's needs.

There is a wide variety of building simulation tools on the market to address both these needs [5]. The most commonly used commercial building simulation programs include *IES Virtual Environment* [17], *TAS* [18], *Design Builder* [19] and *Ecotect Analysis* [20].

Each of these software packages can perform a range of tasks. For example, *IES VE* has a number of separate modules, including a computational fluid dynamics module (*MicroFlo*), a daylighting module (*FlucsDL*) and energy prediction tools (*Apache)*. *DesignBuilder* includes an optimisation package called *DesignBuilder*

Optimisation. The optimisation algorithm uses the non-dominated sorting genetic algorithm (NSGA-II) [21] and the *EnergyPlus* simulation engine [22].

Further investigation of the optimisation methods use in these programs is difficult, due to their proprietary nature. So, instead of reviewing the technical aspects of commercial software, the general trends in the development of commercial simulation packages are now considered.

2.2.1 Building information modelling and Cloud computing

Building information modelling and the advent of *cloud computing* are two recent developments that have the potential to change the way that building professionals design buildings [23].

Currently, the architecture, engineering & construction sector is a highly fragmented, data intensive, project-based industry, involving a number of very different professions and organisations [24]. These include (but are not limited to) landscape designers, building service engineers, acoustic consultants, architects, project managers, quantity surveyors, environmental consultants, lighting consultants, planning consultants, structural engineers, civil engineers and many others. BIM is intended to improve the communication and sharing of information between these different disciplines by providing a central database for the building's data.

Sharing data between different professions tends to happen on an ad-hoc basis, with the design team sharing documents and information through services such as Dropbox and Google Drive [24]. Several companies are attempting to solve this problem by providing centralised storage. These companies include, *Onuma*

[25], *Revit Server* [26], *ProjectWise* [27], AssetWise [28], *Graphisoft BIMCloud* [29], and EDMmodelServer [30]. Many data storage services now include cloud computing services model simulation [31].

There are also government-backed drives to increase the use of BIM-style information sharing. For example, in the UK, the government already requires BIM to be used for some publicly funded projects [32], so it is likely that the trend towards centralised information storage and cloud computing will continue.

The move towards BIM could give building professionals access, not only to information, but to much greater computing power. It is not difficult to see how this could improve the building modelling process. This has the potential to lead to a greater uptake of optimisation, but there are still many technical barriers to overcome before the use of BIM is common [33].

2.3 Academic Research

2.3.1 Introduction

There are a wide range of methods for representing buildings in thermal models. I broadly classify these as either *engineering models* or *statistical models*. Engineering models *physically* represent building performance. Example include finite element models, state space models and simplified resistor capacitor (RC) network models [34][3]. Common statistical methods include linear regression, multivariate adaptive regression splines, artificial neural networks, radial basis

[3] RC models use resistors and capacitors to represent insulation (resistance) and thermal mass (capacitance). In these models, voltage is analogous to temperature and current is analogous to heat-energy transfer.

function networks and Kriging (among others) [35]. Although artificial neural networks they are not strictly speaking a statistical method, we include them in this list because they do not *physically* represent the building.

Clarke [36] describes the currently available software as a work in progress towards a;

> *'truly powerful computational approach to design whereby arbitrarily complex models may be evolved on a task-sharing basis, such models readily exchanged and understood by others, industry standard assessments automatically invoked, and seamless integration within the temporally evolving design process assured'.*

To achieve this 'computational approach to design' it is essential that I develop effective and efficient tools. The following sections discuss the efforts to achieve this aim from the point of view of building energy simulation (BES). The potential pitfalls of BES are also discussed as well as some of the most common sources of uncertainty in the output of BES models.

2.3.2 Engineering models

Engineering models are simulation tools that directly model the thermal processes that occur in a building. To date, a large number of different simulation tools have been developed, all of which vary in complexity. It is impossible to document them all, but some of the most complex and capable tools in use in research include *EnergyPlus* [22], *ESP-r* [37] and TRNSYS[38].

Effort has also been spent on creating engineering models that are quick to execute. These are typically based on simplified engineering models. These tools

include simplified lumped parameter methods [34], [39], [40], manual techniques such as the *LT* method [41] and simplified tools for specific purposes, such as estimating the transient thermal performance of a building [42].

Each of these tools has advantages and disadvantages. Simple models (i.e. those that do not include some of the subtler complexities of building physics) are more amenable to the multiple iterations required by optimisation algorithms, since they are faster to compute. However, their simplification means that they lack detail and accuracy. Complex simulation tools are often seen as being more accurate, but they take longer to run and are therefore less amenable to analysis. Much research has therefore been undertaken to improve complex analysis tools as well as developing tools the balance complexity and 'accuracy'.

Picco et al [43] have designed a methodology for creating a simplified model of a commercial building. In their paper, they detail progressive simplifications of a building model. These include the simplification of the constructions, the removal of external obstructions, zone lumping, the simplification of transparent surfaces, the standardising of floors, the 'squaring' of the zones and the standardising of transparent surfaces. Using a two-output model (the annual heating and cooling energy use), they show that the most simplified model diverges around 15.6% for heating loads and 14.6% for cooling loads, with differences of only −4% and −9% with respect to peak loads. This has shown that building models can be simplified dramatically without the divergence from their 'true' output being too great.

De Rosa et al. have also used a simplified model in their research [44]. They tested and validated a model based on a simplified dynamic model, similar to that proposed by Ramallo [45]. They go on to use this simplified model to create a

modified degree-day-based approach to analyse the relationship between the number of degree days, the building's thermal inertia and energy consumption. Their results demonstrate that, with appropriate changes to the calculation of the number of cooling degree days, a linear relationship between the cooling degree days and the building's energy use is observed.

Ogunsola and Song [34] have also investigated a simplified dynamic model, again for estimating the heating and cooling load. Their research uses a thermal network model, which they solve analytically. The performance of this model is compared to time-series results generated by *EnergyPlus*. The results show that the output of *EnergyPlus* and the simplified model match well. However, estimating the parameters of the RC model is complex and the authors report that the computational time required to estimate these parameters is significant.

2.3.3 Statistical and meta-modelling techniques

Statistical techniques are another approach to the 'simplification' of building models. However, whereas engineering models have elements that simulate the thermal performance of the building (as in RC thermal network models), statistical methods do not. Statistical methods model the *relationship* between the inputs and the outputs.

There are a wide variety of statistical techniques that are involved in the building design. These include polynomial regression, multivariate adaptive regression splines, Kriging, radial basis function networks and neural networks [35], with polynomial regression being one of the most popular statistical methods used in research [46]–[50].

Gasparella et al. [46] have used polynomial regression to investigate the impact of different kinds of glazing systems, window sizes and orientation of facade on the thermal performance of a residential buildings. The heat gains from solar radiation during the winter and summer for buildings in Paris, Milan, Nice and Rome were assessed using polynomial regression trained on results from TRNSYS simulations[38]. They make a number of conclusions about glazing types and orientations suitable for minimising the effects of overheating in summer and maximising useful solar gains in the winter.

Eisenhower et al. [51] have used a support vector machines (SVM) to create a meta-model-based approach to building optimisation. Their analysis bases the meta-model on a Gaussian kernel, which is then fitted to a set of *EnergyPlus* training simulations. A parameter sensitivity analysis is then conducted with both the SVM model and the *EnergyPlus* simulator. The results of this analysis show that the output of both models compare well, with the mean energy use predicted varying by only −0.02%.

Ferreira et al. [52] has investigated the use of an artificial neural network (ANN) for predicting the best building control strategies for heating, ventilation and air conditioning (HVAC) systems for the thermal comfort of the building's occupants. Different rooms in the building of the University of Algarve were monitored and assessed using the control strategy (in summer and winter conditions). The results of these assessments showed that the application of ANN-controlled HVAC reduced the energy consumption by more than 50%. This demonstrates a useful application of ANN for building control.

It is clear that statistical methods have a variety of applications within building models. However, a review of building optimisation techniques would not be complete without considering the role of genetic algorithms (GAs).

2.3.4 Genetic Algorithms

Genetic algorithms (GAs) are by far the most commonly applied optimisation technique in building design (Figure 2-3) [7].

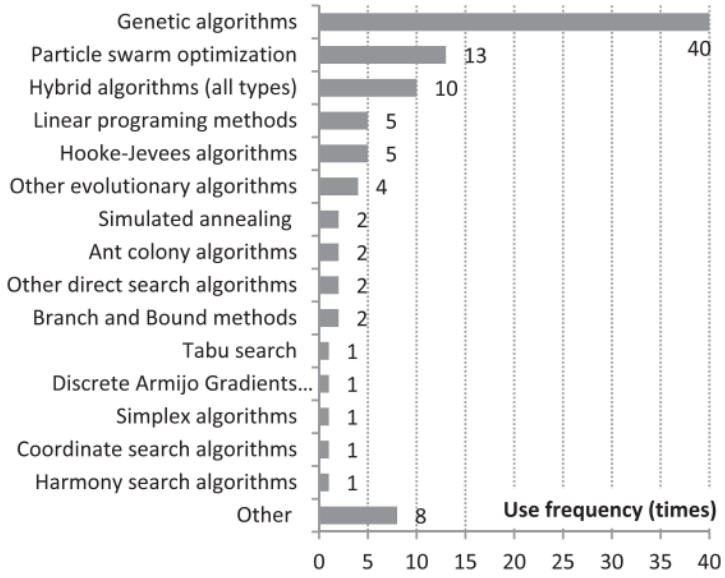


Figure 2-3: Popularity of search algorithms in research

GAs are a subset of evolutionary algorithms (EAs) and are used widely in research applications such as building envelope optimisation [53]–[55], energy conservation measures [56], HVAC design [57] as well as multi-objective optimisation [58]–[60].

The main advantages of GAs are that they are;

- capable of handling both continuous and discrete inputs;

- capable of allowing parallel processing of simulations (i.e. populations of simulations can be evaluated *at the same* time, which, by definition, is not true of sequential optimisation methods);
- well suited to multi-objective optimisation;
- suited to objective functions that have discontinuities;
- capable of handling objective functions that have multiple local optima, since they are less likely to get 'trapped' in one local minimum [7].

However, in order for GAs to optimise a building efficiently, the parameters that control them must be carefully chosen. A typical GA creates an initial 'population' of buildings and uses the operations of selection, mutation and crossover to generate subsequent (hopefully improved) buildings (Figure 2-4). A full description of fitness evaluation, selection, mutation and crossover is given in section 8.2.4.

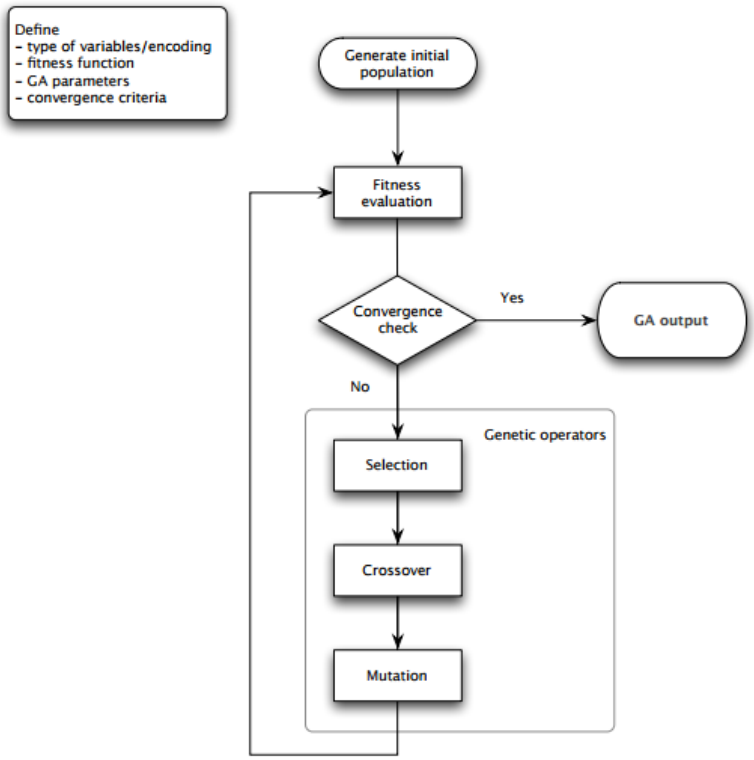


Figure 2-4: Algorithm used by the R *package* GA

Alajmi and Wright [61] have shown that the size of the initial population, as well as the crossover probability and mutation rate, all have significant effects on the performance of the GA. Haupt has shown that smaller population sizes work well along with GAs that have high mutation rates [62]. Alajmi and Wright have also shown that the population size has the biggest effect on performance and that the crossover and mutation rates are less significant. They also found that binary-encoded GAs with small population sizes of around 5 were most effective.

GAs are frequently used for multi-objective optimisation. Of the GA methods used for multi-objective optimisation, the non-dominated sorting GA (NSGA-II) is widely used in research [21], [58], [63]–[67].

NSGA-II have been shown to outperform both Knowles and Corne's Pareto-archived evolution strategy [68] and the Zitzler's strength Pareto evolutionary

algorithm [69]. Deb has shown that the NSGA-II performs better both in terms of finding a more diverse set of solutions and in converging more quickly to the true Pareto-optimal set [21].

In their recent paper, Brownlee and Wright have shown that NSGA-II can be improved with the addition of fitness approximation using a meta-model based on radial basis function [66]. In their paper they have applied variants of the NSGA model to optimise for operational energy and construction costs. These variants used the meta-model to assist in sorting the output population. By simulating a simple commercial office building using *EnergyPlus*, they showed that using the radial basis function augmented algorithm resulted in improved run times for the function as well as an improved final solution.

2.3.5 Multi-objective optimisation using NSGA-II in buildings

NSGA-II is a multi-objective optimisation algorithm which has gained popularity in building design (as well as engineering optimisation in general). NSGA-II is in use in both commercial software as well as peer-reviewed research papers in building design [58], [63]–[66], [70]. It is the most commonly-used GA-based algorithm in building design [71].

Machairas et al have reviewed the use of algorithms in building optimisation [72]. In their research they have investigated work done by other researchers to find out which optimisation algorithms and are generally employed, and why. Nearly all of the paper cited in their research use some form of GA. Of these papers, many used NSGA-II algorithm including Chantrelle et al [64], Magnier and Haghghat [58], Evins et al [73] and Palonen et al [74].

Chantrelle et al have used NSGA-II to create an optimisation tool to assist in finding optimal ways to renovate buildings [64]. They use the NSGA-II as part of an optimisation package they developed. The optimisation package, *MultiOpt,* is based on TRNSYS [38] and uses a NSGA-II to create optimal designs. It also uses databases of economic and environmental data to produce data which can be used in the optimisation process.

In their research they test *MultiOpt* on a four-output simulator optimisation, whose outputs included energy consumption, costs, thermal comfort and life-cycle environmental impact [64]. The results of their assessment showed that the analysis time for their algorithm was significant, but not excessive (although no detailed information was given on analysis times). They also compared the results of the NSGA II-based to the results of alternative assessment software and found the results to be in agreement with each other. They concluded that *MultiOpt* is was an effective tool for aiding design decisions in building renovations.

Evins et al [73] use NSGA-II as part of a three step optimisation process on a two-bedroom, mid-level flat. The steps in the optimisation process are:

1. Perform a Design of Experiments (DoE) analysis to determine which inputs are significant.
2. A ‘less detailed’ multi-objective algorithm is applied to all the variables that are identified as significant. This optimisation is performed by NSGA-II on a simplified building model. This optimises the building for carbon emissions, running costs, capital cost and is constrained by roof area and overheating.

3. A ‘more detailed’ multi-objective optimisation is undertaken on variables that exhibit complex behaviour in *step 2.*

Both optimisation procedures in steps 2 and 3 use NSGA-II, but the optimisation is performed on models with different levels of detail. The first and second steps in the process are performed on a less detailed model. This allows non-important variables to be eliminated quickly. The third phase allows a more detailed exploration of the model using only variables that are found to be significant.

In their results, Evins et al. show that, as the design evolves, the variables do not move towards their optimum values in a linear way. For example, the heating system capacity moves in cyclical way as the design progresses. In an optimisation procedure, we would normally expect the parameters of this system to move in a more linear fashion towards the optimum value. However, Evins et al. find this not to be the case. They concede that many of their findings are project specific and that it is difficult to draw conclusions about other building types.

Carlucci et al have used NSGA-II to design a nearly-zero energy use building that also minimises thermal and visual discomfort [75]. Their research details how a four-output simulator was optimised using a range of discrete values for variables including the U-values of the wall, roof and floor as well as the U-values of the glazing, control strategies for shading devices, and percentage openable area of windows. Carlucci et al report that the NSGA-II algorithm is effective at finding a range of non-intuitive solutions to the design. They also found that the algorithm was effective in finding solutions with relatively few calls to the original building simulation function.

2.3.6 Sources of uncertainty

In trying to minimise the energy consumption and operational costs of buildings, we must consider the effect of uncertainty in many of the parameters and assumptions that are used in building modelling. One of the key sources of uncertainty are the occupants. As we shall see, the behaviour of occupants is unpredictable and leads to significant uncertainties in the predicted performance of the building. However, there are other sources of uncertainty that contribute the performance gap between the predicted and real-world energy use [1] - these will also be explored.

It has been shown that, wherever possible, the buildings occupants will attempt to maximise their thermal comfort. However, as Fanger [76] has shown, different people have different ideas about what makes them comfortable. Some of these factors are long habituated (for instance, people who migrate from hot to cooler countries may find that they need more heating) and some of these are due to acclimatisation due to recent phenomena (such as an extended heat wave or a cold snap). Therefore, it is not only difficult to predict the effect of occupants' actions, but it is also difficult to predict what actions the occupants will take.

The kinds of actions the occupants take vary, from opening windows to adjusting heating controls as well as the operation of electrical and lighting equipment [77]–[84]. Overall, these effects contribute to a great deal of uncertainty in the results of building simulations. It is likely that these effects contribute the difference in energy use between that predicted by deterministic building models, and that achieved in practice; the so-called *performance gap* [1], [85]–[87].

Da Silva et al. have compared occupant behavioural *models* to real world occupant behaviour [78]. In their paper, they consider the occupants' effect on the control of lighting and shading devices in offices. They undertook a two-month assessment, making detailed measurements and observations of the actual occupants' behaviour and comparing these to the assumed occupant models in the building simulation. They found that there were a number of important implications for the occupancy models used in simulation;

- the models themselves vary considerably in the outputs that they achieve
- lighting models work relatively well, but do not always agree well with real-world performance
- regarding shading, they found that the occupant dynamics were mostly related to individual instances of extreme solar incidence (glare)
- models predicting closings of shades are better than the models that predict their openings

In addressing these conclusions, da Silva et al. recommend a probabilistic approach as an alternative to deterministic occupancy models [78].

There are clear advantages to the probabilistic methods, but also drawbacks. In nearly all cases, probabilistic methods require a greater number of simulations. These additional simulations are needed to assist in quantifying the effects of changing single variables. On the other hand, deterministic methods take less time to run, but they do not give the designer any information about how any changes to their assumptions might change the results of their model. These types of considerations are clearly important when considering the effects of building variables that are likely to be influenced by the building's occupants.

Haldi and Robinson have studied the impact of occupancy on the building's energy demand [88]. In their study [88], they made use of extensive field survey data taken over a period of eight years at the Solar Energy and Building Physics Laboratory at EPFL in Switzerland. They used this data to develop detailed models of occupant presence and for the opening and closing of windows and blinds. One of the main findings is that the occupants have a significant impact in that they can affect the energy consumption of the building by a factor of two [84] and that this impact has the potential to be greater due to the diversity of the occupants.

Azar and Menassa [89] have also investigated the effect of occupancy in the commercial sector. However, they take a different approach to Haldi and Robinson by performing a sensitivity analysis on the inputs of the building model to determine the potential impact of occupants. In their analysis, which ranges across a number of climate zones in North America, they found that the single greatest impact on the total energy use was the heating set point, particularly in smaller-size buildings.

One of the key drivers of occupant behaviour is thermal comfort. Ioannoua and Itard have considered the interactions between thermal comfort, the sensitivity of building parameters and their effect on energy performance [82]. In their paper, they conduct a Monte Carlo (MC) analysis to assess the influence of occupant-related and non-occupant related parameters in the building's energy performance. They found that, of the factors that were not influenced by the occupants, the window's U and g-values, along with the wall conductivity were the most critical parameters. However, when considering the occupants behaviour, the findings were similar to that of Haldi and Robinson [88], showing

that both the heating set point and the ventilation parameters were most significant. Furthermore, the effect of these parameters was many orders of magnitude greater than that of the building parameters.

Day and Gunderson have conducted detailed investigations on how the level of occupiers' knowledge of the building system can affect how they use *high performance buildings* [77]. In their paper, they acknowledge the need for occupants to be *actively engaged* in the operation of the building, and to understand how their actions affect the overall energy use.

In their research, they studied the relationship between occupant behaviour, environmental satisfaction and the training they have received in the way that high performance buildings operate. They found that individuals who had been trained in the buildings operation were more satisfied with their level of comfort. They conclude that, for buildings to be successful in their designs, they must give the occupants some control over their environment. However, this control must also be accompanied by training so that the occupants understand better the interaction between their actions and their influence on the building.

We can see from the above research that the effect of occupants on buildings is complex. We have a wide variety of potential interactions, as well as a wide variety of building types to interact with. However, no matter which modelling approach we take, there exists the potential for large margins of uncertainty in the outputs of the design. These uncertainties originate both in the inputs to the model (the parameters that model how the occupants behave) and the fidelity of the modelling itself (i.e. how well the model we have created can represent the 'real world').

It is clear that if we can create comfortable buildings, the occupants are less likely to interact with the building in a way that adversely affects the energy use.

2.4 Other sources of uncertainty

There are many other source of uncertainty in building modelling. The uncertainty about the climate and micro-climate effects can also be significant. De Wit and Augenbroe have identified two key drivers of climatic related uncertainty [90]:

- the uncertainty in the ventilation rate of building space; and
- Uncertainty in the room temperature distribution.

They found that the uncertainty in the ventilation rate is largely due to the uncertainty in the wind pressure coefficients. One cannot know the exact wind pressure coefficients are at different points on the building envelope (this is especially difficult before the building is built). De Wit and Augenbroe found that this difference between the assumed and actual wind pressure coefficients it a significant contributor to uncertainty.

De Wit and Augenbroe also found that uncertainty in the room's air temperature distribution adds significantly to the overall uncertainty in the energy performance of the building. The room air temperature is rarely uniform. To model room temperatures more accurately, methods which solve the Navier-Stokes equations could be used. This is typically achieved using computational fluid dynamics (CFD). However, CFD is rarely employed in building design – not because it is not accurate, but because the individual simulations simply take too long to be commercially viable. Instead, most building simulation software uses simplified empirical equations.

The simplification of both wind pressure coefficients and the internal air temperature distributions adds significant uncertainty to the modelling process. De Wit and Augenbroe suggest that the choice of both these variables has a profound effect on the results of building performance and considering the uncertainty in these values is essential.

2.4.1 Overheating and future climate

The importance of minimising overheating in order to ensure the comfort and productivity of occupants is well understood [77] but the problem of overheating in both domestic and commercial buildings remains problematic. There is a growing body of research that demonstrates the need to consider the increased risk of overheating in a future climate [91]–[97].

The problem of overheating in building design is two-fold. First, the projected changes to the climate are likely to result in weather conditions that are more conducive to overheating [92], [94]–[96]. Second, since we need to design buildings that use less energy, these building designs are typically more prone to overheating due to high levels of insulation and low levels of infiltration [95].

In an attempt to tackle this problem, Banfill et al [95] have created an overheating risk tool based on a simplification of the UKCP09 probabilistic climate projections [98].

Determining the likelihood of overheating can be particularly complex in domestic buildings. Ji et al. [99] have assessed the overheating risk for a replica of a Victorian terraced house in the UK. In their paper they examine the thermal performance of the terrace for existing and future weather after an extensive

energy-performance related retrofit. They found that overheating was most likely to happen in bedrooms, and could begin around 2020 (based on the single temperature criterion in CIBSE Guide A [100]). This is of particular concern because people tolerate high temperatures much less during sleep than during the day.

Nik and Sasic Kalagadsidis [97] have also studied the impact of climate change on buildings; this time looking at how the energy performance is affected by climate change for 153 buildings in Stockholm, Sweden. They considered the effect of four uncertainties in the climate data which affected the global and regional climate models used. Each of the buildings had a range of different ventilation strategies, including mechanical, natural and hybrid systems. The results of the assessment show that the heating demand decreases in each of the climate scenarios. In the case of the period 2081-2100, the heating consumption is predicted to be around 25-30% less than in 2011. However, the amount of heat that is projected to be used in the future is highly dependent on the climate scenario.

The cooling load is also dependent on the climate. Nik and Sasic Kalagadsidis [97] showed that this can vary by as much as 500% depending on the climate model used. These results show that the climate can add significant amounts of uncertainty to the potential energy use of a building.

Localised changes in the climate can also have a significant effect on the energy performance of a building. The urban heat island (UHI) can be a significant contributor both to overheating and therefore cooling load requirements [101]–[106]. Oikonomou et al. [107] have investigated the effect of the UHI in London

on the internal temperatures of a variety of building types. The results showed that the level of variation in internal temperatures is high. An important conclusion of the work is that it is the dwelling design, rather than the UHI, that needs to be improved in order to have the most effect in reducing periods of overheating.

The above research shows that the optimisation of buildings to reduce the changes of overheating is a complex task with many variable and wide-ranging inputs. The research also shows that importance of being able to design buildings which are robust to these changes. The challenge is to effectively search for these robust designs within complex building and neighbourhood-level models, as well as to be able to consider the effects of uncertainty in building designs.

2.4.2 Uncertainty analysis

In their 2011 paper, Hopfe et al. [108] provided a realistic case-study of the use of uncertainty analyses (UA) techniques using MC analysis in building design. The experiments also included the use of Latin Hypercube designs and analyse uncertainty in the *parameters* of the building model. Other sources of uncertainty, such as type C and D uncertainties as defined by Sendhoff [109], are not included[4].

Their UA analysis used step-wise regression to identify the most uncertain parameters that affect the key performance indicators (i.e. annual heating and cooling).

[4] A full discussion of the uncertainty types introduced by Beyer and Sendhoff is given in section 7.2.

The results of Hopfe's sensitivity analysis show that both the annual heating and cooling loads are very sensitive to changes in the infiltration rate (which is to be expected). Other factors, which might be expected to be more influential, such as the U-value of the glazing have less effect (Figure 2-5).

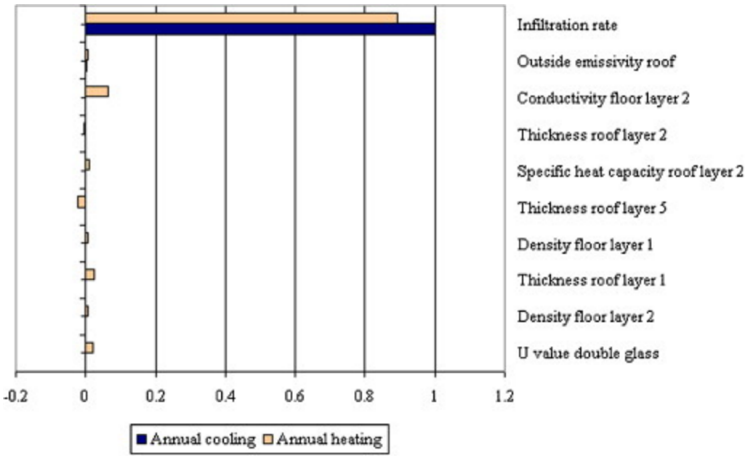


Figure 2-5: Main effects on building energy use

De Beock [5] cites the following sources of uncertainty in building modelling;

- occupant behaviour;
- daylighting control strategies;
- material property uncertainties / design parameters; and
- climatic data

However, of these factors, occupant activity is generally considered to be the largest source of uncertainty in the predicted energy use of the building [83], [110].

Blight and Coley [81] have conducted a sensitivity analysis of the effect of occupancy behaviour on the energy consumption of residential buildings built to the *PassivHaus* [111] standard. They analysed the relationship between occupant behaviour and energy demand using regression analysis for 100

residential units. They found that such low-energy buildings were less sensitive to occupant behaviour than previously thought. One of the research outputs was a regression equation that uses certain occupancy variables to predict the impact on energy use.

De Wit and Augenbroe [112] have investigated how UA methods can fit into the design process to support building design decisions. They note that most design decisions are made in the absence of any sort of uncertainty analysis in the building parameters [112]. Furthermore, they note that there is particular difficulty in achieving any kind of statistical analysis since there is a very limited amount of data on which to base the statistical distributions of building parameters. In their paper, they give a brief overview of:

- data sources on the uncertainties in material properties
- techniques for determining these uncertainties
- uncertainties stemming from the model simplifications

De Wit and Augenbroe identified that there were two dominant sources of uncertainty in the outcome of the energy simulation. These are:

- uncertainty in the ventilation rates of the building spaces; and
- uncertainty in the room air temperature distribution

Walter et al. have considered the wider impact of uncertainty in predicting the performance of energy conservation measures [113]. They cite the main problem in implementing energy efficiency measures are determining whether or not they have worked. This is more of a problem than it sounds, since the environment in which the building is operating is always changing due to the effect of the

occupants and daylight levels, as well as weather conditions and other environmental factors.

We need a model to predict what the energy use would have been before any energy conservation measures were installed. The prediction of this model is the *baseline energy use*.

They point out that, in analysing the potential effects of energy efficiency measures, designers need to be able to consider the effects of uncertainty on the predictions of the baseline energy use (i.e. the energy use before the retrofit measures) and the post-retrofit energy use. However, they note that most statistical methods that have been reported in the literature do not quantify the effects of this uncertainty adequately.

Much of the research into the effects of uncertainty in the energy saving predictions of energy policy has focused on macro-level parameters (the *top-down* approach), such as the uncertainties introduced by socioeconomic factors, national policies and the climate change [114]–[116].

Kagvic et al. [117] have reviewed *bottom up* approaches to modelling energy consumption in the residential sector. In their research, they identified three major issues. The first was the lack of publicly available data relating to the inputs and assumptions used in building models. The second issue was the lack of data on the importance of the modelling parameters on the output. The third and final issue was the uncertainty in the socio-technical drivers of energy consumption. (In other words, the uncertainty about how the occupants operate the building.) All of these issues clearly show that any building model will need to take account of these uncertainties.

Booth & Choudhary [118] have developed a framework for dealing with the uncertainties at the building level to calculate the risk of the overestimation of energy savings. They include uncertainties in the heating set point, the fraction of the space heated, infiltration, coefficient of performance (for mixed heating systems), window-to-wall ratio and the U-value of the glazing. In their analysis however, they undertake MC analysis of the uncertainty using an engineering model, rather than an emulator.

MC analyses are common in building design, and are frequently used in research as a tool to estimate uncertainty. However, in order for MC analysis to be completed in a reasonable time-frame, the models they are operating on need to produce the results quickly. For this, as we have seen, much research has focussed on statistical methods and reduced-order models to allow these analyses to be completed in a reasonable time-frame. Although MC methods are suitable for determining the uncertainty in the output of a design, methods are required to help building professionals find designs that are robust to these uncertainties.

2.4.3 Robust optimisation

The concept of robust optimisation has been around since the mid-20th century [119] and has been applied across a wide range of fields, from finance, statistics, learning, and engineering [120]. Work on robust optimisation continued in the 1970s [121], [122] and underwent a resurgence in the late 1990s [123]–[127].

Robust optimisation has many different definitions across many different fields. In the context of building design, Hopfe et al [128] have defined robust optimisation as;

"not only to optimize the objectives based on deterministic inputs, but also to take care of deviations of objective function values caused by small or large changes or fluctuations in the input variables."

Hopfe's paper considers RO where the uncertainty model is not stochastic, but rather deterministic and set-based [120]. Other approaches, such as that taken by Ramallo et al. [11], have used a *changing environments evolutionary strategy* which is based on changing the *environment* in which the population is emulated on each generation of the evolutionary algorithm. Ramallo et al. demonstrate this technique on a simple domestic building model and show how the model is made more robust to occupancy and weather changes. Further approaches to robust optimisation are considered in detail in Chapter 5

2.5 Summary

In this literature review, the general principles of optimisation have been considered. It has also been shown that there is a limited amount of commercial software that is capable of conducting sophisticated optimisation routines. I have also reviewed recent research. This review has shown that research has increased greatly over recent decades, and a wide variety of tools are now available to assist in the design of buildings. However, there are two areas where there has been limited research, or where a significant amount of further research is required;

- the robust optimisation for building design;
- the use of meta-modelling to enhance existing building optimisation algorithms

In this thesis I introduce and explain Kriging meta-modelling approaches to solving these optimisation problems.

In the following chapters, Kriging is used to create a simple meta-model of a building. I demonstrate that this method is effective for modelling the annual energy use, this approach is then used to analyse the effect of the UK's new Test Reference Year on different building types across this UK. Kriging is then applied to a model of a neighbourhood of buildings to assess the uncertainty in the energy performance across 20 different buildings. Following this, a robust optimisation algorithm is implemented for single objective optimisation. Finally, I demonstrate and analyse the use of Kriging to augment the NSGA-II algorithm for a two-objective building optimisation problem.

Chapter 3 Research Design

I intended that my research cover a range of possible applications for Kriging in building design, and I designed my research to reflect this.

I broadly divided my investigations into three key areas:

1. Initial assessment
2. Direct applications
3. Advanced / hybrid applications

In my initial assessment, my aim was to assess how good Kriging was at capturing the annual energy use of buildings and to determine how well it captures the response of the building model to changes in the input parameters. These results are verified using a number of method developed for the verification of Kriging models. Once verified, I then used a simple building emulator to look at the potential impacts of new design summer years (DSYs) on building design (an analysis that would not have been possible without fast and accurate emulators).

The second phase of the research look at direct applications of Kriging techniques that have been used in other fields. These included optimisation techniques (*efficient global optimisation* otherwise referred to as EGO), and Monte Carlo uncertainty analysis. The EGO method is compared to a genetic algorithm optimisation approach and the results compared. For the MC uncertainty analysis, the results are compared to a 'brute-force' approach (i.e. an approach where the same MC analysis is undertaken with the simulator model only).

The third and final phase look at more novel and advanced applications. In this part of the research, I look at using Kriging optimisation techniques to create a robust optimisation algorithm for building models. I test this method using a number of test functions (to assist in validating the technique) as well as using the method on the simulator only.

I also wanted to see if Kriging could improve the NSGA-II algorithm by using it to improve the population generation and sorting process. To determine whether the method is effective, I compared the Kriging method to the original NSGA-II. To do this the performance of the Kriging improved optimisation procedure was compared to the baseline NSGA-II performance. This was done with both a series of carefully chosen test functions as well as a building simulation problem.

3.1 Rationale for choice of experiments

In choosing the experiments in this thesis, I have introduced a wide range of applications. This is intentional. In my research, I wanted to explore Kriging as a tool that could potentially be applied to many aspects of building design – from early stage analysis of the design to a tool for the statistical analysis are large groups of buildings. I therefore designed my research to reflect this. Whilst this does not necessarily look at any one area in exhaustive depth, it will hopefully introduce the reader to a range of potential areas of future work, whilst providing some initial findings about is effective and what isn't.

I wanted to start with simple applications and progress to more complex ones. Starting with more simple experiments means that they can be easily verified to prove the effectiveness of the methods. As I gained more knowledge about the

emulator creation process, the hypothesis was that I could apply the findings of the less complex research problems to more complex ideas, such as uncertainty analysis, robust optimisation and multi-objective optimisation.

Chapter 4 Emulating the building: An introduction to Kriging

4.1 Background

Kriging, also known as Gaussian process regression (GPR), has its origins in geostatistics, and is closely related to regression analysis. It named after Danie Krige, who developed the technique as part of his Master's thesis in which he applied the technique to estimating the value of gold in mines in the Witwatersrand area of South Africa [129]. The methods introduced by Krige and his colleagues have since been the subject of further development and have been applied to fields as diverse as aerodynamics [130],[131] and epidemiology [132],[133].

The Kriging method aims to provide an estimate of $f(\boldsymbol{x})$ for situations where we have not tested the input configuration $\boldsymbol{x}$. To do this, we assume that the *true* output of the simulator is a *single* realisation of that Gaussian process. However, since we do not know what this single realisation is, the Gaussian process regression provides a model for our uncertainty.

Kriging models can be viewed as multivariate Gaussian processes which can be used as meta-models for more complex simulators.

4.1.1 Basic concept and advantages

Although Kriging has a number of similarities between itself and other regression methods, it offers a number of advantages that make it well suited to analysing uncertainty in complex models.

Figure 4-1 illustrates a simple Kriging model in one dimension.

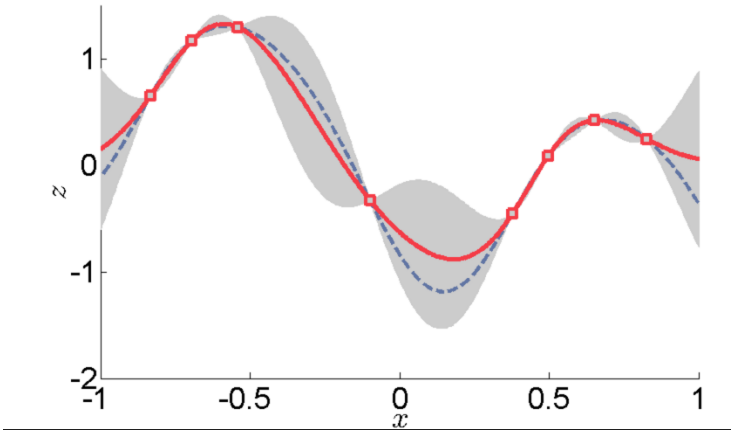


*Figure 4-1: Example of Kriging regression (**x** is the variable input and **z** is the response of the simulator)*

In Figure 4-1 above, the reader will see the following:

- the emulator training points (squares);
- the output of the simulator being modelled (red line);
- the *mean* output of the Kriging emulator (blue line); and
- the 95% confidence boundaries of the Kriging emulator (shaded grey areas).

The main advantages of Kriging over linear regression are that:

1. the simulator output is equal to the emulator output at the simulation points (this is not always possible with polynomial regression analysis);
2. the emulator always provides a prediction of its own uncertainty; and
3. the 'shape' of the emulator's mean function is not limited by order of the polynomial regression function. (as in linear regression).

Furthermore, the emulator provides a prediction of where it is performing poorly. This can be exploited in a number of ways, but can be helpful for emulator

validation, intelligent improvement of the emulator and optimisation algorithms. These concepts will be explored further in the following chapters.

4.1.2 Mathematical view of Kriging

In considering the mathematics of the Kriging emulator, I consider the univariate (i.e. one-output) simulator. In other words, I consider a simulator with one output $f(\boldsymbol{x})$ for a given input vector $\boldsymbol{x}$.

For a deterministic simulator, the output $f(\boldsymbol{x})$ will be the same for each repetition of the input value $\boldsymbol{x}$. The challenge is to build an accurate *emulator* $\hat{f}(\boldsymbol{x})$ for which $\hat{f}(\boldsymbol{x}) \approx f(\boldsymbol{x})$.

Many of the problems that computer simulators are required to solve, such as optimisation, uncertainty analyses and sensitivity analyses, often require many hundreds, or even thousands of simulation runs to obtain their desired output. However, in running such simulations, there can be a great deal of redundancy in running the simulations with similar parameters. For example, if I run two complex building energy simulations that are the same, except for a small change in the insulation thickness, I do not know the *exact* change in the annual energy use for this small change, but I know that it is *likely* to be close. Predicting what this change might be, along with the estimation of the uncertainty around our predictions, is the aim of Kriging emulation.

4.1.3 Gaussian Processes and their Bayes Linear Counterparts

I use GPs to create a probability distribution for what I think the output of the simulator might be. Although this type of function would usually be described as a *stochastic* process, I use it in this instance to allow us to model the *uncertainty*

in the output for input configurations for which I have not yet tested. In this sense, I use a Bayes linear counterpart, which characterises the output of the simulator using an *expected* value along with a *covariance* function. The mean function is the expected value of the output and covariance function is simply the covariance between two outputs:

Mean function

$$E[f(\boldsymbol{x})] \qquad \text{Eq. 3}$$

Covariance function

$$\mathrm{Cov}[f(\boldsymbol{x}), f(\boldsymbol{x}')] \qquad \text{Eq. 4}$$

I can describe the output of the emulator as a probabilistic specification of uncertainty about the simulator.

For the purposes of the investigations described in this thesis, I think of the mean function $E[f(\boldsymbol{x})]$ as a *surrogate* for the simulator and the covariance function $\mathrm{Cov}[f(\boldsymbol{x}), f(\boldsymbol{x}')]$ as an expression of confidence in the surrogate's output. (e.g. a low covariance means that I am highly confident in the output and conversely, a high covariance means that I have *low* confidence.)

4.1.4 Emulator / simulator terminology

Before the mechanisms of building a univariate emulator are described, it is worth noting some basic terminology that will be used throughout this thesis.

It is important to make the distinction between a surrogate and an emulator. In this thesis, I use the term surrogate model to refer to the *form* of the Kriging model *before* I have observed the outputs of the simulators. I reserve the term *emulator* for the probabilistic estimates of the simulator output *after* I have observed the outputs of the simulator.

4.1.5 Building an Emulator

I now describe the creation of a Bayes linear emulator based on a joint probability distribution for $f(\boldsymbol{x})$.

The GP distribution is an extension of the multivariate Normal distribution A multivariate normal distribution is similar to a traditional Normal distribution, but has a vector representing the mean and covariance values:

$$z \sim N(m_z, V_z) \qquad \text{Eq. 5}$$

Where m_z is the the mean of the i-th element of z, and the i,j-th element of V_z is the covariance between the i-th and j-th element of z. Note that z is the response of the *emulator* not the simulator. If we are considering a building model that models the annual energy use, the emulator output z is the prediction of the annual energy use for *untested* designs.

The GP model extends this idea so that m_z and and V_z become functions, rather than static distributions. These functions are then based on the input to the surrogate, $\boldsymbol{x}$ to create a Gaussian Process:

$$\hat{f}(\cdot) \sim \mathrm{GP}(m(\cdot), v(\cdot,\cdot)) \qquad \text{Eq. 6}$$

Where $v(\cdot,\cdot)$ is the covariance function and $m(\cdot)$ is the mean function.

The relationship between the Gaussian Process emulator and the simulator outputs is not immediately obvious. In equation 3, we are defining the output of the emulator as a multi-variate Normal distribution. In my research, I have typically used building models to estimate the annual energy consumption. In this case the output of the emulator will predict a normal distribution of energy uses for a given building design. It is important to understand that the output here is not *expected* to be a Normal distribution, but that the distribution expresses the *uncertainty* about what the energy of the building model might be.

For any set of training points $\boldsymbol{D} = \{x_1, x_2 \ldots x_n\}$, the output vector $f(\boldsymbol{D})$ has a mutlivariate normal distribution with a mean vector $m(\boldsymbol{D})$ and a covariance $V = \sigma^2 c(\boldsymbol{D}, \boldsymbol{D})$. (Note the $c(.,.)$ is the correlation function between elements in $\boldsymbol{D}$ – there are many different forms that the correlation function can take. For more information see sections 4.1.7 and 4.1.8)

As I have already discussed, before building the emulator, we need some knowledge about the output of the simulator for a given training set $\boldsymbol{D}$. Before I can use this information to build the emulator, I first need to establish the *prior* forms of the mean and covariance functions on which I can build the model.

4.1.6 Prior forms of the mean functions

In developing the prior mean and covariance functions, I am expressing our prior belief about what the simulator output may look like.

In the case of the mean function, I express the *form* of the function along with *hyperparameters* that modify it. For example, the form and hyperparameters for a one-input mean function could look like this:

$$\mathrm{m}(\boldsymbol{x}) = \beta_0 + \beta_1$$

In this case, the *form* of the function is linear, where β_0 and β_1 are the hyperparameters that modify it.

The hyperparameters of the mean function are determined based on the results of the simulator's training set. However, I will describe this in more detail in the next section.

Most commonly, the mean function takes the form of a linear model:

$$m(x) = h(x)^T \beta \qquad \text{Eq. 7}$$

The elements in $h(.)$ are referred to as the *basis functions* where β is a vector a q hyperparameters, where q is the number of dimensions plus one, such that:

$$h(x) = \left(1, x_1, x_2, \ldots, x_p\right)^T \qquad \text{Eq. 8}$$

Note that q is normally $1 + p$, where p is the number of input dimensions to the simulator. Note also that there are many other forms of the mean function, but these are not used within this thesis. For more information the reader is referred to Rasmussen and Williams [134].

4.1.7 Prior covariance function

The second element in the GP function is the covariance function $v(.,.)$. The two elements of the covariance function are the *prior variance* σ^2 and the *correlation function* $c(.,.)$. The covariance function is most commonly written as:

$$v(x, x') = \sigma^2 \, c(x, x') \qquad \text{Eq. 9}$$

This function formulation assumes that the prior variance σ is constant (since $c(x, x) = 1$ for all x).

4.1.8 Correlation functions

It is also common to assume *stationarity* of the correlation function, which means that I base it on the difference between the two inputs (i.e. $x - x'$). In order to simplify the understanding of the correlation function, I first assume that I can formulate the correlation function based on a measure d that is modified by the distance between two points. I define d as:

$$d = |(x - x')\Delta\,(x - x')|^{\frac{1}{2}} \qquad \text{Eq. 10}$$

In this function I have a matrix Δ, which is a diagonal matrix of smoothness coefficients δ_i. These smoothness coefficients regulate the magnitude of the variance in their respective dimensions (there is a smoothness coefficient for each dimension). It's important to recognise that these δ_i values cannot be determined analytically - they have to be determined by trial and error. The smoothness coefficients for each dimension i are given by δ_i, such that

$$\Delta_{ii} = \delta_i \qquad \text{Eq. 11}$$

I can then define the correlation function $r(\cdot)$ as some function of d, such that:

$$c(x, x') = r(d) \qquad \text{Eq. 12}$$

The following functions are commonly used as the correlation function in Kriging:

- Exponential function: $(\text{for } \delta_0 < 2)\ \exp(-d^{\delta_0})$
- Gaussian: $\exp(-d^2)$
- Matérn ($\delta_0 = {}^3/_2$): $\left(1 + \sqrt{3d}\right)\exp(-\sqrt{3}d)$
- Matérn ($\delta_0 = {}^5/_2$): $\left(1 + \sqrt{5d} + {}^5/_2\, d^2\right)\exp(-\sqrt{5}d)$

The Matérn function is based on the generalised form:

$$\frac{2^{1-\delta_0}}{\Gamma(\delta_0)}(\sqrt{2\delta_0}d)^{\delta_0}K_{\delta_0}(\sqrt{2\delta_0}d)$$

Where $\Gamma()$ is the *gamma function*, and $K_{\delta_0}$is the *modified Bessel function of the second kind* [135].

4.1.9 Posterior mean and covariance functions and the hyperparameters

So far I have defined the form of the mean function and covariance function, along with the options for its associated correlation function $c(.,.)$.

However, the most challenging process in building the Kriging emulators is the estimation of the *hyperparameters*. The choice of these hyperparameters defines

the emulator, however, they cannot be derived directly from the training set, but must be *estimated*. The hyperparameters cannot be analytically derived from the training set, I must determine their *most likely* values. To do this, I use prior probability distributions for the hyperparameters. From these defined prior distributions, I can calculate their posterior probability functions based on the training data. I then choose the 'most likely' value for each of the hyperparameters to finalise the design of the emulator.

The hyperparameters that need to be estimated are:

- $\hat{\beta}$ - the vector that defines the mean function
- $\hat{\delta}$- the vector the defines the *smoothness* of the emulator in each dimension
- $\hat{\sigma}$- the scalar that defines the prior variance of the correlation function.

We denote the *posterior* values for β, δ and σ with a 'hat'.

In order to estimate the hyperparameters, it is assumed that the *prior* distribution of the hyperparameters has the following form:

$$\pi(\beta, \sigma^2, \delta) \propto \sigma^{-2}\pi_{\delta}(\delta) \qquad \text{Eq. 13}$$

where $\pi(.)$ is the *likelihood* function. This function determines the probability that any combination of β, σ^2 and δ represents the true output of the simulator (based on the assumption that the simulator is Gaussian process). Therefore, in order to find about what values β, σ^2 and δ should take, the function $\pi(\beta, \sigma^2, \delta)$ should be *maximised*.

Each of the hyperparameters is dependent on δ. This means that, once the most likely values of δ are known ($\hat{\delta}$), we can derive the values for $\hat{\beta}$ and $\hat{\sigma}$. These

most-likely parameters are derived be maximising the log-likelihood function, which is based on the likelihood function in equation 10:

$$\ln(\pi_\tau^*(\tau)) = -0.5(n-q)\ln(\hat{\sigma}^2) - 0.5\ln(|A|) - 0.5\ln(|H^T A^{-1} H|) \quad \text{Eq. 14}$$

Where:

$$H = h(\boldsymbol{D}^T)^T \quad \text{Eq. 15}$$

$$A = c(\boldsymbol{D}, \boldsymbol{D}) \quad \text{Eq. 16}$$

$$\hat{\sigma}^2 = \frac{\left(f(D) - H\hat{\beta}\right)^T A^{-1}(f(D) - H\hat{\beta})}{n-q-2} \quad \text{Eq. 17}$$

The full derivation of this log-likelihood function is too long to be detailed in full here. Interested readers are directed towards Conti et al's paper: '*Gaussian process emulation of dynamic computer codes'* [136]

Based on A and the training set results $f(\boldsymbol{D})$, we can define the *posterior* mean and covariance functions which form the basis for 'trained' emulator:

Posterior mean function

$$m^*(x) = h(x)^T\hat{\beta} + t(x)^T A^{-1}(f(D) - H\hat{\beta}) \quad \text{Eq. 18}$$

Where $t(\boldsymbol{x}) = c(\boldsymbol{D}, \boldsymbol{x})$ Eq. 19

Posterior covariance function

$$\begin{aligned} &v^*(x, x') \\ &= \hat{\sigma}^2\{c(x, x') - t(x)^T A^{-1} t(x') + (h(x)^T \\ &- t(x)^T A^{-1} H)(H^T A^{-1} H)^{-1}(h(x')^T - t(x')^T A^{-1} H)^T\} \end{aligned} \quad \text{Eq. 20}$$

One method for maximising the log-likelihood function is to 'try out' different values of δ. For problems with a small number of dimensions this is relatively trivial, since the computation cost of trying out most of the values of δ is relatively low. However, for higher dimensional problems, the curse of the dimensions is encountered again. To overcome this, we use GA-based optimisation. In the *DiceKriging* package for *R* (which is used in chapters 3-6 of this thesis), for example, the derivatives-based GA package *rgenoud* is used [137], [138].

4.2 Proof of concept for the creation of a Kriging emulator for a building design problem

As I have shown, building energy simulations and subsequent optimisation procedures are a computationally expensive process. I therefore first test a Kriging-based regression model of a building.

In this section, I present an emulator of a complex building model which I create using a Kriging model [139]. The model is created using a four-step process:

1. Run a set number of training simulations (typically generated using a Latin hypercube design – it is important to provide good coverage of the input space).
2. Use the results of the training simulations to create an emulator.
3. Validate the emulator by completing additional training runs of the simulator and comparing these to the output of the simulator.
4. Rebuild the emulator using the validation runs and use it to make predictions about the output of the simulator.

To avoid confusion, the building model is always referred to as the *simulator* and the Kriging meta-model is always referred to as the *emulator*.

4.2.1 The building simulator

The building simulator is based on a three-zoned thermal network model of a two-storey building with a roof space. The building has a simple form and is glazed to one of the façades (Figure 4-2).

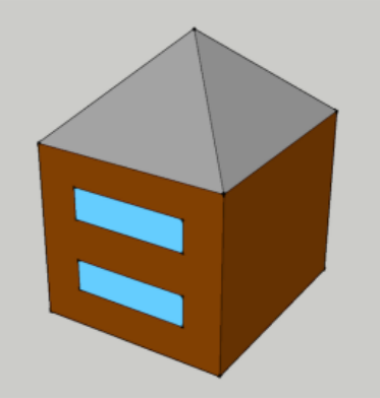


Figure 4-2: Form of the building used in the simulator

There are many approaches to creating thermal models of buildings, but for the purposes of this experiment, I use a thermal network model. Such models have been extensively used to solve such thermal modelling problems in the past as the calculations are completed analytically rather than numerically.

RC models allow any building to be represented by an analogy with an electrical model. The circuit is then ‘solved’ as a series of first order differential equations. The system consists of a series of nodes, which are coupled by conduction and convection and the thermal storage is represented by a capacitor. The heat flow by radiation is represented by a current source on the node where the radiation is incident and the node voltage represents its temperature. Each layer of the construction contributes to the dynamic response of the building and is modelled

as an independent element when creating the thermal model. The nodes and boundary conditions of the thermal elements are determined from the diffusivity of the material and the time step of the dynamic model [140].

4.2.2 About the building

Each of the building's stories is a heated zoned of dimensions 8 m × 8 m × 2.4 m and has a constant air exchange rate of 0.2 air changes per hour. The roof is pitched at an angle of 30º with no overhang. The roof space is unheated and has a constant air exchange rate of three air changes per hour. For simplicity, at each time step, the heating and cooling loads are calculated for each heating zone to maintain the internal temperature at 21 ºC using an ideal plant with an unlimited capacity. The thermal properties for the materials used for the construction for each surface are shown in Table 4-1 and the glazing system is shown in Table 4-2. The building is located in Plymouth and the standard CIBSE Test reference year is used [141]. No other internal gains are considered.

Table 4-1: Properties of constructions used in the building model

Surface / material	Thickness / mm	Thermal conductivity / $Wm^{-1}K^{-1}$	Density / kgm^{-3}	Heat Capacity / $Jkg^{-1}K^{-1}$
External Wall				
Cast concrete	220	1.13	2000	1000
Insulation	10-100	0.038	25	1030
Cast concrete	220	1.13	2000	1000
Ground floor				
London Clay	750	1.41	1900	1000
Brick	220	0.77	1750	1000
Concrete	100	1.13	2000	1000
Insulation	80	0.025	30	1400
Chipboard	25	0.15	800	2093
Carpet	10	0.06	160	2500
Internal Floor/ceiling				
Carpet	10	0.06	160	2500
Chipboard	25	0.15	800	2093
Cavity	100	-	-	-
Insulation	10	0.04	15	1300
Plaster Board	13	0.16	600	1000
External Roof				
Clay tile	5	0.84	1900	800
Glass fibre	25	0.04	12	840
Roofing felt	5	0.19	960	837
Insulation	180	0.043	12	840
Plaster Board	13	0.16	600	1000

Table 4-2: Properties of the glazing used in the building model (these values are taken from typical argon filled 4-16-4 double glazing)

Glazing Layer	Glass 1	Argon Cavity	Glass 2
Thickness / mm	4	16	4
Thermal conductivity / $Wm^{-1}K^{-1}$	1.06	0.016	1.06
Transmittance	0.899	-	0.672
Forward reflectance	0.08	-	0.188
Backward reflectance	0.08	-	0.163
External emissivity	0.837	-	0.059
Internal emissivity	0.837	-	0.837

The building model created allows the glazed area, insulation thickness (of the external wall only) and orientation of the building to be varied. If we consider the emulator as a function, $f(\boldsymbol{x})$, then the elements in the input vector $\boldsymbol{x}$ are:

- Percentage glazed area (50 - 90%)
- The external wall insulation thickness (10 - 100 mm)
- Orientation of the glazed surface (-90 to +90 degrees)

Note that each of the inputs are continuous and the objective function $f(\boldsymbol{x})$, refers to the annual building energy use (in kWh). Continuous input variables are used in this instance to test the Kriging emulator's ability to respond to a complex search surface.

4.2.3 Emulator variables

The variables create a three-dimensional input space. Previous researchers have shown that the number of training simulations required to create a good balance between the number of training simulations required and the quality of the emulation is 10 times the number of emulator dimensions (i.e. input dimensions) [14]. Since I have 3 input dimensions, I train the emulator using 30 training simulations.

4.2.4 Selection of the training data

Previous research has shown that, when training an emulator of an unknown response, Latin Hypercube sampling provides good efficiency in capturing the response of the simulator. In order to ensure that training points are not too

correlated (i.e. too close to each other), I employed a *maxi-min* Latin Hypercube design in which I generated 100 samples and selected the LH with the maximum minimum distance between training points. The samples were then used to train the emulator.

4.2.5 Validating the emulator

I validate the emulator by running additional training simulations and compare the output of the emulator to those points that I have already simulated. I compare these formally using the following methods.

4.2.6 Individual standardised errors

The individual standardised errors are perhaps the easiest way of providing a measure of the accuracy of the emulator. The standardised errors take account of the predicted mean and variance of the emulator and are calculated as part of each additional training simulation. The standardised errors are signified by e_j and are calculated as follows:

$$e_j = \frac{f(x_j') - m(x_j')}{\sqrt{v(x_j', x_j')}} \qquad \text{Eq. 21}$$

where e_j are the individual standardized errors, x_j' are the individual simulator inputs for the validation samples, and j is the index of each sample (where $j = 1,2,3 \ldots n'$).

Typically, the number of validation samples required should be less than the number of simulations undertaken in the training set (otherwise, this would negate the need for a training set in the first place). However, for the purposes of

research, I ran an additional 30 validation samples to check the performance of the emulator.

The standardised errors help to show how well the building emulator is performing. Since the emulator provides both a mean (the expected value) and variance for each input configuration, I know that, if the emulator fits the output well, I would expect the standardised errors to have a Normal distribution with a mean of 0 and a standard deviation of 1. Therefore, if the emulator is accurate, then 95% of the standardised errors would be within 2 standard deviations of the mean. The results of the validation showed that, for the 30 samples taken, 93% of the standardised errors lie within 2 standard deviations. This result demonstrated that the emulator is providing an appropriate level of confidence (for a normal distribution, 95% of the residuals would be within 2 standard deviations of the mean). The results of the standardised errors are shown in Figure 4-3.

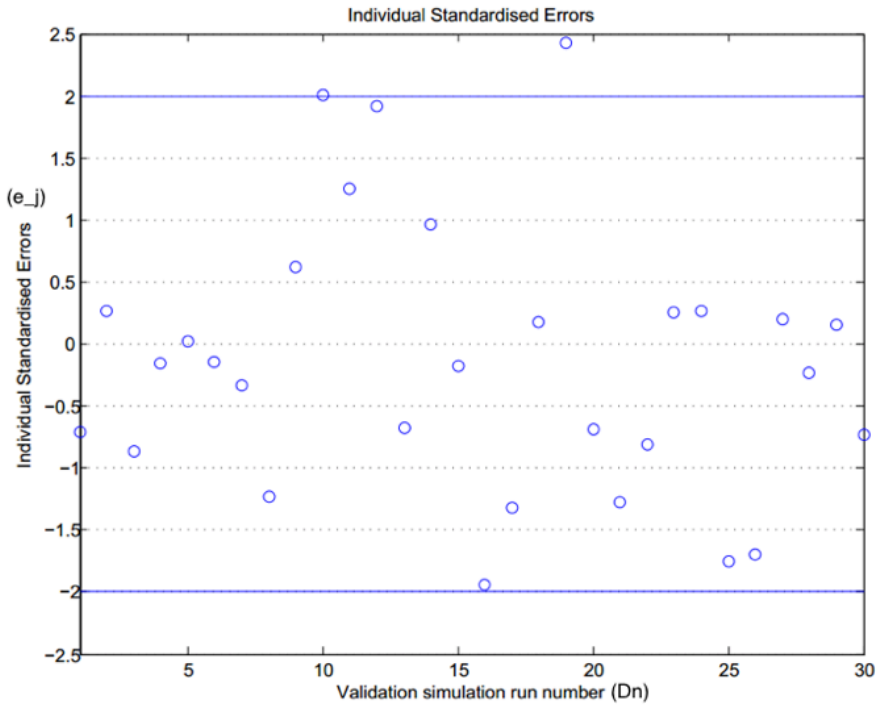


Figure 4-3: Plot of the standardised errors of the validation points

4.2.7 Calculating the Mahalanobis distance

The Mahalanobis distance M is a scalar value that provides an alternative means of validating the emulator [142]. The validation simulations and the associated mean function and variance from the emulator are used to calculate M:

$$M = (f(D') - m^*(D'))^T (v^*(D', D))^{-1} (f(D') - m^*(D)) \qquad \text{Eq. 22}$$

For a valid emulator the M will have a scaled F-distribution with degrees of freedom given by n' and (n-q), with an expected value of $\frac{(n-q)}{(n(n-q-2))}$. Therefore, the expected value of M is 30 ($E(M)$) and variance 147.3 ($\mathrm{var}(M)$).

The Mahalanobis distance for the validation samples was 34.0, which falls within the expected distribution of M for a valid emulator. The probability density curve

in Figure 4-4 is the expected distribution of the Mahalanobis distance for a valid emulator. By calculating the Mahalanobis distance for our validation samples, we are effectively taking one sample from the distribution in Figure 4-4. Therefore, if the sample represents a plausible draw from this probability distribution, then it is *more likely* that the emulator is valid.

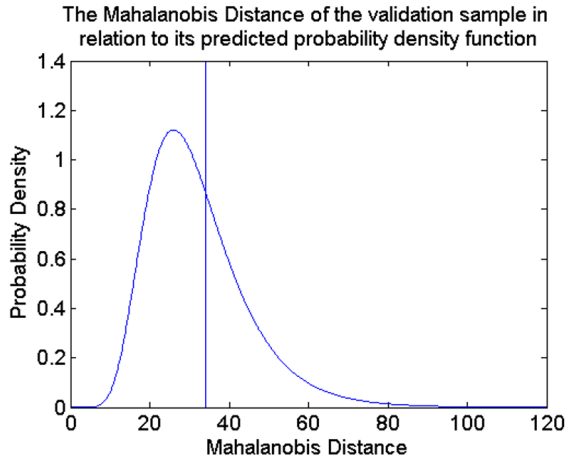


Figure 4-4: Plot of the Mahalanobis distance vs. its expected probability distribution

4.2.8 Brute force validation

To further demonstrate the validity of the building emulator, a 9×9×9 grid of equally spaced input points ($\boldsymbol{D_{729}}$) was simulated. The results of these input points ($f(\boldsymbol{D_{729}})$) were then compared to the results of the emulator. Figure 4-5 shows a histogram of the percentage errors in the emulation over the 729 points. Figure 4-6 shows the location of emulation inputs where the simulation output is more than two standard deviations from the mean function (as predicted by the covariance function). Figure 4-5 shows that the emulator tracks the simulator reasonably accurately over the ranges shown. The histogram in Figure 4-5 is also close to a Normal distribution. For a valid emulator, we expect that the standardised errors will normally distributed about zero mean.

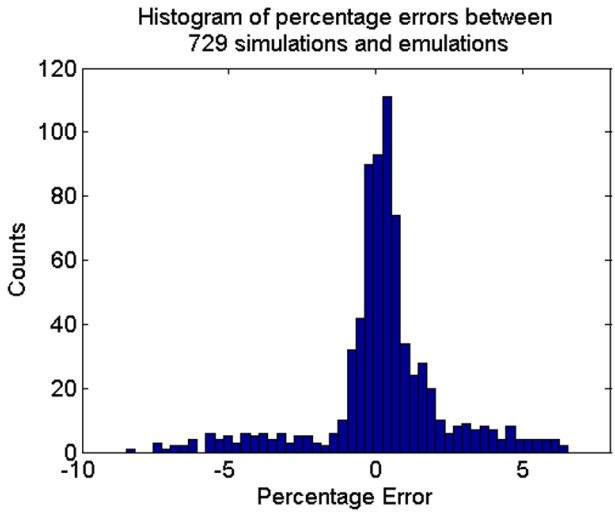


Figure 4-5: Histogram of percentage errors of the simulator

The results shown in Figure 4-6 show the points where the error in the emulator is most significant. Most of these points are at the 'edges' of the input space, suggesting poorer emulation at these points. However, these points represent only 17% of the total input. It is expected that around 5% of such outputs would be greater than two standard deviations from the true value given the probabilistic nature of the emulator. With 17% instead of 5% beyond two standard deviations, this would suggest that the emulator is a little over confident (i.e. the covariance function is underestimating the variance at these points).

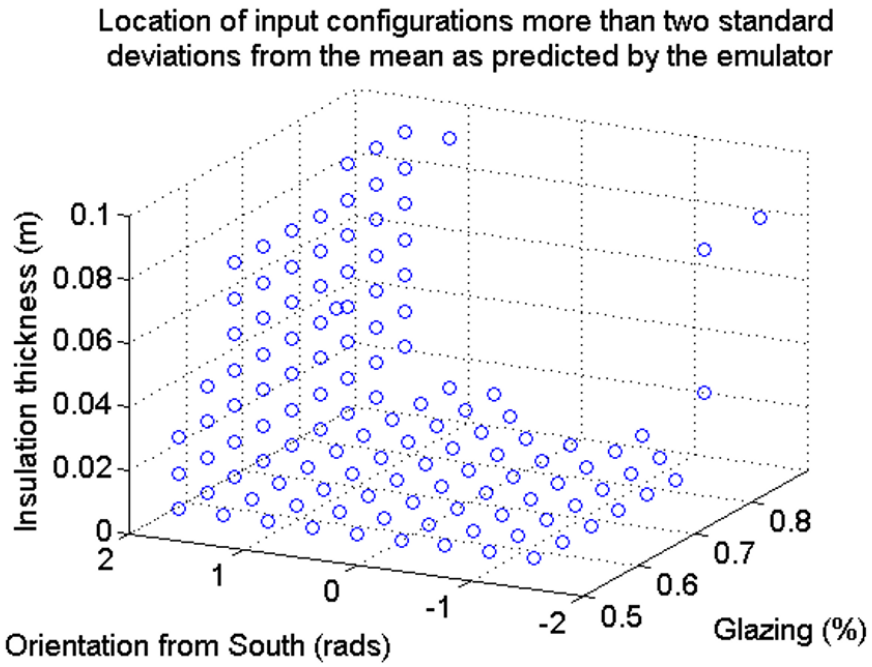


Figure 4-6: Location of points that are greater than 2 standard deviations from the predicted mean function (as predicted by the covariance of the emulator)

4.2.9 Discussion of the results

This paper has demonstrated that a Kriging emulator can be built to predict the modelled energy use of a simple building model within a reasonable range of accuracy. Based on a training set of 30 simulations, the validation diagnostics (individual standardised errors and the Mahalanobis distance) indicated that the emulator's mean and variance predictions were likely to be correct. Further interrogation of the emulator by using a brute-force method showed that 78% percent of 729 emulators mean function values were within 2% of the true simulator values.

Although the validation diagnostics are clearly in line with what would be expected from a valid emulator, there are clearly areas of the emulator that can be improved (as indicated by the brute force analysis). Such improvements are likely to come from improved estimation of the hyperparameters.

One of the major advantages in creating the emulator is to be able to identify key trends and patterns in the output of the simulator with a limited number of simulator runs. It has been shown that trends in the output of the emulator closely match those of the simulator. I found that;

- the orientation of the building is not as important as might have been expected in determining the total heating and cooling load of the building (i.e. there is not much variation observed in the orientation axis).
- an increase in glazed area appears to lead to a linear increase in the total amount of energy used (combined heating and cooling); and
- increasing the amount of insulation used in the building provides a very steep reduction in the amount of energy used from around 10 mm to 40 mm, but between 40 mm and 100 mm the increasing thickness of the insulation appears to have less effect. This indicates that, above a baseline level of insulation (around 40 mm), increasing the amount of insulation has less and less effect per mm of insulation added.

The Kriging methodology allows a variety of analyses, which are usually too computationally expensive to perform with the building simulator. These analyses include applying MC analysis to the emulator mean and MC uncertainty analysis (i.e. assessing the effect of uncertainties in input values on the output) as well as sensitivity analyses [143] and model calibration [144].

I shall revisit the subject of MC analysis later in this thesis. In the following section, I demonstrate a real world application of this method, with the aim to determine the potential effect of new UK test reference years (TRYs) on building design.

4.3 Testing the effect of new test reference years using Kriging[5]

Average weather years are used all over the world to test the thermal performance of buildings. However, as the climate changes, the 'average' weather will change too, and this needs to be reflected in the weather files that we use for building design.

4.3.1 Test references years in the UK

In the UK, Test Reference Years (TRYs) have been used by most building designers since 2006.

The data which makes up each of the UK's most recent TRY files is taken from 14 locations across the UK and uses data from that has been logged by weather stations between 1983 and 2004. However, rather than use an entire year's worth of data from each weather station, each TRY is made up of a series of average months from different years, with the idea being to create an 'average' weather year.

It has recently been proposed that the UK updates its new TRY files to take account of more recent data from as recent as 2013.

[5] This section represents the combined work of Dr Matthew Eames and the author.

4.3.2 Choosing the ‘most average’ weather files

There are a variety of different methods for choosing the ‘most average’ weather year. However, the most common are the Scandia method [145] and the ISO method [146], with the Scandia method being the most complex.

In the Scania method, each day is characterised by nine primary indices. These are;

- the maximum, minimum and mean dry bulb temperatures;
- the dew point temperature;
- the mean radiant temperature;
- the maximum and minimum wind velocity (this can affect both the ventilation and infiltration rate of the building, as well as increasing the convective heat loss); and
- the total global horizontal solar radiation.

In contrast, the ISO method only considers;

- the mean dry bulb temperature;
- the total global horizontal solar radiation; and
- the mean relative humidity;
- the wind speed (but only considered as a secondary variable)

The previous UK method for creating TRYs was similar to the ISO method, but with a slight modification, making mean wind speed a primary index instead of relative humidity.

Previous studies have shown that the existing UK TRYs provide a good representation of their baseline data [147]–[149]. However, they are only based on data collected up to 2004 and therefore they do not take into account the more recent changes in the climate. The full method detailing this creation of the updated TRYs can be found in Eames et al. [150].

4.3.3 Implications for the new TRYs on buildings

The months that make up the 14 locations that are currently available and new UK test reference years are shown in Table 4-3.

Table 4-3: Make-up of the current and updated TRYs for the UK

	1	2	3	4	5	6	7	8	9	10	11	12
Belfast												
Original	2003	1985	1993	1998	1997	1997	2001	1999	2001	1988	1989	1985
Update	2000	2005	1993	1995	1988	2000	2008	1996	1997	1988	1984	2012
Birmingham												
Original	2000	2004	2004	2000	1995	1983	2001	1996	1995	1988	1991	2000
Update	2003	2005	2004	2006	1988	1984	2010	1996	1995	1988	2007	2007
Cardiff												
Original	1988	2003	1993	1988	2000	1983	1996	1996	1996	1988	1995	1983
Update	1986	2005	1993	2006	1988	1986	1997	1991	2010	2002	2008	2007
Edinburgh												
Original	1988	1982	1981	1985	1997	1999	1996	1980	1990	1988	1998	1979
Update	2003	2005	2004	2010	2013	1993	1987	2007	2013	2010	2008	1984
Glasgow												
Original	1986	1985	1978	1998	1997	1979	1996	1998	1997	1988	1998	1984
Update	1988	1999	2008	1988	1988	1998	1997	2005	2010	2010	1998	1996
Leeds												
Original	1995	1993	1993	1996	1997	2001	2001	1994	1995	1991	1990	1985
Update	1995	2005	2010	1995	2003	1993	2005	2013	2013	2000	1991	2007
London												
Original	1988	2004	2004	1992	2000	2001	1991	1996	1987	1988	1992	2003
Update	2011	2001	2004	1988	2004	1994	2005	2000	2007	2009	1991	2003
Manchester												
Original	1999	1992	2004	2000	1985	2001	1996	1996	1996	1986	1987	1987
Update	1999	2004	2001	1988	1985	1984	1996	1998	1989	1988	2007	1991
Newcastle												
Original	1988	1999	1992	1998	1997	2000	1996	1998	1996	1985	1989	1984
Update	1992	2001	1988	1998	1985	1998	1987	1984	1985	1988	1987	1984
Norwich												
Original	2004	1999	2004	1995	1993	1990	2002	1996	1985	1987	2001	1998
Update	2000	2005	2004	2005	2003	2005	2001	2012	2007	2002	2012	2003
Nottingham												
Original	1995	1999	1993	1998	2003	1984	2001	1994	1987	1999	1987	1994
Update	2003	2005	2004	1999	1988	2000	2008	2007	2007	1988	1990	2012
Plymouth												
Original	2004	1999	2001	2004	2000	2000	1994	1996	1988	1983	1984	1983
Update	1994	1999	2005	2006	2012	1994	1994	2000	2007	1986	2001	2003
Southampton												
Original	1982	1999	1983	1988	1985	1995	1981	1987	1988	1987	1987	1982
Update	2013	2004	2004	2008	1997	2013	1985	2000	1995	2002	2012	1997
Swindon												
Original	1988	1999	1993	2000	2000	1988	1996	1996	1996	2002	1987	1983
Update	2003	2005	2004	1995	1993	2008	2005	1987	1987	1985	2001	2007

The main driver for heating and cooling energy use is temperature. Figure 4-7 shows the difference between the mean temperature for the old and the updated TRYs. A comparison of the cooling degree days is given in Figure 4-8

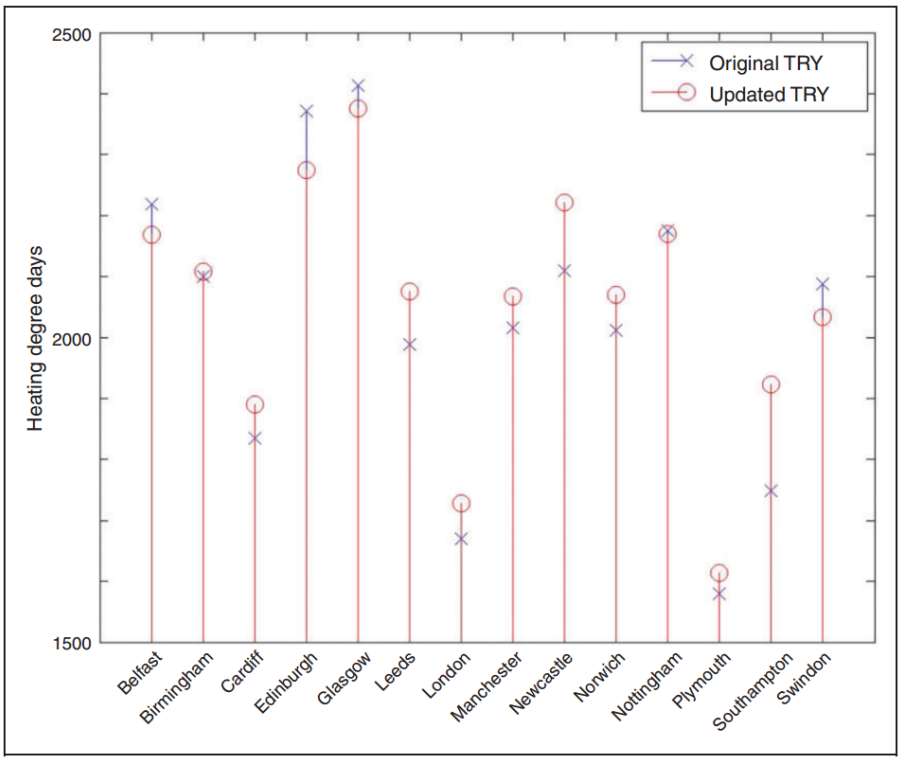


Figure 4-7: A comparison between the heating degree days at each location for the current and updated TRYs

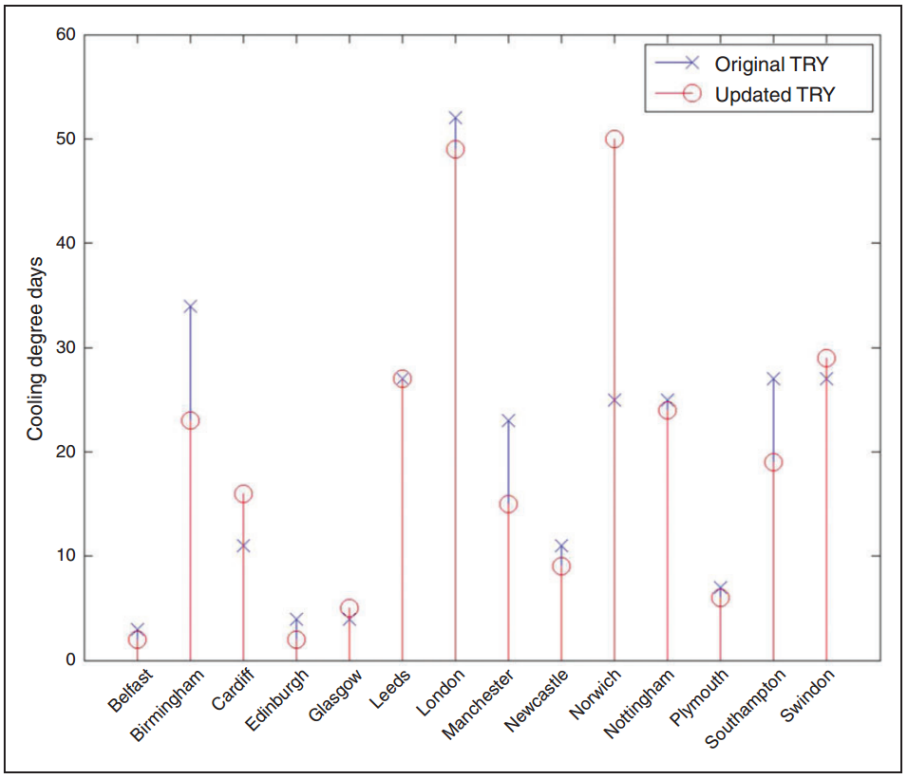


Figure 4-8: A comparison between the cooling degree days at each location for the current and updated TRYs

It is clear from the changes in the heating and cooling degree days that the new TRYs are likely to influence the building's design. However, due to the large number of different building types and locations, analysing the effect of the TRYs for all potential combinations of parameters would be computationally intractable – there are simply too building combinations to simulate within a reasonable timeframe. To solve this, we used a Kriging meta-model instead of the building simulator. This reduced the computation required by many orders of magnitude, allowing us to examine the effect of the new weather files on each building's heating and cooling requirements.

4.3.4 Kriging meta-model

The Kriging meta-model used in this analysis is based on a 5-input building simulator (which was determined by a sensitivity analysis). Since we are primarily

interested in the effects of the TRYs on heating and cooling, we chose five parameters that would have the most effect. To study the effect of the changes to these parameters, we randomly assigned 10,000 building configurations from the ranges shown in Table 4-4[6].

Table 4-4: Range of variation of the building parameters (note that each variables in continous)

Building parameter	Parameter minimum	Parameter maximum
Aspect ratio	0.33	3
Wall U Value ($Wm^{-2}K^{-1}$)	0.05	0.5
Roof U Value ($Wm^{-2}K^{-1}$)	0.05	0.5
Infiltration (ACh-1)	0.05	0.5
Glazing percentage	10	60

4.3.5 The building model and emulator

The building models use a standard cavity brick wall construction which is common to the UK [151]. We modelled the thermal performance of the building in *EnergyPlus* [22]. Table 4-5 shows the properties of the materials used in the modelling.

[6] Note that in this case the distribution of the variables was assumed to be flat

Table 4-5: Properties of the building constructions

	Material Thickness (mm)	Conductivity (W/m.K)	Density (kg/m3)	Heat Capacity (J/K)
External Wall				
Brick	106	0.89	1920	790
Insulation	36-586	0.03	43	1210
Brick	106	0.89	1920	790
Plasterboard	12.5	0.21	700	1000
Ground Floor				
Insulation	110	0.025	700	1000
Concrete	100	2.3	2300	1000
Cavity	100	-	-	
Chipboard	20	0.13	500	1600
Carpet	10	0.04	160	1360
External Roof				
Clay Tile	12.7	0.84	1900	800
Membrane	0.1	1	1100	1000
Insulation	69-594	0.03	43	1210
Plasterboard	12.5	0.21	700	1000
Internal Walls				
Plasterboard	12.5	0.21	700	1000
Brick	0.005	0.89	1920	720
Plasterboard	12.5	0.21	700	1000

The cavity brick construction was chosen as it is the most commonly used construction for houses in the UK. This is also true for the roof construction (clay tile, membrane and insulation).

For each training simulation, we calculated the heating and cooling load as well as the total exergy[7].

We took 75 samples from the simulator and used these to train the emulator. This equated to 2,100 training simulations (75 training simulations for each TRY, old and new, for each of the 14 locations considered). The parameters for these

[7] In this case, the cooling load could be met with natural ventilation, but this is not the focus of the work; we only use the cooling load as a proxy for *how much* cooling the building requires, not to size a real cooling system.

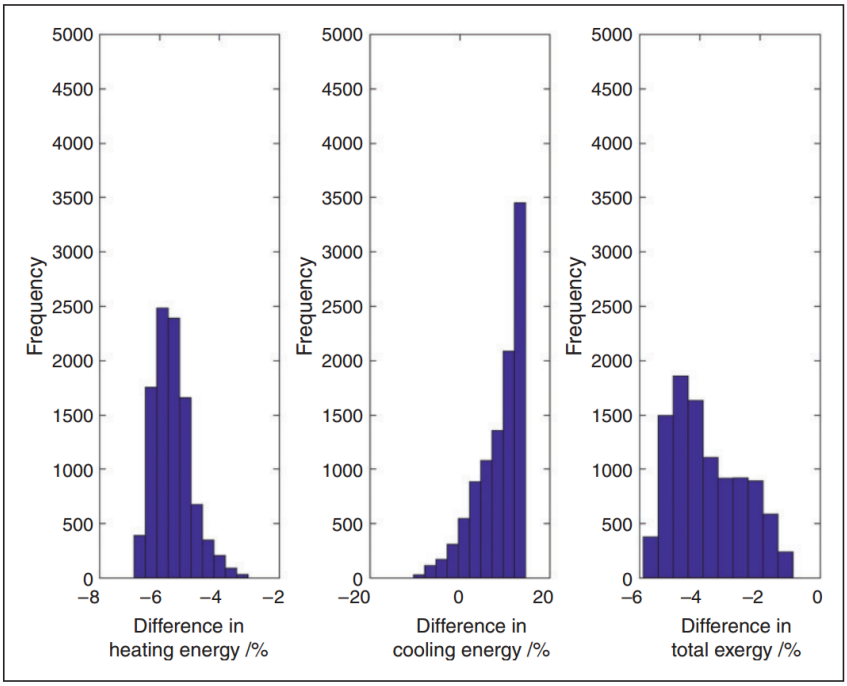
simulations were chosen using a *maximin* Latin Hypercube design method [152] and the emulator was built using the *R* package *DiceKriging* [137].

Once the emulator was trained, we conducted an MC analysis on the 10,000 buildings by varying the samples detailed in Table 4-4. The random buildings were samples using a uniform distribution (this would ensure that the entire parameter space within the building design is covered).

4.3.6 Results

In analysing the results, we are most interested in determining how the new TRYs change the thermal load characteristics of each building. We therefore present the results as a series of histograms showing the difference in total heating and cooling loads, as well as the total exergy for each location. The histograms for each location are shown in Figure 4-9 and Figure 4-10.

Edinburgh



London

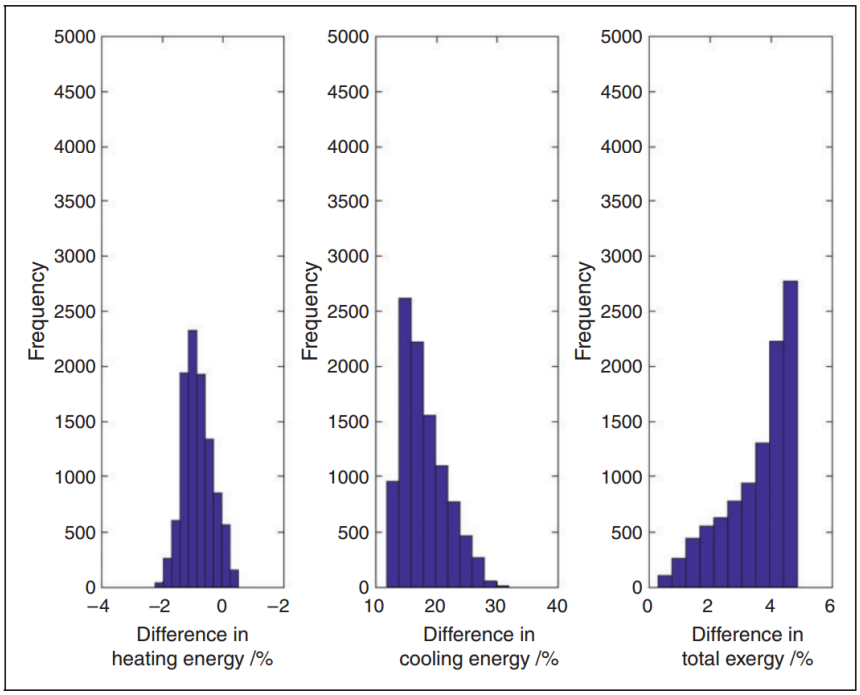
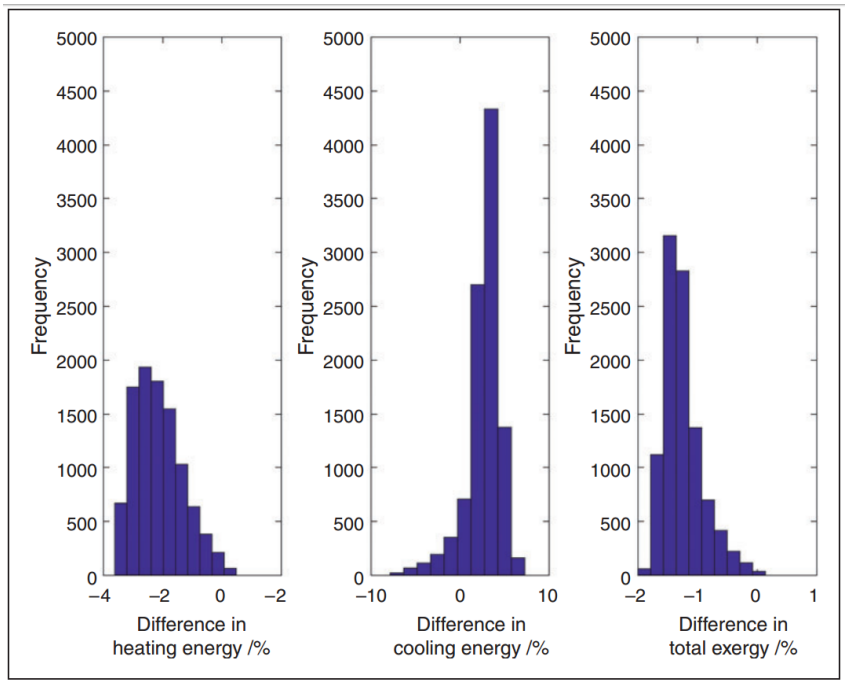


Figure 4-9: Histograms of the difference between the heating and cooling energies of the buildings as well as the change in total exergy (Edinburgh and London)

Manchester



Plymouth

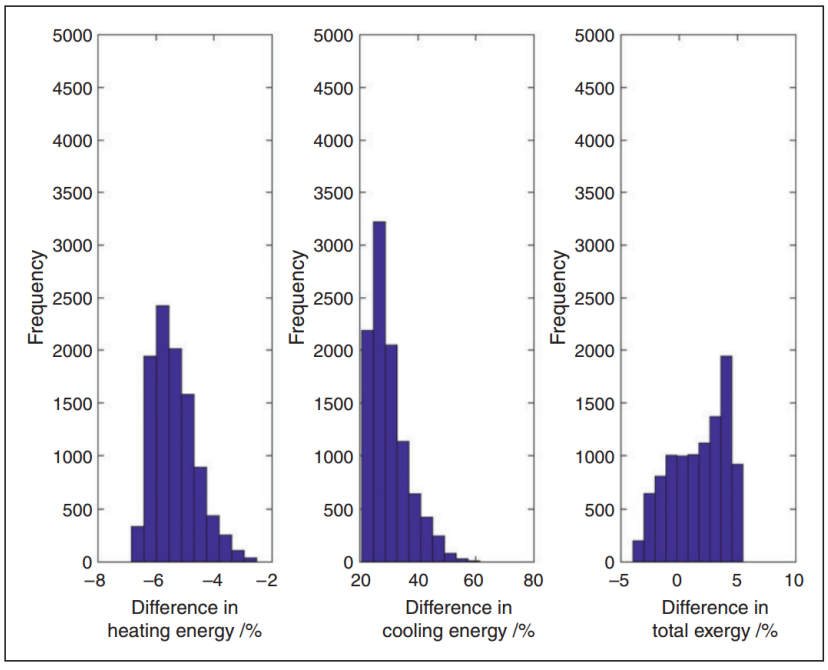


Figure 4-10: Histograms of the difference between the heating and cooling energies of the buildings as well as the change in total exergy (Manchester and Plymouth)

In analysing the results of the modelling, we also investigated the relationship between each building's thermal insulation levels and its effect on energy use. The results of this analysis are shown in Figure 4-11.

These results are based on the heating energy, cooling energy and total exergy for all 10,000 building configurations modelled using the new London TRY and are plotted against each building's total heat transfer coefficient. The heat transfer coefficient is calculated as the sum of the fabric losses and ventilation losses. The fabric losses are calculated as follows:

$$\mathrm{h} = \sum_{i=1}^{n} A_i U_i \qquad \text{Eq. 23}$$

Where h is the heat transfer coefficient, n is the number of building surfaces, A_i is the areas of surface i, and U_i is the U-value of surface i.

As before, the plant in each building is used to maintain the internal temperature at 21°C across the year.

In the analysis of the 10,000 building configurations, the results show that a building which has a lower heat transfer coefficient generally uses less energy, whereas buildings that have a higher heat transfer coefficient generally use more energy. The 50 buildings which use the least total exergy are shown by the black circles on the plot.

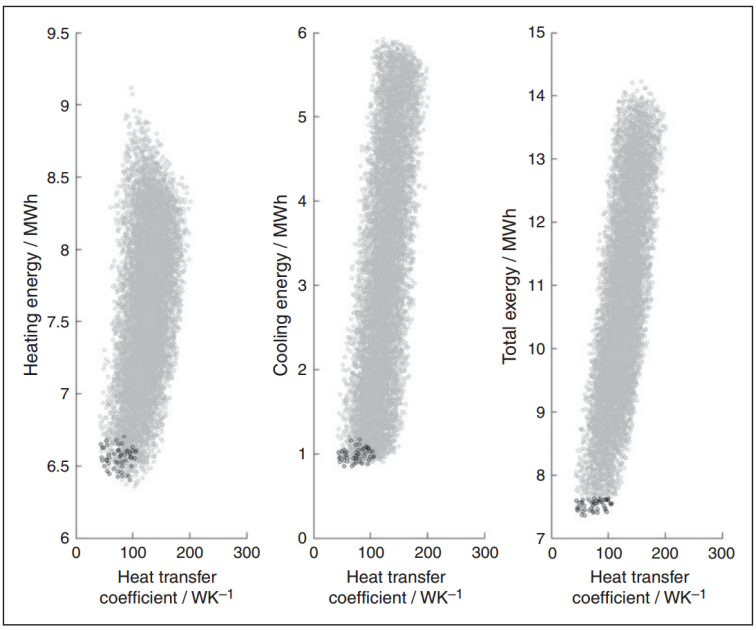


Figure 4-11: Energy use vs. heat transfer coefficient - London

4.4 Discussion

As shown in the figures, the new TRYs predict different heating and cooling energy uses for Manchester, Edinburgh, Plymouth and London.

In the case of Edinburgh, the buildings modelled appear to use less exergy overall. This is because most of the 10,000 buildings modelled tend to use less heating energy, but more cooling energy (remembering that cooling energy is used here as a proxy for overheating). This indicates that, using the new TRYs, it is more likely that a building being modelled will result in overheating. This is also reflecting the fact that there are fewer heating degree days.

The updated London TRY shows that we can expect that the cooling energy required to maintain each building to 21°C will increase. For London, the cooling energy dominates the total energy use. The cooling energy and the total exergy are highly correlated (the correlation coefficient $R^2 = 0.85$). The distribution of the heating load difference is relatively small and is between -2% and +1%.

For Manchester most building configurations are predicted to use less heating energy and less energy in total. The change in total exergy is correlated to both the change in heating energy *and* the change in cooling energy. However, the percentage change in the cooling energy used by the buildings is much higher than the percentage change in heating energy:

- cooling energy change (-6 to 7%)
- heating energy change (-4 to 1%)

Since the cooling energy was much smaller (in absolute terms) than the heating energy to begin with, this difference is reflected in the total exergy[8].

The differences for the heating, cooling and total exergy for Plymouth show that the total heating load is expected to increase for all building configurations. The cooling load is also expected to increase by a similar magnitude. The change in exergy is roughly evenly distributed around 0 kWh, however this is more correlated to the cooling load than the heating load.

For all locations, the absolute change in heating energy and total exergy is small (less than 10%).

The dominant energy source for both the original and the new weather files is the heating energy. In the results, we have observed large *percentage* changes in the cooling energy, of up to 100%. However, compared to the heating energy, the *absolute* change in cooling energy is low.

---

[8] Note that the wind in the simulator is modelled uses equations that are empirically based. Full details of these are given in the supporting documentation of EnergyPlus [166]

The results can be summarised as follows:

- most buildings use less heating energy
- most buildings use more cooling energy (i.e. overheat more)
- the total exergy is highly correlated to the heating *energy* for eight locations (Belfast, Birmingham, Edinburgh, Glasgow, Manchester, Newcastle, Nottingham and Swindon)
- the total exergy is highly correlated to the heating *exergy* for eight locations (Cardiff, Glasgow, Leeds, London, Manchester, Norwich, Plymouth and Southampton);

The only location where the total exergy is predicted to increase is Newcastle. In this case, the predicted exergy is highly correlated to the heating load.

The analysis shows that the new TRYs are likely to impact building design, since they appear to be more likely to overheat. However, it is difficult to predict how an increased rate of overheating in building models will affect building design. It does however underscore the importance of updating the TRYs used in building design, since using out-of-date weather files could lead to new building designs not being sufficiently prepared for the change in the climate.

4.5 Summary

In this chapter we have examined the basics of how Kriging can be used to build a meta-model for a simple building energy simulation and shown that this method works well for energy use predictions.

We then applied this approach to a statistical analysis of the effect of using updated TRYs on a range of different building types. Using this method, 10,000

buildings in 14 locations were analysed, which is an analysis that would not have been possible using the original engineering model alone (at least not in a reasonable timeframe). The results provided some useful information on how the new TRYs could influence future building design in the UK.

In chapter 3, I explore further applications of Kriging, this time in the field of building optimisation. In this work, I introduce the concept of *efficient global optimisation* using Kriging and demonstrate its application to a simple building design problem.

Chapter 5 Kriging optimisation methods

5.1 Introduction

In the last chapter, I demonstrated how Kriging can be used to create simple emulators for buildings. I then adapted the model for use in a statistical analysis to test the effect of new UK Test Reference Years. In this chapter, I explore the use of Kriging as an optimisation tool and demonstrate its application in a building design optimisation problem.

5.2 Background

The Kriging emulation method has a number of advantages over other Kriging methods. These advantages are due to the fact that the emulator makes predictions using both a mean *and* a covariance function. The existence of both these functions means that I have additional information that I can exploit. Consider the example in Figure 5-1 where I am trying to emulator the *toy* function $y = -(\cos(5x) + 2x^2 + \sin(10x))$.

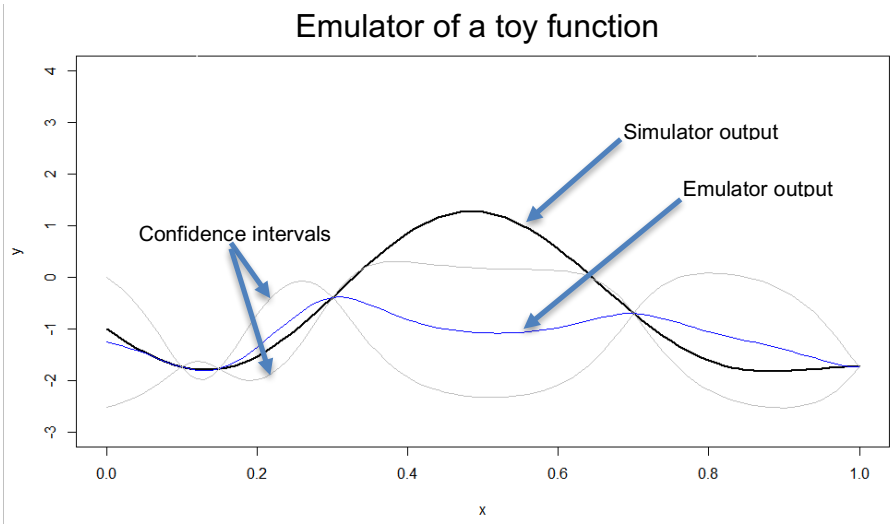


Figure 5-1: An example Kriging model for a toy *function*

Figure 5-1 shows that some areas of the emulator perform poorly (around x = 0.5). In these areas, the mean function diverges significantly from the simulator output. However, as the 95% confidence intervals on the graph show, the emulator 'knows' that its predictions are likely to be poor in this area. I can therefore exploit this information to improve the emulator.

5.2.1 A simple strategy for improving the emulator

A simple strategy for improving the emulator would be to add an extra simulation point where the predicted covariance is at its highest and re-train the emulator. Figure 5-2 shows the new emulator having had an additional training point added at $x = 0.5$.

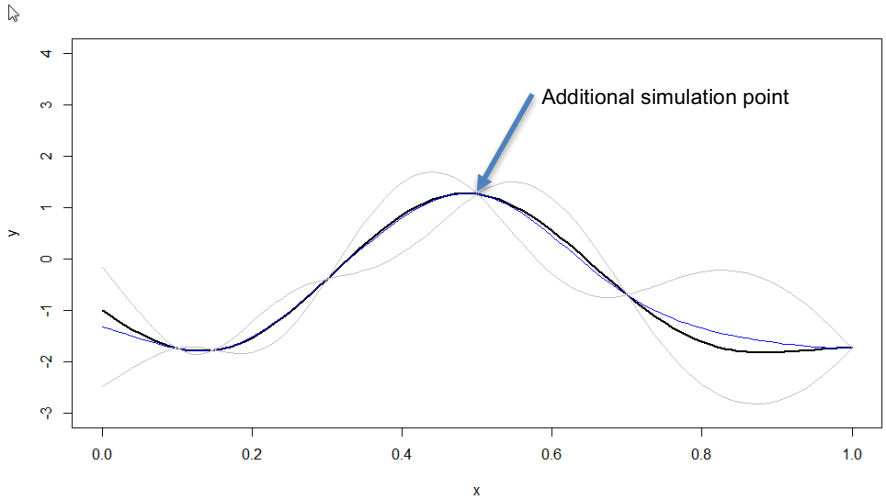


Figure 5-2: An improved emulator for the toy *function using an additional simulation point*

I could then iterate this approach to create an emulator where the mean function emulates the simulator output more and more closely.

However, in many of the problems that I am trying to solve, it is not necessarily a good strategy. For example, if I were to use the emulator for optimisation, I might approach emulator improvement in a different way.

5.2.2 Kriging optimisation

In a global optimisation problem, I am interested in minimising or maximising a given objective function. To do this, I might add a training sample at the point where the *mean* function is the lowest as shown in Figure 5-3.

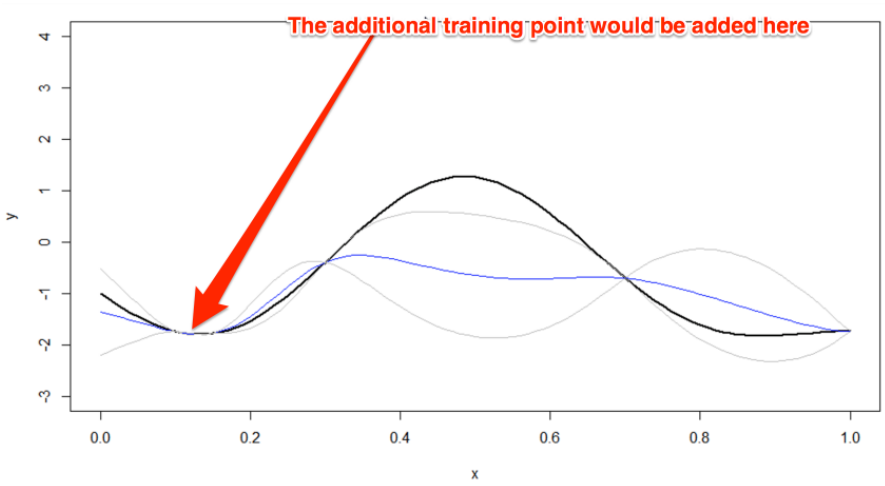


Figure 5-3: A redundant additional simulation point

However, as we can see, the algorithm has some obvious drawbacks:

- There is nothing to stop the emulator adding points where they are not needed
- it does not explore the emulator fully

I therefore need a more efficient approach if I am to use the emulator for optimisation.

5.2.3 Efficient Global Optimisation using Expected Improvement

An effective method for Kriging-optimisation is efficient global optimisation with expected improvement (EGO) [153]. Like the simplified example above, EGO uses information from the emulator's mean and covariance functions to identify areas of the emulator where it is *likely* that the global optimum will be found.

Suppose I have a function that I am trying to minimise. If I have already trained the emulator with a sample set $\boldsymbol{D}$, then I might find a good candidate for the optimum in the output set $f(\boldsymbol{D})$. For a minimisation problem, this would be $y_{\min}$.

If $y_{\min}$ is the true optimum, then I do not need to go any further. However, I have no way of knowing whether this is true of not. If I assume that I have not yet found the optimum, then any output of the simulator has the potential to improve on $y_{\min}$. I define the improvement $I(\boldsymbol{x})$ as:

$$I(\boldsymbol{x}) = y_{min} - f(\boldsymbol{x}) \qquad \text{Eq. 24}$$

To find the optimum of the emulator, I therefore need to maximise the improvement. However, this approach does not get us very far in solving the problem, since I would have to evaluate $I(\boldsymbol{x})$ over the whole input space. This is clearly not a useful approach.

I can, however, substitute the output of the emulator for the output of the simulator to define the *expected* improvement over $y_{\min}$ for a given input $\boldsymbol{x}$. The expected improvement (EI) is defined as:

$$E\big(I(\boldsymbol{x})\big) = y_{min} - \hat{y}\,\boldsymbol{\Phi}\left(\frac{y_{min} - \hat{y}}{s}\right) + s\,\phi\left(\frac{y_{min} - \hat{y}}{s}\right) \qquad \text{Eq. 25}$$

Where $\boldsymbol{\Phi}$ is the Normal cumulative distribution function, ϕ is the Normal probability density function, $y_{\min}$ is the lowest sampled value (so far), $\hat{y}$ is the mean output of the emulator and s is the mean squared error (MSE) of the emulator [154]. A full derivation of the EI function can be found in Knowles [154].

To maximise the expected improvement, I still need to maximise the function $E[I(\boldsymbol{x})]$, but since I only evaluate this function using the emulator, it becomes a much more manageable problem than trying to maximise $I(\boldsymbol{x})$ directly.

In simple terms, the equation calculates the relative *probability* that the *real* simulator output is below $y_{\min}$ at x. The mean function may run above $y_{\min}$, but if a significant portion of the probability density function is below it, then this may be a suitable candidate for a new simulation point.

The derivatives-based GA package *rgenoud* for the maximisation of $E[I(\boldsymbol{x})]$ [137], [138].

5.3 Aims

As we have seen in chapter 1, GAs are the most commonly used tool by researchers for optimising buildings by direction application on the building simulator. Therefore, in order test the efficacy of GAs, I set up a building model to compare the efficiency of GA optimisation techniques to Kriging optimisation with EGO Method.

The building model used in the assessment is a single story medium office with a corridor to the north façade and windows to the south, east and west, each with a brise soleil (Figure 5-4).

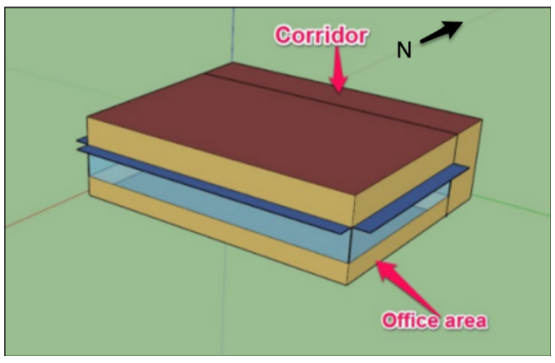


Figure 5-4: Building model to be optimised

The construction of the building is based on the Medium Office of ASHRAE 189.1-2009 [155]. In optimising the model, I varied the following inputs detailed in Table 5-2.

Table 5-1: Constructions used in the building model

Element	Construction
External walls	25 mm Stucco 200 mm Concrete (heavyweight) Wall insulation 40 mm 12.5 mm Gypsum
Windows	ASHRAE 189.1-2009 *ExtWindow ClimateZone 4-5*
Ceiling	M11 100 mm lightweight concrete F05 Ceiling air space resistance F16 Acoustic tile
Roof	Roof Membrane Roof Insulation [21] Metal Decking
Internal wall	19mm gypsum board Air gap 19 mm gypsum board

Table 5-2: Inputs varied in the 9-input building model

Variable	**Lower Range**	**Upper Range**
Building floor area	200 m^2	1000 m^2
Aspect ratio	1	10
S window to wall ratio	0.02	0.9
E window to wall ratio	0.02	0.9
W window to wall ratio	0.02	0.9
S shade projection factor	0.05	1
E shade projection factor	0.05	1
W shade projection factor	0.05	1
Orientation	-90°	90°

The variables in Table 5-2 were chosen so that a wide range of building designs could be covered. The variables are not intended to recreate a typical design problem, but rather to provide a large search space of continuous variables to provide a thorough test of both the GA and Kriging optimisation methods.

The annual energy use of each building was predicted using *EnergyPlus* [22] with weather data from the test reference year for Cardiff, UK [156].

5.3.1 The GA optimisation settings and software

I implement the GA optimisation using the *GA* package in *R* [157], which implements GA optimisation as follows:

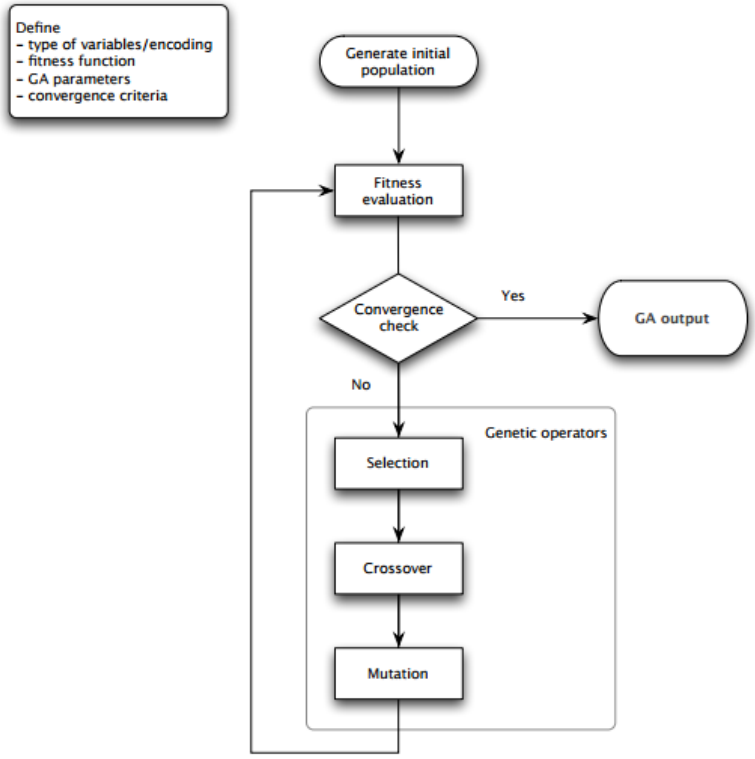


Figure 5-5: Overview of the algorithm used by the R package GA *(this figure is reproduced from earlier for convenience)*

In our analysis, I maintain 5% of the best buildings between each iteration. This ensures that the best results are not lost between iterations and is known as *elitism*.

The selection, crossover and mutation process is repeated for a set number of iterations. The best output from the final population is the 'optimum' solution. For further details on these operation, please see Scrucca's description of the *GA* package [157].

5.3.2 Kriging with Efficient Global Optimisation

To optimise the emulator, I used the EGO method and added 7 additional training simulations to improve the emulator[9]. The steps in this process were to:

1. Create a set of design input points, $f(\boldsymbol{D})$ using a maximin Latin Hypercube design;
2. Run the simulator to determine $f(\boldsymbol{D})$ for each of the inputs;
3. Build a Kriging emulator using $\boldsymbol{D}$ and $f(\boldsymbol{D})$;
4. Maximise the expected improvement criterion (EI) to determine the best input configuration for the next simulation ($\boldsymbol{D}_{\mathrm{opt}}$);
5. Use $\boldsymbol{D}_{\mathrm{opt}}$ from step 4 to re-estimate the Kriging model (including covariance parameters re-estimation) based on the $\boldsymbol{D}$ and $\boldsymbol{D}_{\mathrm{opt}}$;
6. Repeat steps 4-5 up to 7 more times adding the new $\boldsymbol{D}_{\mathrm{opt}}$ point to the Kriging model each time.

The result of the optimisation process is $\min(\boldsymbol{D})$, which is the minimum of the final training set.

5.3.3 Test setup

I created Kriging models with varying numbers of simulations (Table 5-3). The number of simulations includes the initial training set and the seven additional simulations which were added at the points in the model where $E(I)$ was the highest.

[9] Seven simulations we chosen as these we found to provide the most effective improvements for the minimum number of additional simulations.

Table 5-3: Number of training simulations and repetitions of the Kriging algorithm (these were chosen to roughly mirror the number of simulations in each GA routine in Table 5-4)

Number of simulations (training + EI evaluations)	Number of Kriging emulators	Repetitions
37	4	3
57	4	3
107	3	3
207	4	3
407	5	3

The number of simulations used by the GA depends on the ratio between the size of the population and the number of iterations. The number of simulations is therefore equal to the product of the population size and the number of iterations.

Alajmi and Wright [61] have shown that the choice of population size has the biggest effect on the performance of the GA. Each GA was tested with different population sizes and generation numbers. For example, a 100-simulation GA can be set up as a population of 20 with 5 iterations, or a population of 10 with 10 iterations. Note that these routines are performed on the *simulator*, not the emulator. The different routines are intended to test a range of different GA setups.

Table 5-4 shows the GA simulation routines that I used in the assessment. Note that these routines are performed on the *simulator*, not the emulator. The different routines are intended to test a range of different GA setups.

Table 5-4: Populations, iterations, simulations and total repetitions used for the GA routines

Routine no.	Population	Iterations	Total Sims	Repetitions
1	2	15	30	3
2	2	25	50	3
3	3	10	30	3
4	5	10	50	3
5	5	20	100	3
6	5	40	200	3
7	5	80	400	3
8	10	3	30	3
9	10	5	50	3
10	10	10	100	3
11	10	20	200	3
12	10	40	400	3
13	15	2	30	3
14	20	5	100	3
15	20	10	200	3
16	20	20	400	3
17	25	2	50	3
18	40	5	200	3
19	40	10	400	3
20	80	5	400	3

5.4 Results

Figure 5-6 shows the best results from each GA optimisation process against the number of simulations required to reach it. For the GA optimisation, the number of simulations required to reach each optimum point the size of the population multiplied by the number of iterations. Each point in the graphs is the result of a single optimisation procedure.

The results from the Kriging approach are shown in Figure 5-7. Each point in the Figure 5-7 represents one EGO process. In this case, the number of simulations are equal to the number of samples in the training simulation plus the number of steps use in the EGO process.

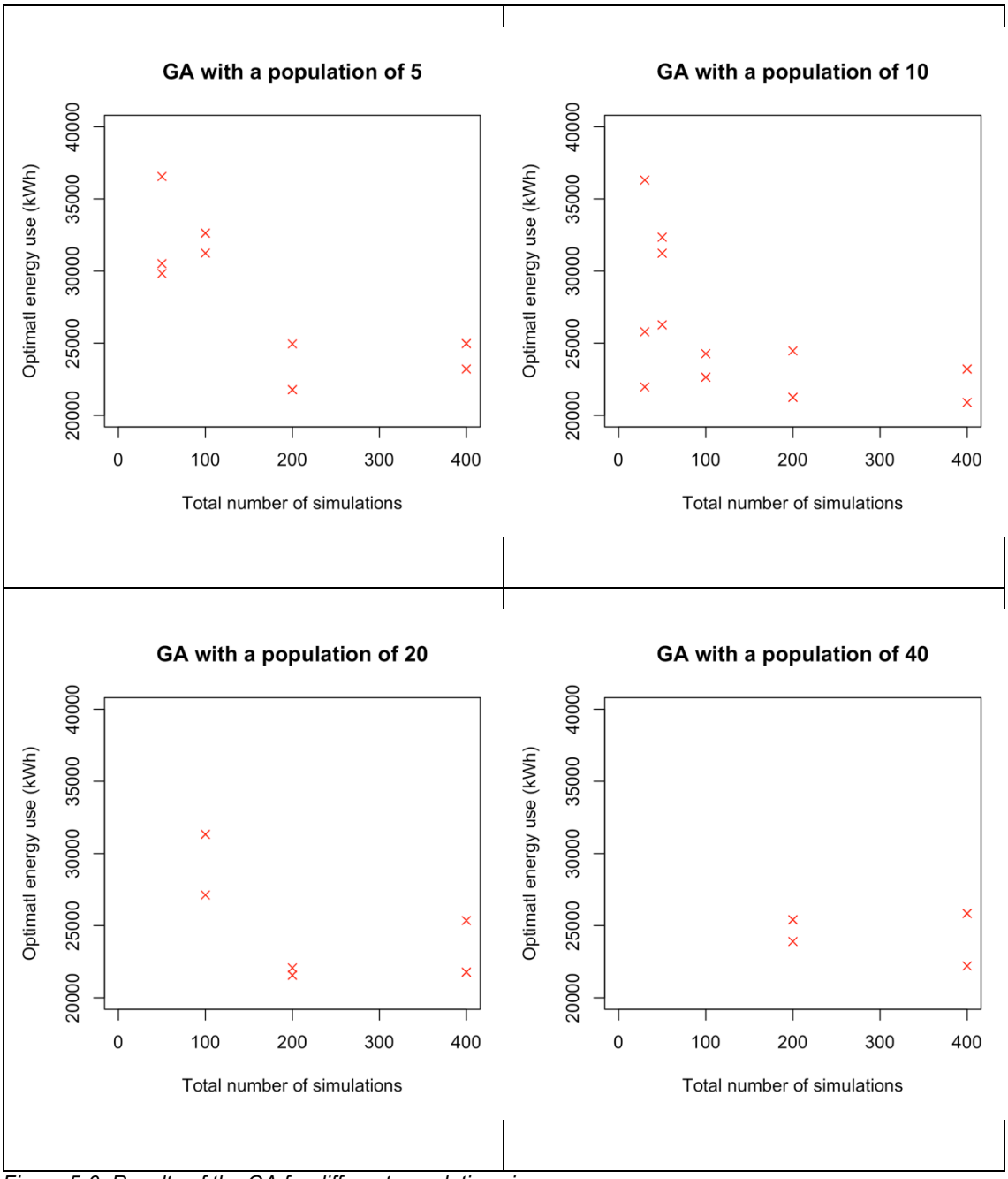


Figure 5-6: Results of the GA for different population sizes

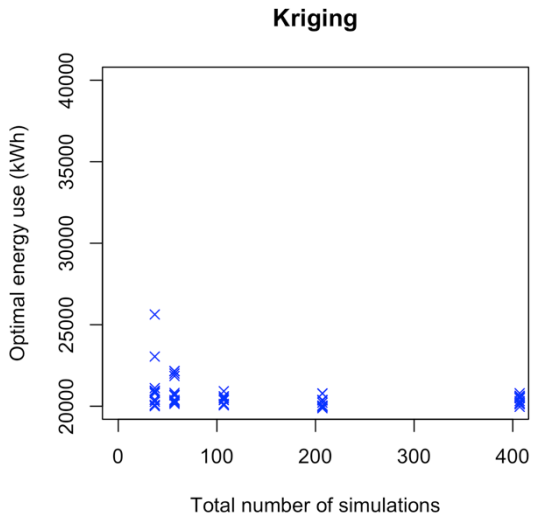


Figure 5-7: Range of the results from Kriging

5.5 Discussion

The Kriging-based EGO optimisation method outperforms the GA in terms of convergence and stability. GAs perform best with a small population size, but the results of this assessment show that they are still far behind the performance of the Kriging model.

In examining the two 'best' buildings that were designed by the two algorithms, we find that the energy use of each are within 5% of each other. Table 5-5 shows the two solutions.

Table 5-5: Solutions offered by the Kriging and GA approaches

	Kriging	Genetic algorithm	Unit
Number of simulations	407	400	Integer
Best predicted annual energy use	19934	20894	kWh
Building area	200	206	m^2
Aspect ratio	1.00	1.16	Ratio
Glazed ratio (S)	0.59	0.37	Ratio
Glazed ratio (E)	0.02	0.42	Ratio
Glazed ratio (W)	0.15	0.16	Ratio
Projection factor (S)	0.65	0.47	Ratio
Projection factor (E)	1.00	0.35	Ratio
Projection factor (W)	0.05	0.31	Ratio
Orientation	-58	1	Degrees

The major differences between the two solutions are the glazing ratio, the *brise soleil* overhang factor and the orientation. However, given that the two buildings have roughly the same floor area and aspect ratio, the form of both buildings are very similar (see Figure 5-8 and Figure 5-9). Both buildings have different orientations but have little or no glazing to the Northerly façade.

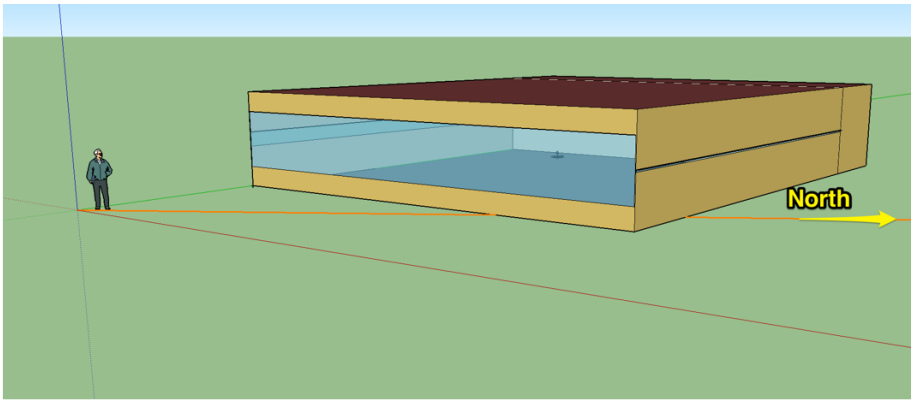


Figure 5-8: Best solution by Kriging

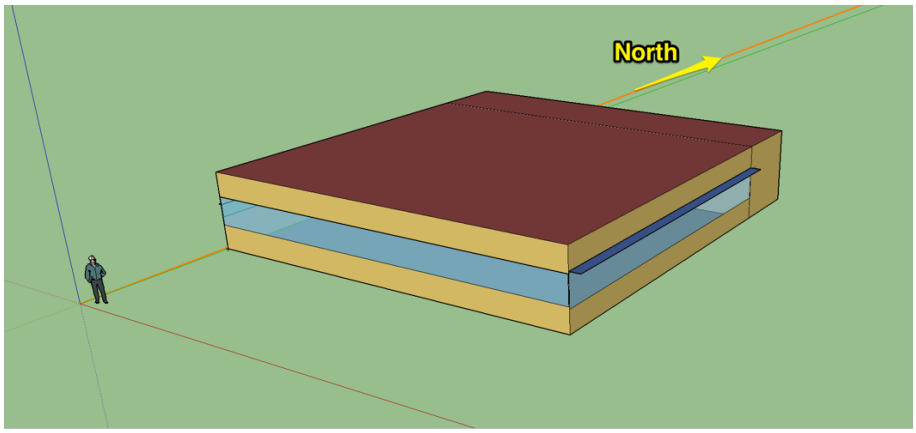


Figure 5-9: Best solution by GA

5.6 Summary

The results of the optimisation approaches show that Kriging-based EGO optimisation appears to exceed the performance of the GA alone, both in its search efficiency and in its stability.

There are however some practical limitations that need to be explored further. These include the limitation on the number of inputs that can be reasonably considered with the Kriging model. As we have seen from Chapter 2, the maximisation of the log-likelihood function needs to be evaluated for each step. Since each step of the maximisation process needs to invert the A matrix, then the computation time taken for each step in this process is $\propto n$ (where n is the number of elements in the emulator's training set). For each step added to the optimisation, the emulator is rebuilt. This means that the log-likelihood maximisation problem grows with the size of the emulator. This is likely to be an important consideration, particularly when considering high-dimensional optimisation problems.

The next chapter focusses on uncertainty management using Kriging.

Chapter 6 Uncertainty management

6.1 Introduction

In almost all areas of engineering, consideration is given to how input uncertainty affects the overall performance. However, this is rarely considered in building design. This is all the more concerning, since buildings are frequently shown to underperform when it comes to important design features, such as energy use [1], [85]–[87]. This is particularly important when considering investment decisions for energy improvement measures [158].

Uncertainty analysis in regards to building design can be described as a way of;

> *"identifying uncertainties in [the] input and output of a system of simulator tool."* [108]

The main motivations for being able to undertake an uncertainty analysis of a building design are to;

1. enable simplification of the building model;
2. allow analysis of the model's *robustness*;
3. identify unexpected sensitivities that may lead to errors and incorrect specifications; and
4. provide decision support for designers and policy-makers.

6.2 Aims

In the work described in this chapter, I aim to address the third point above. I will examine the application of Kriging methods to determining the potential uncertainty in energy conservation measures applied to a neighbourhood of buildings.

6.3 Method

I compared Kriging methods with a so-called *brute-force* approach to uncertainty analysis. To do this, I analysed a group of hypothetical houses to be retrofitted with additional insulation.

The hypothetical group of houses were assigned different design parameters at random. I also assumed that there was some degree of uncertainty associated with each of these parameters. This would then lead to an overall uncertainty in the total energy use of the 'neighbourhood'.

The first stage of the investigation was to test how well a Kriging emulator can predict the uncertainty distribution of this total energy use. To do this, I compared the output of an MC analysis on the Kriging model to the output of a 'brute-force' MC analysis on the original building model.

The second stage was to predict the impact of improving the insulation in all of the houses in the neighbourhood. Again, I compared the output of the MC on the Kriging model to the original building model for the improved case. I also tested:

- how well the emulator represents the annual energy consumption for the baseline and improved cases; and

- how well the emulator represents the *improvement* in the energy use of the buildings (i.e. by training the emulator on the *improvement* in energy use for the training sample)

The *brute-force* approach provides a baseline to comparison for the Kriging-based results. By comparing the results of the Kriging models with the *brute-force* approach, I was be able to determine how well the Kriging model represents the output.

6.3.1 Overview of the process

I investigated the effect of uncertainty in the input parameters of 20 different buildings. The buildings considered in the 'neighbourhood' each has a different orientation, insulation, infiltration rate and heating set point level (Figure 6-1 and Table 6-1).

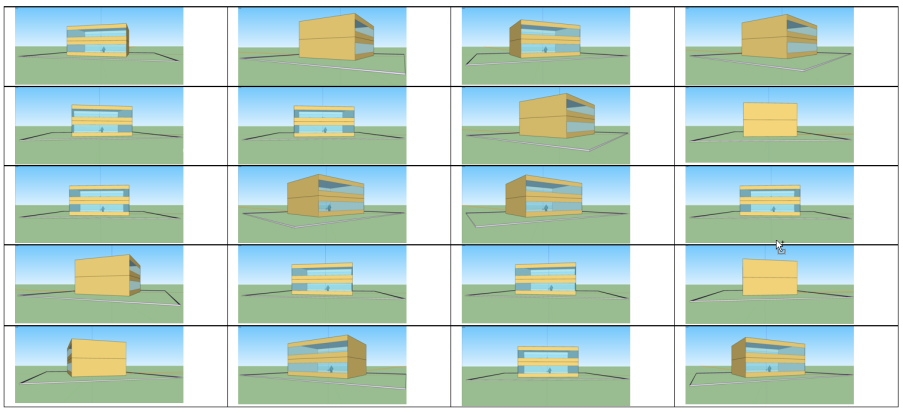


Figure 6-1: Range of buildings used in the neighbourhood model

Table 6-1: The neighbourhood of building under analysis

Building No.	Wall insulation thickness (m)	Roof insulation thickness (m)	Orientation (deg)	Air Permeability (ACH)	Set point (deg)
1	0.08	0.235	-16	2.8	22.9
2	0.094	0.281	58	5.5	21.3
3	0.029	0.072	26	2.2	22.1
4	0.069	0.199	51	3.1	21.2
5	0.021	0.046	10	2.4	21.0
6	0.034	0.087	5	4.1	22.3
7	0.045	0.122	52	2.0	20.6
8	0.011	0.014	-86	3.2	22.1
9	0.044	0.121	-4	4.7	20.4
10	0.088	0.262	42	1.1	20.7
11	0.041	0.109	35	5.3	20.4
12	0.053	0.15	-4	2.4	20.7
13	0.064	0.184	65	5.1	20.2
14	0.054	0.153	-11	2.5	21.9
15	0.027	0.064	-46	2.4	22.6
16	0.084	0.25	-77	3.2	22.3
17	0.07	0.204	-72	5.4	22.4
18	0.081	0.24	-33	5.3	21.4
19	0.02	0.041	3	2.7	21.2
20	0.075	0.22	29	4.8	22.4

The buildings in Table 6-1 are intended to represent a diverse range of buildings that exist in a typical neighbourhood. Each building in the table represents its ‘pre-insulation treatment’ condition, with each building having different levels of insulation, air permeability and heating set points. These are intended to simulate the kinds of differences that might be found in a typical housing survey (though they are not intended to be representative of any particular housing stock).

In assessing the improvement of each of the buildings, I assumed that the wall and roof insulation levels are improved to a new common baseline. I then

modelled the effect of this treatment on the annual energy use of the neighbourhood.

6.3.2 Assumed improvements to the buildings

In our energy improvement scenario, I assume that I am only treating the wall and roof insulation to bring each of the houses up to a common baseline. I assume that each building in the neighbourhood is improved so that;

- there is 300 mm of roof insulation
- there is 100 mm of wall insulation

In modelling the improvements, I do not change any other aspect of the building setup. However, I do still assume that the other variables have a certain amount of uncertainty.

6.3.3 Other variables

Notwithstanding the individual building variations shown in Table 6-1, each building has 50% glazing to the north and south facades[10], an aspect ratio of 1.25 and an assumed thermal mass layer 100 mm thick on the first and ground floors. The thermal mass used in the model is equivalent to dense concrete and covers an area equivalent to the ceiling area. The properties of the other construction materials are shown in Table 6-2 (glazing properties) and Table 6-3 (other construction elements).

[10] Note that this glazing is nominally to the north and south facades when the building is a 0 degrees orientation, the orientation of the glazed facades therefore changes with the orientation of the building

Table 6-2: Properties of the glazing

	Low emissivity clear 3 mm	Argon gas	Clear glazing 6 mm
Thickness (mm)	0.003	0.013	0.006
Solar transmittance at normal incidence	0.63	-	0.775
Front side solar reflectance at normal incidence	0.22	-	0.071
Back side solar reflectance at normal incidence	0.19	-	0.071
Visible transmittance at normal incidence	0.85	-	0.881
Front side visible transmittance at normal incidence	0.079	-	0.08
Back side visible transmittance at normal incidence	0.056	-	0.08
Infrared transmittance at normal incidence	0	-	0
Front side infrared hemispherical emissivity	0.1	-	0.84
Back side infrared hemispherical emissivity	0.84	-	0.84
Conductivity	0.9	-	0.9

Table 6-3: Construction properties used in the building

	Thickness (m)	**Conductivity (W/mK)**	**Density (kg/m3)**	**Specific Heat (J/K)**
Wall Construction				
100 mm Brick	0.2	0.89	1920	790
Insulation	*VARIABLE*	0.03	43	1210
100 mm Brick	0.2	0.89	1920	790
Roof construction				
Clay tile	0.0127	0.84	1900	800
Membrane	0.0001	1	1100	1000
Insulation	*VARIABLE*	0.03	43	1210
Plasterboard	0.0125	0.21	700	1000
Ground floor construction				
Insulation	-	0.03	43	1210
Concrete	0.1	2.3	2300	1000
Air	0.1	0.0257	1.205	1005
Chipboard	0.02	0.13	500	1600
Carpet	0.01	0.04	160	1360
Mid-floor construction				
Carpet	0.01	0.04	160	1360
Chipboard	0.02	0.13	500	1600
Insulation	0.11	0.03	43	1210
Plasterboard	0.0125	0.21	700	1000

6.3.4 Uncertainty Bounds

In creating the neighbourhood model, I assume that there is some uncertainty in the parameters. These uncertainties are shown in Table 6-4.

Table 6-4: Uncertainty ranges used in the analysis

	Min	Max	Range	Uncertainty range	Unit
Wall insulation thickness	0.01	0.1	0.09	0.018	mm
Roof insulation thickness	0.01	0.3	0.29	0.058	mm
Air permeability (ACH)	0.6	6	5.4	1.08	ACH
Heating set point	20	23	3	1.5	ºC

The ranges above are intended to represent uncertainty in the both the survey data and the occupancy behaviour.

6.3.5 The occupancy model

In addition to the data above, I also assumed that the estimated number of occupants, which can vary between 1 and 5, also had an uncertainty of ± 1. The effect of occupancy was modelled using the CREST domestic energy use model [80]. The CREST model is based on the Time Use Survey (TUS) data [159].

The data from the TUS is used to create Markov Chains [160] of the probability of the house being occupied (and by how many occupants) at a given time of the day. The day is divided up into 10-minute steps. The probability of the occupation level for the next 10-minute period is dependent on the previous 10-minute period. To illustrate this, below is an example of one of the probability matrices used in the Markov Chain:

Table 6-5: Example of a Markov chain probability table

		Next state at 21:10		
Current state at 21:00	No. of occupants	0	1	2
	0	0.892	0.082	0.025
	1	0.038	0.878	0.084
	2	0.003	0.043	0.954

Table 6-5 is used to predict the probability of no of occupants at 2110 based on the number of occupants present at 2100. For example, if the house is unoccupied at 2100, then the probability that there will be one occupant at 21:10 is 0.082. Similar probability matrices are used for creating transitions for each 10-minute time-step.

Probability matrices like those above were used to generate five occupancy profiles, each for houses with 1 to 5 residents. Note that this makes the assumption that the *activity* of the residents is the same across all simulations

when the number of residents is the same. In other words, I am saying that when the house has the same number of occupants, the occupants' interaction with the building does not change. For the purpose of this assessment I view this as a reasonable assumption, since other aspects of behaviour that affect energy use (such as heating set point and ventilation) are being modelled as uncertain parameters.

6.3.6 Emulator setup

I tested 9 emulation types for determining the pre and post treatment of energy distributions, as well as the distribution in the *difference*. The reference codes for theses emulators are shown in Table 6-6.

Table 6-6: Emulator types used in the analysis

Correlation function of the emulator	**Pre Treatment emulator name**	**Post Treatment emulator name**	**Difference emulator name (emulation of pre minus post treatment energy use)**
Matern 5	M5Pre	M5Post	M5Diff
Matern 3	M3Pre	M3Post	M3Diff
Gauss	G5Pre	G5Post	G5Diff

The two methods for emulating the *improvement* to the building performance are:

- Using the difference between the 'before' and 'after' emulators to model the effect of the improvement (e.g. before insulation improvement *minus* after insulation improvement)
- Creating a dedicated emulator for 'emulating the difference'. In other words, an emulator trained on $f(x) = y_{\text{before}} - y_{\text{after}}$.

In each of these cases I assumed that each building had been treated with additional insulation so that the total wall insulation was 100 mm and the total roof

insulation was 300 mm. However, the uncertainties regarding the other building parameters still remain.

6.3.7 Monte Carlo Analysis

In order to propagate the uncertainty to the output, I used a Monte Carlo (MC) analysis. So that I could compare the emulated outputs to the 'real' simulator output, I performed the MC analysis with both the simulators and emulators pre and post treatment. I can then compare the results of the 'brute force' MC analysis completed by the simulator with the results of the different emulation approaches.

I have 20 buildings, and for each building I took 100 draws from the uncertainty distributions. This means that the brute force MC analysis sampled the building simulator 2000 times. I then undertook the same number of samples from each of the emulators and compared the results.

6.4 Results

I present the results of the MC analysis in three stages. First I have the results of the *baseline* modelling, which is the results of the emulation and simulation of the buildings as they currently are (with the associated uncertainty). I then present the same results, this time for the *improved* scenario. These are the results of the MC analysis run on both emulator and simulator for the buildings that have been improved, again with the associated uncertainties. The third stage is in two parts:

1. Histograms of the results of the difference between the MC analysis of the baseline and improved simulator, and the baseline and improved emulator (i.e. $\mathrm{hist}(\hat{f}(x_{\mathrm{improved}}) - \hat{f}(x_{\mathrm{baseline}}))$ and $\mathrm{hist}(f(x_{\mathrm{improved}}) - f(x_{\mathrm{baseline}}))$)

2. Histograms of the results of the difference between the MC analysis of the baseline and improved simulator, and the *emulated difference* (i.e. $\mathrm{hist}(\hat{f}(x_{\mathrm{improved}} - x_{\mathrm{baseline}}))$.

The results of the baseline modelling are in Figure 6-2 below with the *improved* building results shown in Figure 6-3. The *difference between the emulators* and the *emulated difference* are shown in Figure 6-4 and Figure 6-5. The graphs for each of the different correction function types (i.e. Matern 5/2, Matern 3/2 and Gaussian), have all been plotted against the MC output for the simulator. This allows the performance of each correlation function type to be compared to the MC analysis of the simulator.

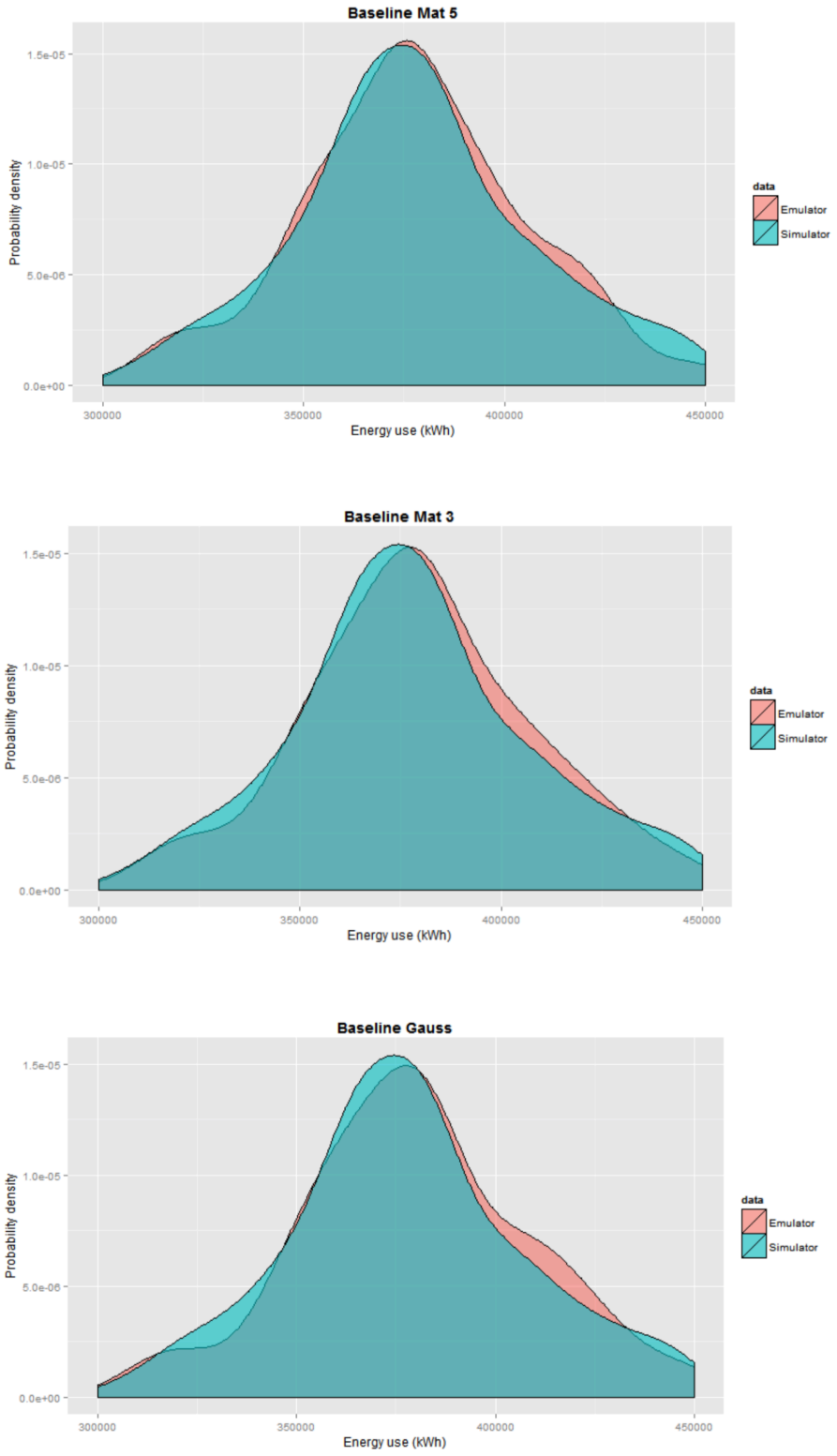


Figure 6-2: MC results for the baseline case

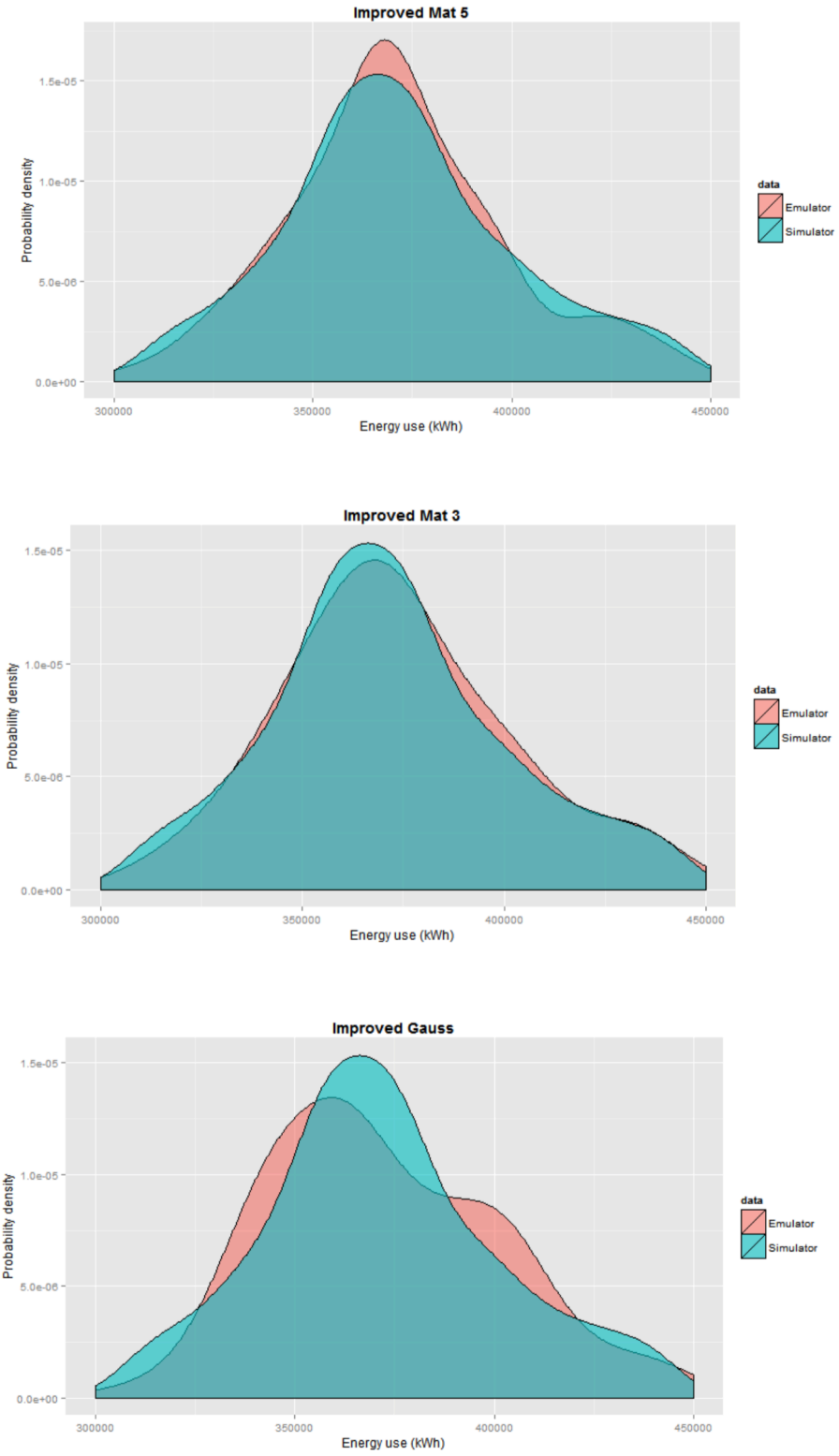


Figure 6-3: MC results for the improved case

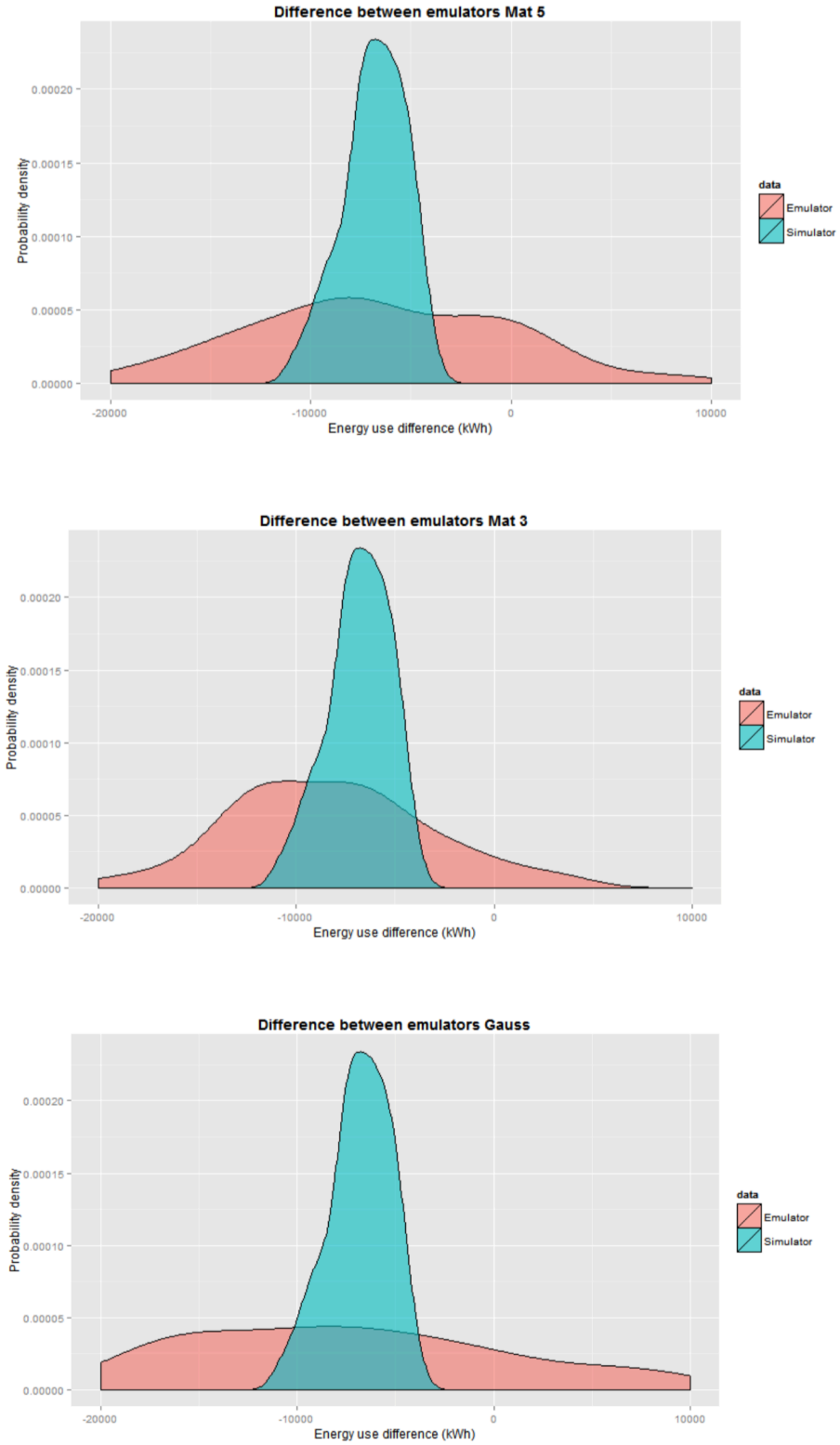


Figure 6-4: MC results for the difference of the emulators

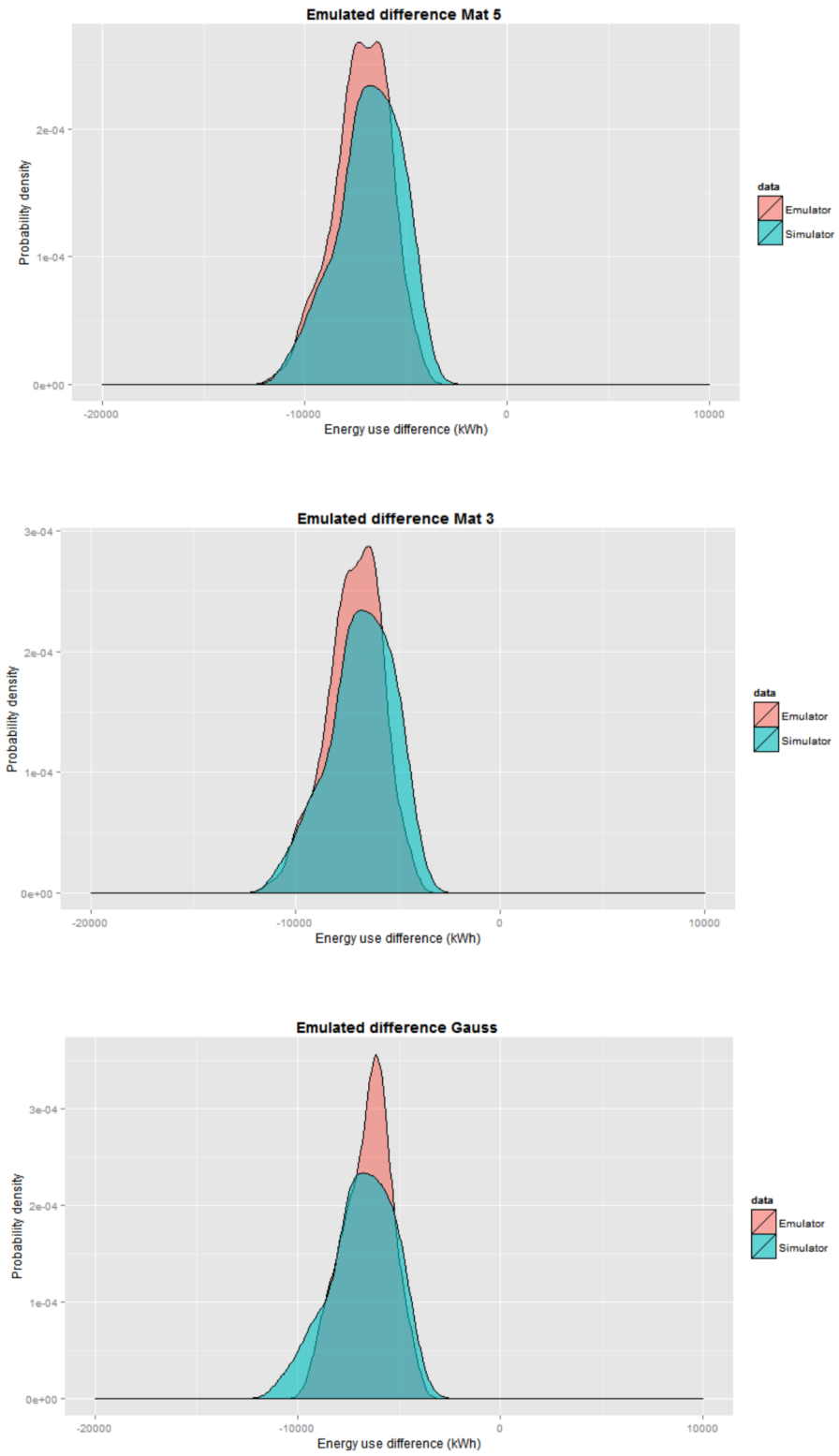


Figure 6-5: MC results for the 'emulated difference'

6.5 Discussion

The results of the MC analysis on the simulator and the emulator show some successes and some anomalies.

Regarding the building physics, the results show that the potential uncertainty in both the *baseline* and the *improved* case are large (in excess of 30,000 kWh). However, once the improvements are made, the uncertainty range for the improvements reduces to around 10,000 kWh. This means that we are more certain about the *magnitude* of the improvements in the energy performance over the baseline than the total amount of energy consumed. In other words, the results show more certainty about the energy *saved* than the absolute amount of energy *used*.

The results of the baseline emulators show that each of the Matérn $^3/_2$, Matérn Matérn $^5/_2$ and Gaussian emulators all perform well, with each fitting the results of the MC simulation well across the distribution of the energy use.

A similar pattern is observed across the emulators of the 'improved' building. However, although the Matérn $^3/_2$ and Matérn $^5/_2$ functions largely represent the output well (though less well than for the baseline building), the Gaussian function does diverge more significantly from the simulated MC output in some areas.

The greatest divergence between the simulator and emulator outputs is observed when I subtract the baseline and improved emulators to obtain the difference in the energy consumption. This is an indication that, although the emulator works well *on average* (as indicated in the baseline and improvement MC simulations), the *individual* pairs of emulator and simulator results can at times be quite far

apart, and the effects of this are clear. However, the results of the *baseline* and *improved* MC analyses show that these results appear to 'even out'.

If I look at the 'emulated difference' then I see much better agreement between the emulated outputs and simulated outputs. Each of the Matérn $^3/_2$, Matérn $^5/_2$ and the Gaussian correlation function fit the simulated output well. However, the Matérn $^3/_2$ has a slightly lower mode value, whereas the Gaussian has a slightly higher mode value than the simulator. Interestingly, the Matérn $^5/_2$ function appears to have two modes, with one slightly higher and one slightly lower than the simulator's mode.

These findings are potentially useful for this type of problem. However, it is interesting to speculate as to the cause of some of the divergences from the simulator's distribution. In the case of the baseline emulator, we see that the emulator represents the output well, but we begin to get divergences as we investigate the improved building emulator. I speculate that this could be due to the new shape of the output of the objective function.

In the improved emulator, I am keeping the parameters for the insulation in the wall and roof static, since I am improving them to a new baseline level. As the levels of insulation were different for all the houses before, I am likely to find that the output space of the objective function varies significantly, and there is no reason why the emulator would continue to perform well. In other words, I may be picking a hyperplane within the emulator where the emulation is not as good. If this is the case, this could contribute to the poorer emulation seen in the improved case. Investigating these issues is the subject of proposed further work.

6.6 Conclusions

I have shown that the emulator works well as an uncertainty analysis tool. However, it has to be designed in different ways depending upon the type of analysis required. In our example, I have shown that the approach of emulating the difference works much better than computing the difference between two emulators. However, there is the potential to explore other methods.

In the methods described in this chapter, I have only considered using the mean function for the uncertainty analysis and have not explored the option of improving the emulator. As we have seen in chapter 2, the emulator outputs both a mean function and covariance function, with the mean function providing an estimate of the simulator output and the covariance function providing an estimate of *confidence* in this prediction. Since I have an estimate of confidence, I can use this information to identify areas of the emulator that need improvement. Furthermore, if we can develop areas of the emulator that we are interesting in improving (such as the hyperplane where the insulation of the wall and roof as of a particular value), as well as identifying poorly performing areas of the emulator, we would be one more step to creating a usable tool for building designers and policy makers to be able to make decisions.

6.7 Summary

In order to ensure that I am reducing the impact that the energy use of buildings has on the environment, I need to consider robustness of individual designs. The work on Kriging-based efficient global optimisation has been discussed in the previous chapter. In the following chapter I explore an enhancement to this

method which allows buildings to be optimised in a manner that is robust to uncertainties in the parameter inputs.

Chapter 7 Robust Optimisation Methods

7.1 Introduction

In the previous chapter, I have explored the impact of uncertainty on the results on the energy use profiles of buildings. This method is useful for examining the impact of particular solution (such as increasing the insulation level, as in the previous example), but on its own, it does not enable us to generate potential solutions to a problem, only to examine their effect.

One possible solution to this problem is to combine our knowledge about uncertainty with optimisation methods in order to develop better solutions. A solution that maximises the energy reduction whilst also minimising the uncertainty in the output would be useful.

7.2 Background

Robust optimisation (RO) is a term that is commonly found in engineering, though it rarely used in building design. In their paper 'a comprehensive survey of robust optimisation' [109], Beyer and Sendhoff identify four key sources of uncertainty. These are:

A. Changing environmental and operating conditions
B. Production tolerance and actuation imprecision
C. Uncertainties in the system output
D. Feasibility uncertainties

In buildings, uncertainties of type A could include factors such as the weather, climate and the occupants' behaviour. Type B uncertainties could be related to

uncertainties in the construction tolerances – for example, such as small changes in the real U-values of the insulation and walls, as well as differences in the infiltration rates and other tolerances.

Uncertainties of Type C are related to our uncertainties about how the building is actually performing. For example, if we were to measure the annual energy use, there may be some error associated with this measurement. This type of error would fall under type C.

Type D uncertainties have a subtler influence. Even if we represent the building by a very large number of parameters, we cannot capture every nuance of the design. By constraining the building to a set number of parameters we are introducing an unknown amount of uncertainty. This type of uncertainty is called *feasibility uncertainty.*

In my research, I have focussed on type A and B uncertainties. These uncertainties are sometimes referred to as *parameter uncertainties.*

7.3 Aims

In this chapter I aim to investigate an approach to robust optimisation (RO) proposed by Rehman et al. [16]. The method uses a Kriging to create a worst case emulator of the simulator's output, based on pre-defined uncertainty ranges in the input parameters. I aim to test this method on a simple building design problem, and to compare the robust optimum output to the 'traditional' Kriging EGO optimisation approach which I described in chapter 3.

In developing the uncertainty bounds for our input parameters, I considered the main sources of uncertainty in buildings. According to Gill et al. [83], the uncertainty in modelling has is origins in;

- occupant behaviour;
- daylighting control strategies;
- material property uncertainties / design parameters; and
- climatic data

In our analysis I primarily focus on those elements that are affected by occupant behaviour.

- Air changes per hour
- Heating set point
- Residents

I then aim to find an optimum solution for the uncertainties in the above by finding the best combination of wall insulation, orientation, roof insulation, thermal mass, aspect ratio, glazed fraction (north and south)

7.4 Method

As I have shown in chapter 3, Kriging-based EGO optimisation performs well and has fast convergence rates and stability. However simply minimising the objective function doesn't provide an indication of how robust the solution is.

A robust solution can be considered as an input configuration $\boldsymbol{x}$ where any change $\Delta\boldsymbol{x}$ results in a minimal change in the output function $f(\boldsymbol{x})$. As an

example, Figure 7-1 shows two solutions to a 1-D optimisation problem taken from Augusto et al. [161].

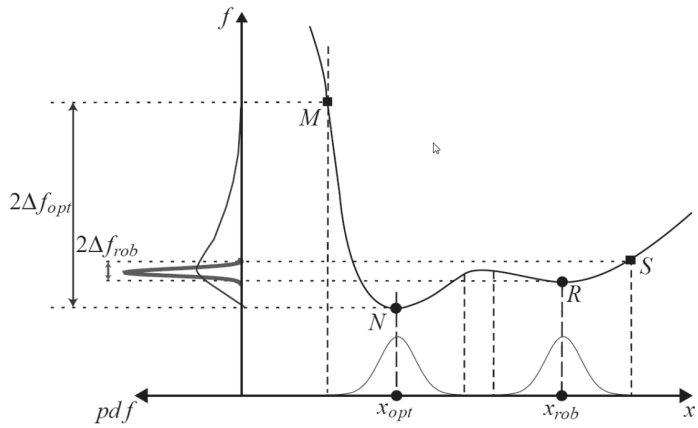


Figure 7-1: Example of a robust optimisation problem

The robust solution to the problem is point x_{rob} and the minimum of the objective function is shown by x_{opt}. Although the solution to the objective function R is greater than the global minimum N, this solution is less sensitive to the distribution of x. We can see that, if *pdfs* are applied to x_{rob} and x_{opt}, the distributions in the output $f(\boldsymbol{x})$ are very different, as shown in the figure: $2\,\Delta\,f_{\mathrm{opt}} \gg\ 2\,\Delta\,f_{\mathrm{rob}}$.

7.4.1 Robust expected improvement using Kriging

The example above is simple to apply when the objective function is known and is easy to evaluate. However, for more complex simulators, such as buildings, where the output could be extremely non-linear and where each point takes a significant time to evaluate, the problem becomes more difficult.

In their paper 'Efficient Kriging-based robust optimisation of unconstrained problems', Rehman et al. [16] introduced a new method based on robust optimisation of an adaptation of the expected improvement function.

At this stage, I revisit the form of the original EI function. The EI function is based on the simple premise of improvement over the best result for y that I have so far. In the case of a minimisation problem, the best y is $y_{\min}$, and at any new point, x, the improvement over our current minimum is $I = \max(y_{\min} - f(\boldsymbol{x}), 0)$. Using the emulator, I can think of this probabilistically as an *expected* improvement:

$$E[I(\boldsymbol{x})] = (y_{\min} - \hat{y})\, \boldsymbol{\Phi}\left(\frac{(y_{\min} - \hat{y})}{s}\right) + s\, \phi\left(\frac{(y_{\min} - \hat{y})}{s}\right) \qquad \text{Eq. 26}$$

Where $\boldsymbol{\Phi}$ is the normal cumulative distribution function and ϕ is the normal probability density function.

The EI function gives the most likely location of the simulation point where I am likely to achieve an improved output. However, I would like to have a function that estimates where the next best simulation point for the *robust* optimisation lies. For this I use the robust optimisation suggested by Rehman.

In a similar way to the EGO method, the robust optimisation method starts with the creation of a meta-model based on the initial set of sample and response data from the simulator. This process has three key steps;

1. Create a reference robust optimum on the meta-model;
2. Create a worst-case Kriging prediction, with respect to the uncertainty set U, estimated by the meta-model; and
3. Sample the highest EI over the current reference robust optimum.

The above process is repeatedly applied until convergence (i.e. when the model is no longer returning increasingly improved robust optima). Creating the worst-

case emulator is achieved using what I refer to as the *Rehman windowing algorithm*.

7.4.2 Rehman windowing algorithm.

The Rehman windowing algorithm is used to create a 'worst-case' emulator for a given model input vector $\boldsymbol{x}$. If $\hat{f}(\boldsymbol{x})$ refers to the Kriging model of $f(\boldsymbol{x})$, then, given a set of inputs $\boldsymbol{x}$, the predicted worst-case output for a given input $\boldsymbol{x}$ for an uncertainty set U is:

$$y_{\text{worst}} = \max\left(\hat{f}(\boldsymbol{x} + \Delta)\right) \qquad \text{Eq. 27}$$

Where $\boldsymbol{x} \in \mathbb{R}^n$ and $\Delta \in U$.

To compute the robust optimum, I can calculate the best worst-case result. For a minimisation problem, this is:

$$\hat{r}_{\text{opt}} = \text{minmax}(\hat{f}(\boldsymbol{x} + \Delta)) \qquad \text{Eq. 28}$$

Where $\hat{r}_{\text{opt}}$ is the predicted robust optimum. I can therefore express the best worst case at any input $\boldsymbol{x}$ as being:

$$\hat{y}_{\text{max}} = \max\left(\hat{f}(\boldsymbol{x} + \Delta)\right) \qquad \text{Eq. 29}$$

From this, I can derive the expected improvement on the robust function [16]. I can express the robust expected improvement as:

$$E[I_w(x_{max})] = \int_{I_w=0}^{I_w=\infty} I_w \left\{ \frac{1}{\sqrt{2\pi}s(x_{max})} exp\left[-\frac{(\hat{r} - I_w - \hat{y}_{max})^2}{2s^2(x_{max})} \right] \right\} dI_w \qquad \text{Eq. 30}$$

Then if I let

$$t = \frac{\hat{r} - I_w - \hat{y}_{\max}}{s(x_{\max})}. \qquad \text{Eq. 31}$$

Also, if I take into account that the standard normal probability density function is defined as:

$$\phi(z) = \frac{1}{\sqrt{2\pi}} exp\left(\frac{-z^2}{2}\right). \qquad \text{Eq. 32}$$

I can re-write the equation for the robust expected improvement function as:

$$E[I_w(x_{\max})] = (r_K - \hat{y}_{\max}) \int_{t=-\infty}^{t=\frac{\hat{r}-\hat{y}_{\max}}{s}} \phi(t)\mathrm{dt} - s \int_{t=-\infty}^{t=\frac{\hat{r}-\hat{y}_{\max}}{s}} t\phi(t)\mathrm{dt} \qquad \text{Eq. 33}$$

Then, as shown by Rehman, the equation can be simplified to:

$$E[I_w(x_{\max})] = (\hat{r} - \hat{y}_{\max})\Phi\left(\frac{\hat{r} - \hat{y}_{\max}}{s}\right) + s\phi\left(\frac{\hat{r} - \hat{y}_{\max}}{s}\right) \qquad \text{Eq. 34}$$

Comparing this to the original expected improvement function, I can see the following parallels between the variables:

- The robust optimum r_{opt} is analogous to absolute optimum $y_{\min}$

- The maximum value in the robust window of the $\hat{y}_{\max}$ is analogous to the actual emulator point $\hat{y}$.

7.4.3 Adapted windowing algorithm

In the analysis, we I used a pre-defined uncertainty set, rather than the maximisation approach used by Rehman et al [16]. This is similar to the methods described in Bertsimas et al. [120] and is used to estimate the likelihood that a performance criterion will be achieved, rather than the maximum possible error calculated by the Rehman windowing algorithm.

The windowing algorithm that has been proposed by Rehman et al. [16] finds *maximum* value in the uncertainty window. However, although this is a useful approach, there are some drawbacks of the Rehman approach in high dimensional spaces. For example, we may find a value based on our uncertainty bounds that is extremely unlikely to happen. A more reasonable approach to RO may be to estimate a confidence interval for the performance. I therefore propose a robust optimisation window where we minimise to the 95th percentile.

In order to calculate the location of the 95th percentile, I use a randomly generated *uncertainty set* U from which I estimate the 95th percentile. To implement this adapted algorithm, I use a simple Monte Carlo algorithm on the variables within that window. I assume that each of the variables has a flat distribution. The MC analysis on each point is performed on the emulator for 1,000 samples. The standard deviation of the results is calculated and assuming a Normal distribution, the approximate 95th percentile is calculated:

$$y_{\max,2\sigma} = f(x) + 2\times \mathrm{sd}(f(\boldsymbol{x}) + \Delta_{\mathrm{MC}}) \qquad \text{Eq. 35}$$

Where $\mathrm{sd}()$ calculates the standard deviation and Δ_{MC} is the uncertainty set of perturbations in $\boldsymbol{x}$ that constitute the MC simulation.

In performing the optimisation on the worst-case emulator, the emulator was updated at each step according to the following process:

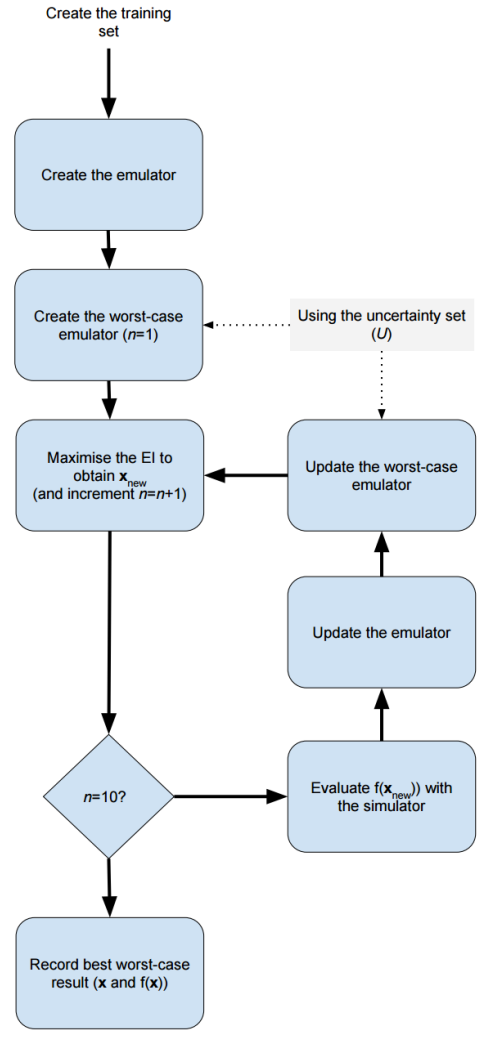


Figure 7-2: Algorithm for creating the confidence level emulator

7.4.4 Experimental setup

For the purposes of comparison, I compare the results of the robust optimisation with traditional EGO. The purpose of this was to see if the robust model really

does come up with more robust solutions that the standard EGO approach. Using the *RStudio* packages *DiceKriging* and *DiceOptim*, the vast majority of the computation time is taken up with running the simulator (the time taken to create and update the emulators is negligible compared to the time taken to run the simulator). I therefore use the *simulation count* as a proxy for the computation time. A fair test will therefore mean that both the EGO approach and the EGOrobust approach each take the same number of simulations to reach their conclusion. Table 7-1 shows the experimental setup.

Table 7-1: Setup for analysing the robust optimisation and global optimisation algorithms

Optimisation type	Training sims	No. of steps	Repetitions
EGOrobust	100	10	3
EGOrobust	100	20	3
EGOopt	100	10	3
EGOopt	100	20	3

7.4.5 Validation approach

To validate the results of the optimisation processes, I took a 4-step approach to the experiment. I:

- trained the initial emulators;
- ran the EGO and EGO robust optimisation processes;
- calculated the 'best' building in each case; and
- tested the robustness of each of the building solutions by brute force.

Both simulation processes will output a global optimum and a robust optimum solution. The solutions can then be used to produce uncertainty analysis based on the uncertainty limits placed on the input parameters. These plots are useful to give a visual representation of why the algorithm may have favoured that point (since is it the emulators on which the robustness and optimum points are

measured). However, this does not necessarily confirm that the new points found are as robust as required. I therefore validate the design using a 'brute force' MC uncertainty analysis.

7.4.6 The building model

I tested the robust optimisation method on a building similar to the building model used for the neighbourhood uncertainty analysis in chapter 4. However, this time I added additional input parameters.

In defining the optimisation problem, I first had to define the range of the input space to be searched (i.e. the limits on the input variables) and the size of the uncertainty in each input being considered.

Table 7-2 shows the new inputs that were added to the emulator and the uncertainties added to each variable in the robust optimiser.

Table 7-2: Uncertainty of variables used in the analysis

	+/-	**Units**
Aspect ratio	0.0015	ratio
Glazed fraction (north)	0.05	percentage
Glazed fraction (south)	0.05	percentage
Residents	2	count
Wall insulation	4.5	mm
Orientation	0.18	degrees
Roof insulation	14.5	mm
Thermal mass	9.5	mm
Air changes per hour	0.27	count
Heating set point	2.7	degrees

The ranges above were chosen to highlight variables with typically large uncertainties (such as the number of building residents and the heating set point), but also to consider the smaller effects of uncertainties related to glazed areas and the effectiveness of thermal insulation in the walls and roofs. In running the

model, it has been assumed that the variables do not exceed the ranges in Table 6-4.

7.5 Results

7.5.1 10 optimisation steps

The histograms for the EGO and EGOrobust are shown in Figure 7-3 to Figure 7-8 (see over page). These histograms show the results of an MC analysis on the solutions offered by both the robust optimisation routine (EGOrobust) and the traditional global optimisation approach. The same uncertainty set was used for the MC analysis in both cases. Since each algorithm was run three times, the algorithms are likely to arrive at different results each time the algorithm is run. I therefore ran an uncertainty analysis for each of the three different solutions offered EGO and EGOrobust.

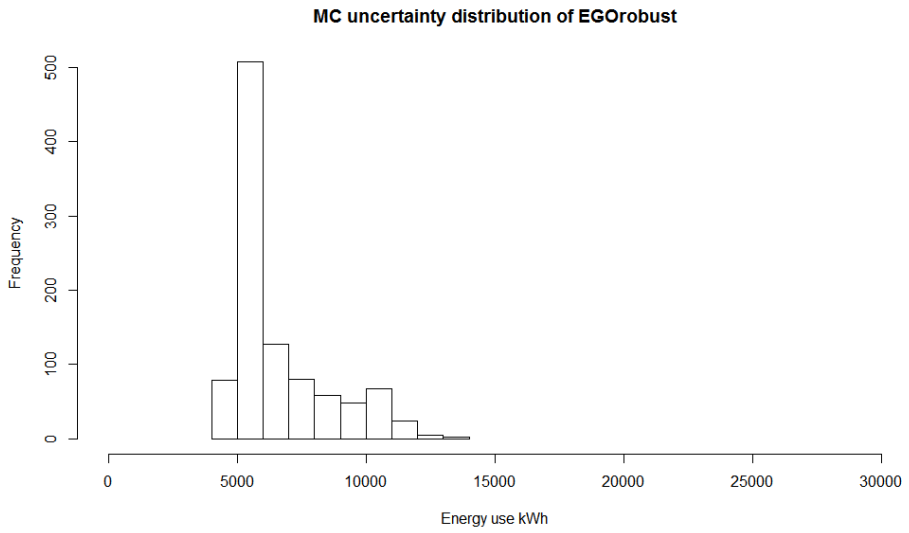


Figure 7-3: Histogram of MC analysis on the robust solution (10 optimisation steps, run 1)

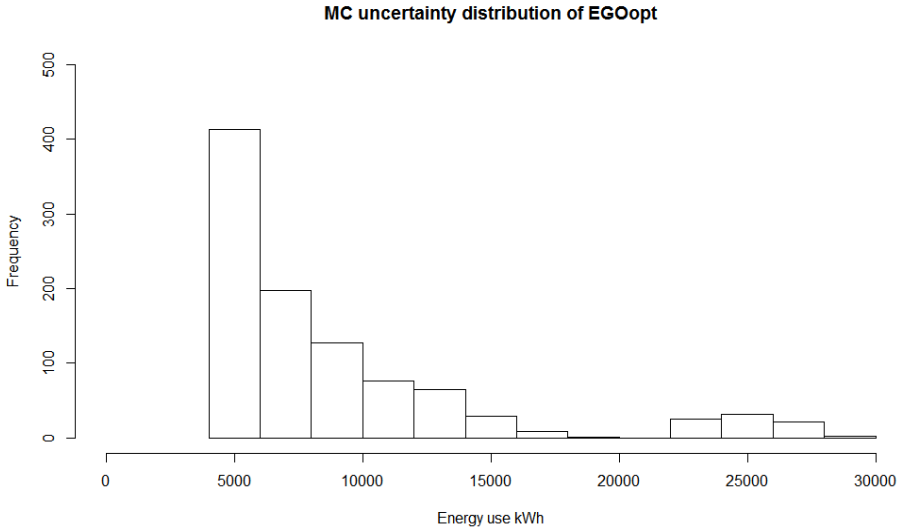


Figure 7-4: Histogram of MC analysis on the global solution (10 optimisation steps, run 1)

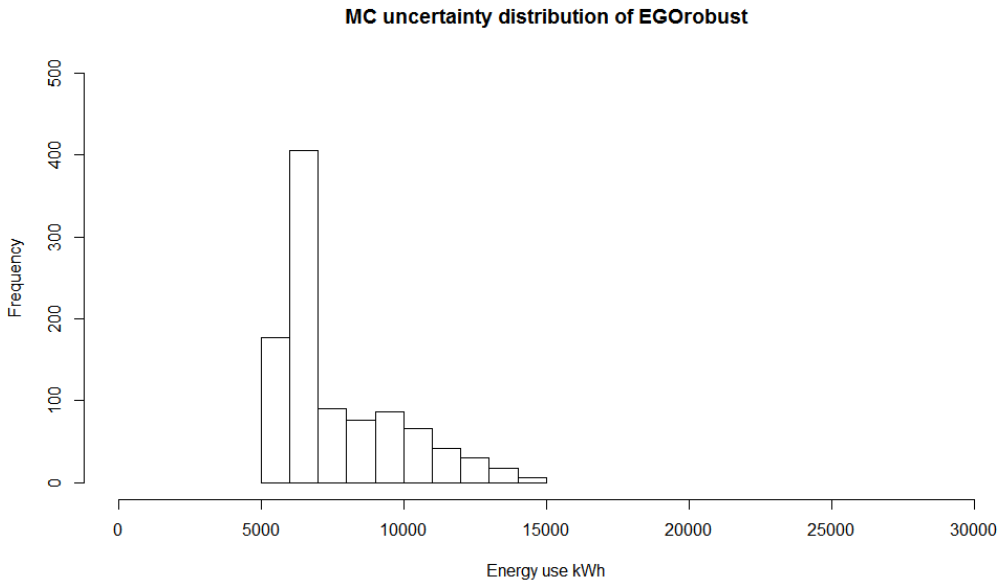


Figure 7-5: Histogram of MC analysis on the robust solution (10 optimisation steps, run 2)

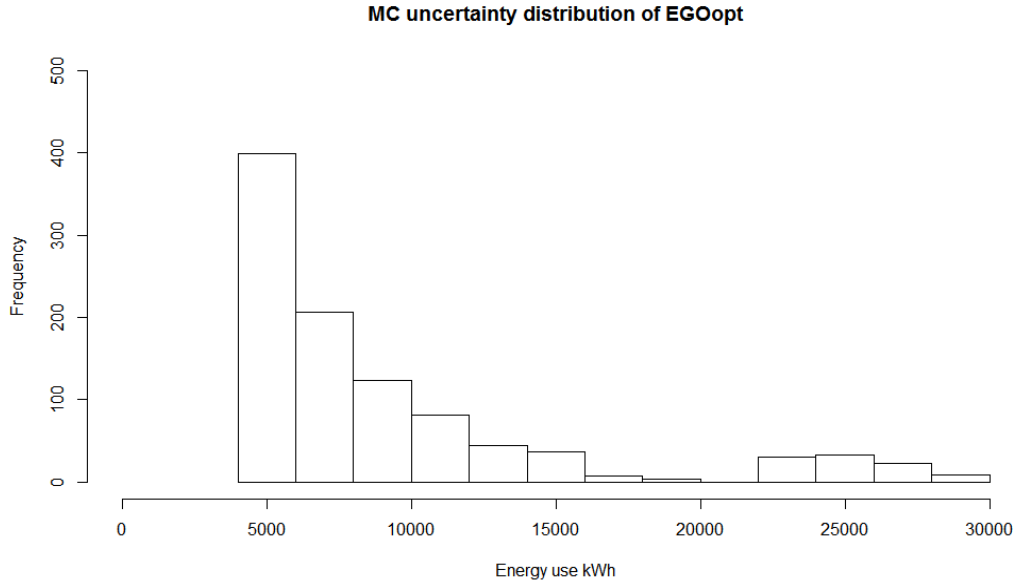


Figure 7-6: Histogram of MC analysis on the global solution (10 optimisation steps, run 2)

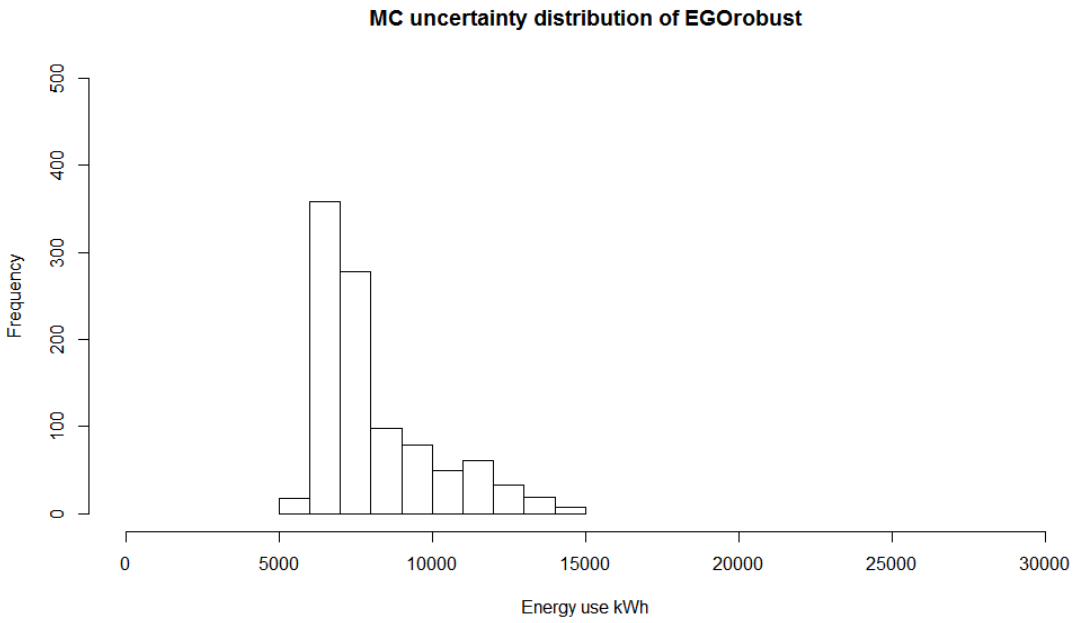


Figure 7-7: Histogram of MC analysis on the robust solution (10 optimisation steps, run 3)

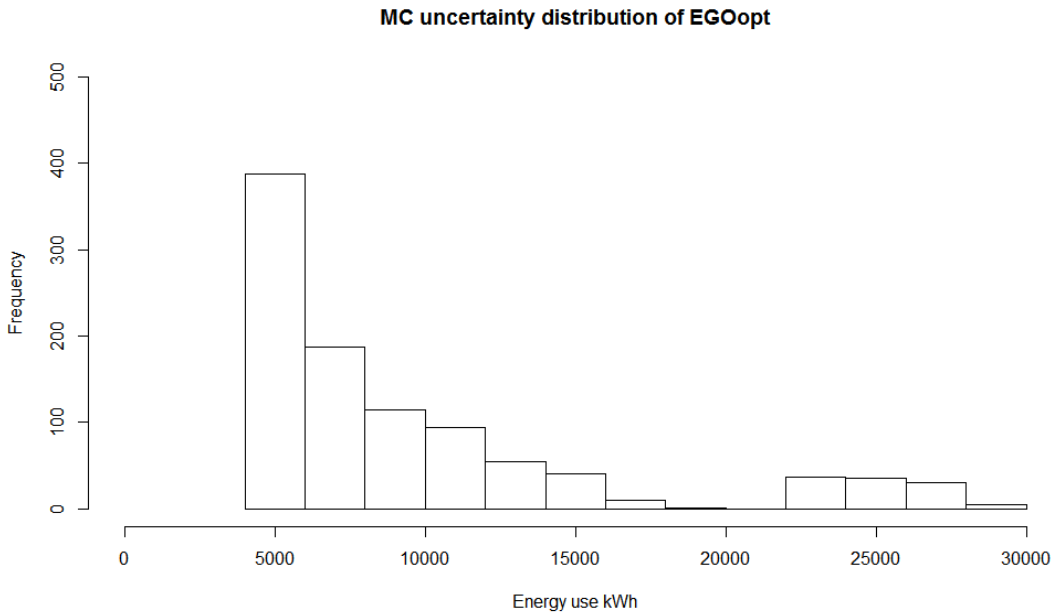


Figure 7-8: Histogram of MC analysis on the global solution (10 optimisation steps, run 3)

7.5.2 20 optimisation steps

The histograms for the EGO and EGOrobust are shown in Figure 7-9 to Figure 7-14. As with the 10-step optimisations, the histograms show the results of the MC analysis on the robust optimisation results (EGOrobust) and the traditional global optimisation results.

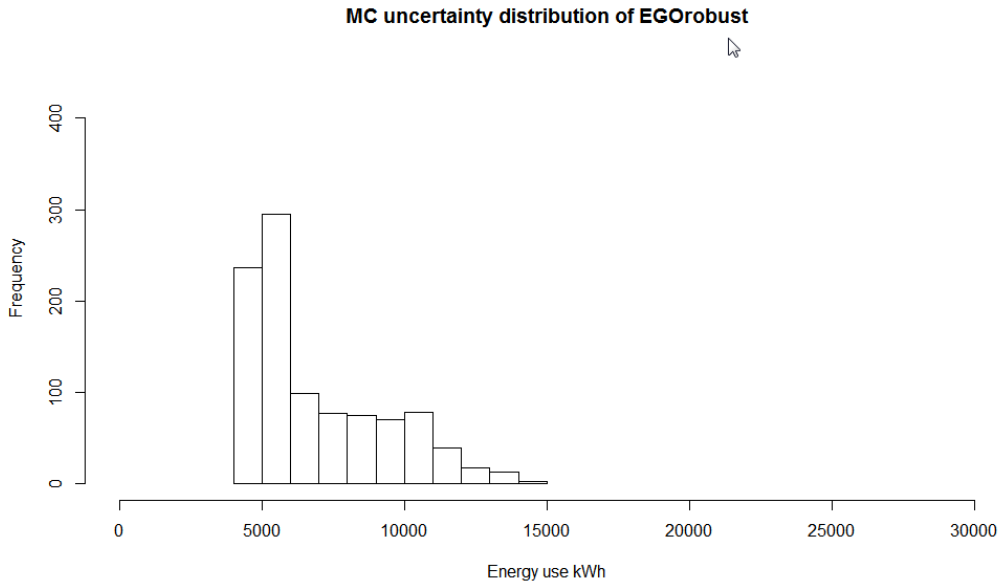


Figure 7-9: Histogram of MC analysis on the robust solution (20 optimisation steps, run 1)

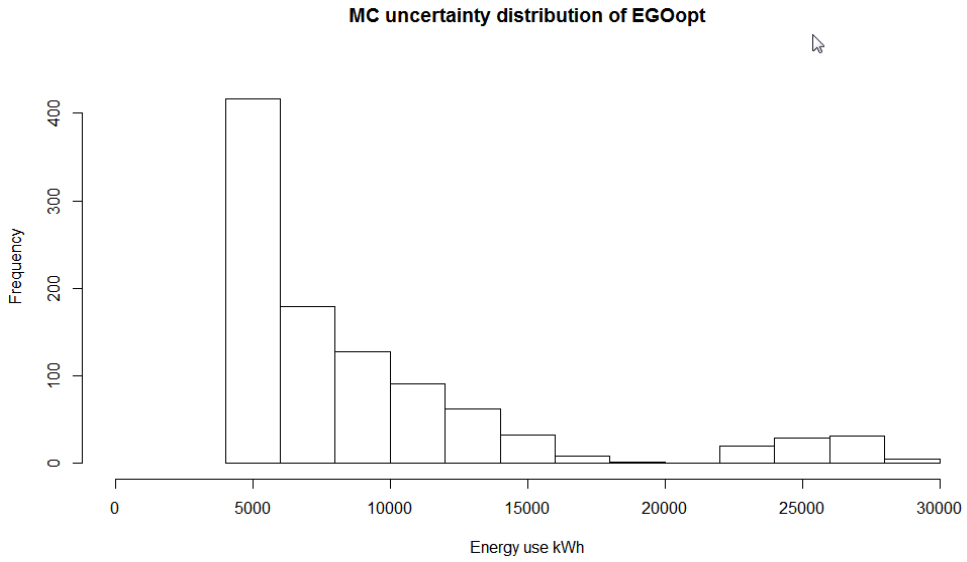


Figure 7-10: Histogram of MC analysis on the global solution (20 optimisation steps, run 1)

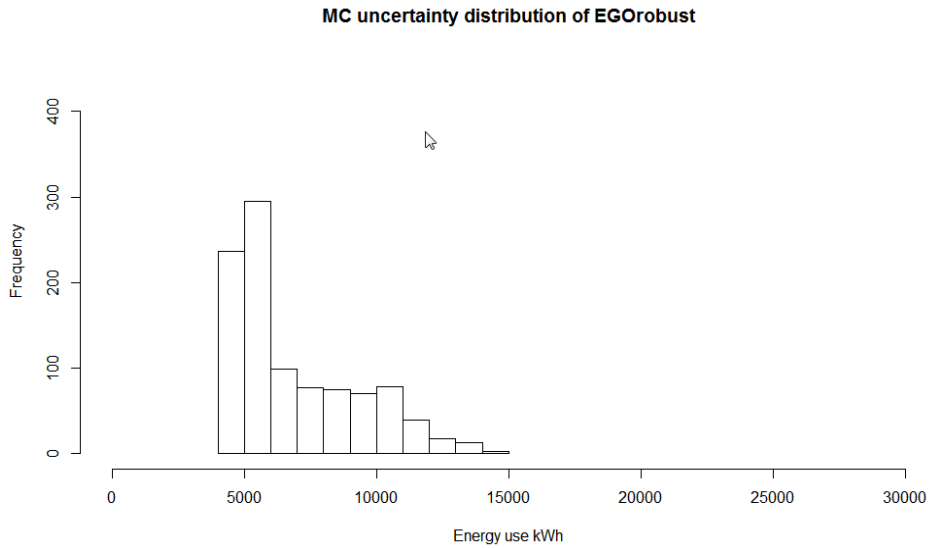


Figure 7-11: Histogram of MC analysis on the robust solution (20 optimisation steps, run 2)

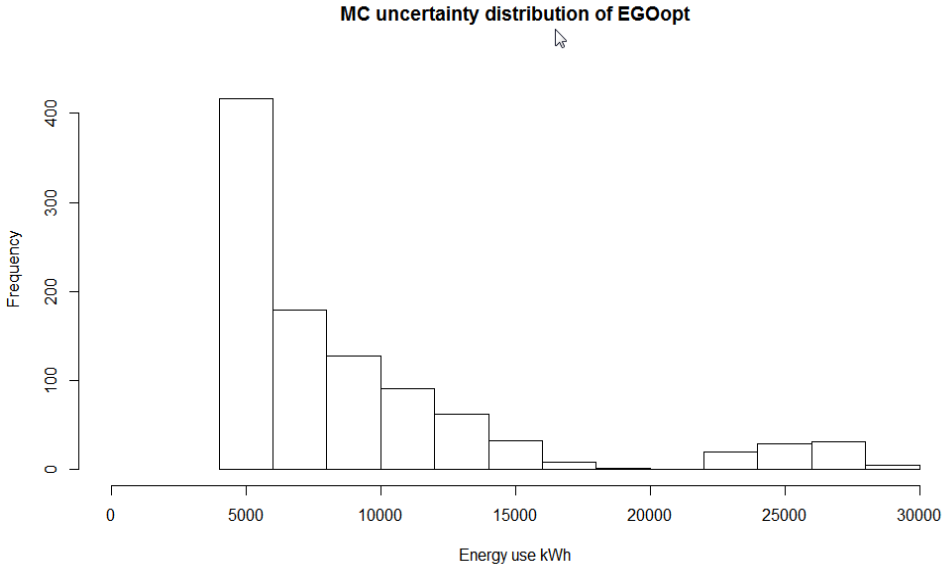


Figure 7-12: Histogram of MC analysis on the global solution (20 optimisation steps, run 2)

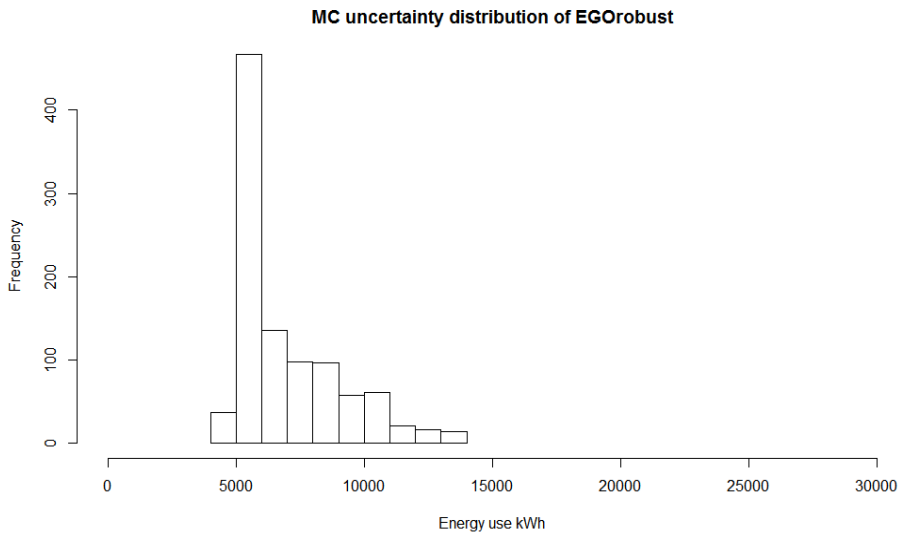


Figure 7-13: Histogram of MC analysis on the robust solution (20 optimisation steps, run 3)

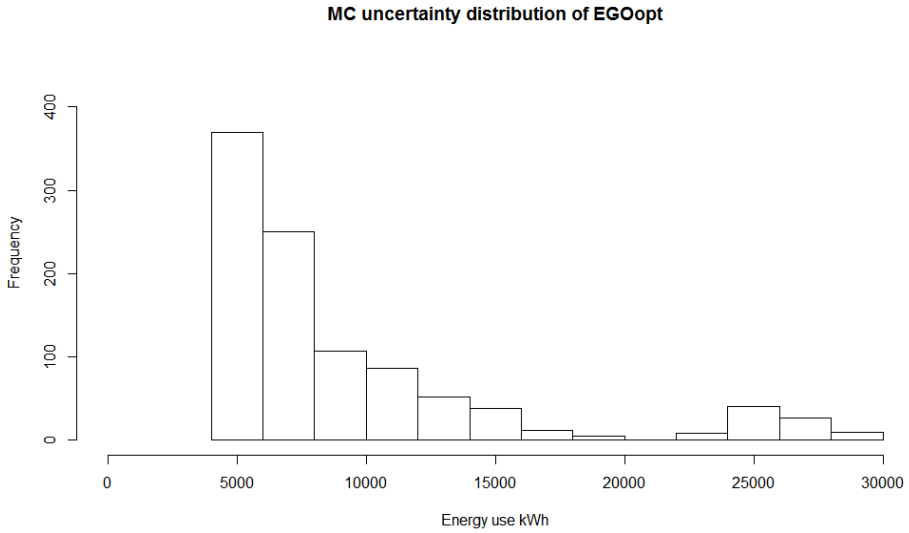


Figure 7-14: Histogram of MC analysis on the global solution (20 optimisation steps, run 3)

7.5.3 Building solutions

Table 7-3 shows the building configurations created by the different algorithms and Table 7-4 shows the configurations obtained by the 20 step optimisation.

Table 7-3: Solutions offered by each of the optimisation algorithms based on 10 steps

	Robust output 1	Optimisation output 1	Robust output 2	Optimisation output 2	Robust output 3	Optimisation output 3
Building x (m)	13.8	7.1	10.0	7.1	11.5	14.1
Building y (m)	7.3	14.1	10.0	14.1	8.7	7.1
Glazing N (%)	46%	10%	54%	29%	23%	29%
Glazing S (%)	36%	10%	59%	10%	34%	10%
Thermal mass (m)	0.022	0.015	0.013	0.195	0.174	0.015
Wall insulation thickness (m)	0.018	0.012	0.100	0.012	0.016	0.098
Roof insulation thickness (m)	0.214	0.293	0.040	0.293	0.272	0.293
Orientation (deg)	77	90	-70	-90	-45	-90
Air permeability (ACH)	0.6	0.7	0.6	0.7	1.5	0.7
Heating set point (deg)	20.0	20.8	20.1	20.8	20.0	20.8

Table 7-4: Solutions offered by each of the optimisation algorithms based on 20 steps

	Robust output 1	Optimisation output 1	Robust output 2	Optimisation output 2	Robust output 3	Optimisation output 3
Building x (m)	12.0	11.4	14.1	10.4	14.1	14.1
Building y (m)	8.3	8.8	7.1	9.6	7.1	7.1
Glazing N (%)	18%	10%	46%	10%	15%	10%
Glazing S (%)	53%	10%	35%	10%	31%	10%
Thermal mass (m)	0.149	0.015	0.185	0.195	0.155	0.195
Wall insulation thickness (m)	0.087	0.012	0.016	0.012	0.087	0.012
Roof insulation thickness (m)	0.219	0.293	0.292	0.293	0.254	0.293
Orientation (deg)	2	0	14	-90	-81	-90
Air permeability (ACH)	0.7	0.7	0.8	0.7	1.2	0.7
Heating set point (deg)	20.3	20.8	20.1	20.8	20.1	20.8

7.6 Discussion

In both the 10 and 20-step cases, the distributions shown in the histograms clearly show that the EGOrobust optimiser uncovers a much more robust solution than the EGOopt optimiser. This is a positive indication that the EGOrobust optimiser has been more successful in uncovering a robust solution. Furthermore, the results of the 10 and 20-step optimisation processes show very similar distributions in the output when presented with the uncertainty set. This suggests that the adding having 20 steps in the optimisation process does not add any significant improvement over the 10 step solutions. More work is needed to see if 10 steps is sufficient for other building optimisation problems.

However, even though each of the EGOopt histogram outputs are very similar and each of the EGOrobust outputs are also similar, I might expect that the building designs found are also similar. However, this was not the case. Interestingly, the robust optimisation tool tended to uncover results with more glazing than the EGOopt tool. This was true of both the 10-step and 20-step optimisation. In all cases, the EGOopt optimiser resulted in southerly glazing of 10%, which was the minimum possible.

In setting the uncertainty levels for the EGOrobust and the subsequent MC analysis on the uncertainty set, some variables had higher levels of uncertainty than others. This was intentional, since there are variables that I can expect to change a lot (such as the number of residents and the heating set point), and variables that I would not expect to change much, or that I am more certain about (such as the orientation and the glazed fraction). I attributed the biggest uncertainties to the number of residents (±2) and the heating set-point (±2.7).

This means that the results should be robust to the difference occupancy patterns and heating set-point, since their full-scale ranges were 5 occupants and 6 degrees respectively. From examining the individual solutions and the building histograms, I might conclude that it is important to ensure that each of the buildings has enough glazing to ensure robustness. Although the actual robust *designs* are expected to use more energy, they are *less likely* to use more energy overall using the uncertainties considered.

The building solutions are different across both the global optima and robust optima (Table 7-3 and Table 7-4), however there both differences and similarities between the global and robust solutions. The robust solutions generally have more glazing than the traditionally optimised solutions, with generally more glazing on the north façade. Nearly all of the traditionally optimised solutions minimise the glazing. Other parameters, such as the aspect ratio, the thermal mass and orientation vary quite considerably for both the robust and non-robust solutions. Neither the robust or non-robust solutions appear to favour any particular values in for these parameters. For example, the orientation varies considerably across all solutions, as does the thermal mass.

Despite the differences between the solutions, there are some commonalities. The heating set point is kept low in all solutions (as might be expected) and the air permeability is low in nearly all cases. The roof insulation is above 200 mm in nearly all of the solutions (with wall insulation appearing more variable across solutions).

In terms of computation time, the time taken to run the training simulations and subsequent optimisation steps is many orders of magnitude greater that either

the time taken to create the emulators, or the time taken to run the MC analysis on the emulator. Also interesting to note is the fact that the robust optimisation algorithm does not require an increased number of *simulations* to arrive at its results.

7.7 Conclusions

I have shown that it is possible to build an effective robust optimiser using Kriging methods, with only a negligible increase in computer power compared to traditional Kriging-based EGO. Both the EGOopt and EGOrobust methods have considerable advantage over GA optimisation methods. I have shown in chapter 3 how EGOopt can be more efficient than the GAs alone. The EGOrobust method further extends this by adding the ability to measure the predicted robustness.

However, as with the EGOopt method, there are significant areas for further work in developing RO techniques using Kriging. I highlight three key areas where I believe that further work should be focused.

1. The development of convergence criteria for EGOrobust.
2. Testing the algorithm on a wide range of building types and building problems.
3. Developing suitable uncertainty sets and robust criteria

The development of a reliable convergence criterion for EGOrobust is an essential step towards building an optimisation tool that can have impact in the commercial environment. There are a number of challenges in this, including deciding what the convergence criterion should be, what value it should take for certain building problems and if there is any relationship between it and the uncertainties being studied.

It is also important to test and develop the EGOrobust method on a wide variety of building types and building problems. This will require extensive computing power, since the EGOrobust results will need to be compared to brute-force MC simulations for each building type.

The final area of further work will be to focus on the uncertainty set U that is used to build the worst-case emulator. In our example I take a 1000-sample set for the MC sampling used to generate this emulator. I then take the 95^{th} percentile result as the worst-case output used to create the worst-case emulator. I suggest that finding a good balance between the sampling used for U, the number of MC simulations (i.e. the number of elements in U) and the percentile used is a useful area for further research.

7.8 Summary

In the previous chapters, I have implemented Kriging directly as an optimisation and uncertainty analysis tool and have compared the results to those obtained by GAs and brute-force methods. In the final chapter, I investigate a hybrid Kriging-GA-based method for multi objective optimisation based on the widely used non-dominated sorting genetic algorithm II (NSGA-II).

Chapter 8 Multi-objective optimisation using Kriging

8.1 Introduction

So far I have explored various methods for optimising buildings using Kriging methods, however, these methods have only focused on single objective optimisation. Although such methods are useful for some design purposes, it is frequently the case the designers are trying to make decisions to maximise more than one feature of the building, which often have competing requirements.

Two examples of these competing requirements are maximising the natural light within a building whilst minimising the amount of energy the building uses. The competing nature of these requirements is due to the fact that, to increase daylight, more windows are required, however, windows are also responsible for most of the heat loss in a building, and therefore (potentially) lead to greater energy use.

There are a number of multi-objective optimisation (MOO) methods that are currently available, which I will briefly review. I will then focus on the implementation of a hybrid Kriging-GA method for performing a multi-objective optimisation, based on previous work with GAs and radial basis function by Brownlee and Wright [66].

8.2 Background

As discussed in chapter 1, there are a variety of multi-objective optimisation methods. Commonly used multi-objective optimisation methods that are used in

building energy modelling include particle swarm [162],[163], the Hooke-Jeeves algorithm [75], [163] and simulated annealing [7]. However, the most commonly used optimisation approaches in building design use some form of evolutionary algorithm [71]. In this chapter, I focus on a development to NSGA-II.

NSGA-II is a genetic algorithm-based approach that uses a non-dominated sorting method to create Pareto fronts for multiple objectives. In examining its characteristics, it is useful to consider the key concepts of multi-objective optimisation, which include non-domination, diversity and the dominated hypervolume of a Pareto front. I will address these separately.

8.2.1 Non-domination and dominated zones

A non-dominated point is one which is better than all other points when considering one or more output objectives. Figure 8-1 shows an example of two non-dominated points and one dominated point.

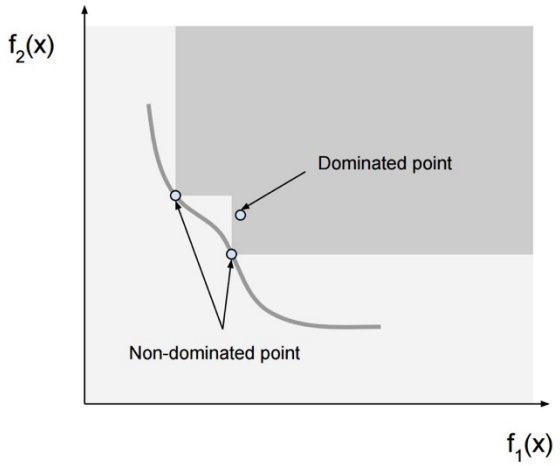


Figure 8-1: Two non-Pareto dominated points

8.2.2 Diversity

In general, Pareto fronts are commonly generated so that a designer can pick from one of the available designs. It is therefore desirable that there a number of diverse designs to choose from. Figure 8-2 compares a diverse Pareto front to one which is less diverse.

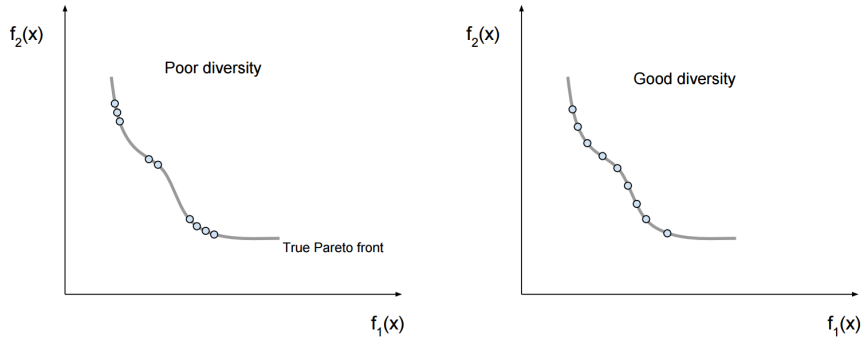


Figure 8-2: Demonstration of diversity in Pareto optimal frontiers

A good algorithm for generating Pareto optimal solutions should therefore create and maintain diversity in its non-dominated solutions.

8.2.3 Dominated hypervolume

In comparing the effectiveness of algorithms, it is also useful to be able to have a method for comparing the potential outputs. One way of doing this is to compare the *dominated hypervolume*. In a two-dimensional output space, this is the area in which any point would be 'dominated' by the Pareto front. Figure 8-3 illustrates a dominated hypervolume for a simple problem.

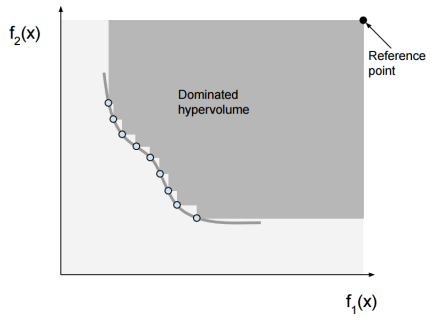


Figure 8-3: Example of the dominated hypervolume of Pareto optimal set

8.2.4 NSGA-II

NSGA-II is an efficient and popular algorithm for multi-objective optimisation. It has been shown to have good convergence properties as well as producing diverse output populations [21]. NSGA-II has been applied widely in building optimisation research [58], [63]–[65] as well as in commercial optimisation products [19].

NSGA-II is one of the most commonly used algorithms in building optimisation. We have seen that it is commonly applied in many commercially available optimisation routines.

NSGA-II is based around the typical operations found in other genetic and evolutionary algorithms. These operations are referred to as *mutation*, *crossover* and *tournament selection* [21]. To make these descriptions relevant to building design, we can think of 'populations' as being a group of candidate building designs. Individuals within a population are, likewise, individual building designs.

Genetic operators used in NSGA-II

The initial phase of NSGA-II is the generation of a new offspring population (**Q**) from an initial parent population (**P**). The offspring population **Q** is created by

performing a number of ‘genetic’ operations on **P.** These genetic operations are *tournament selection*, *mutation,* and *crossover*.

The purpose of tournament selection is to create a ‘mating pool’ from which the offspring population **Q** will be generated. This is achieved by:

1. Randomly selecting *n* samples from the parent population (where *n* is the *tournament size*)
2. Add the *n* samples to the *mating pool.*
3. Repeat steps 1 and 2 until size of *mating pool* is equal in size to **P**.

The tournament selection phase produces the initial offspring population. However, this population contains replicas of the population **P**. In order to generate new solutions, the new population is ‘evolved’ through *crossover* and *mutation*.

The crossover phase allows the swapping of genetic information between members of the mating pool. ‘Parent’ chromosomes from the mating pool are selected at random for crossover as are the specific *genes* on the chromosome of each parent. Figure 8-4 shows the crossover of genetic information on the third gene of two parents from the mating pool:

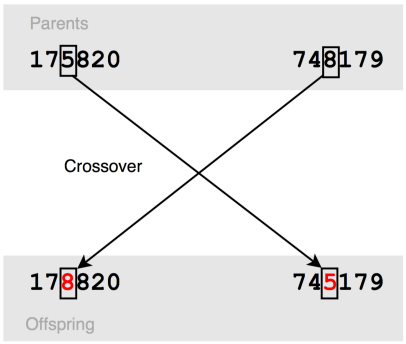


Figure 8-4: Crossover operation

Mutation is an operation where one (or more) of the elements of a gene are changed at random. The location of the change and its magnitude are chosen at random. Figure 8-5 illustrates a mutation of magnitude 2 on the third gene to two parents in the mating pool.

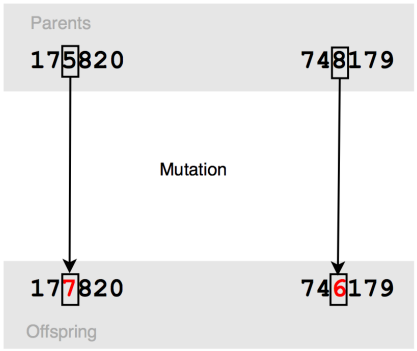


Figure 8-5: Mutation operation

Process of NSGA-II

So far I have described the process for generating an offspring population **Q** from a parent population **P**. The offspring in **Q** will be different from the parent population because of the mutation and crossover operations that they have undergone. The steps described so far constitute steps 1 and 2 of the 4-step process shown in Figure 8-6.

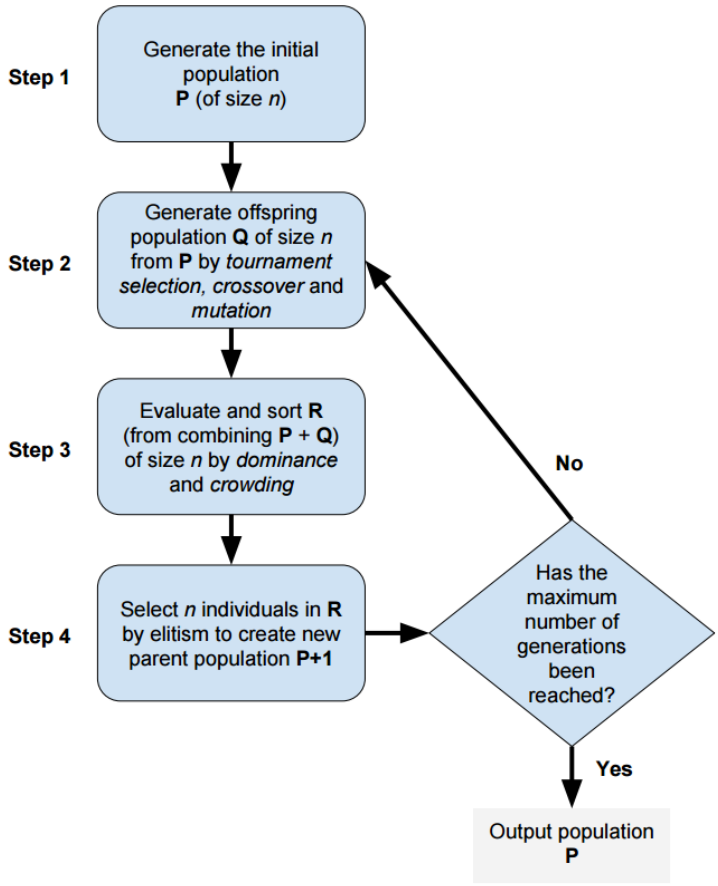


Figure 8-6: Steps in the NSGA-II algorithm

Step 3 involves sorting the combined parent and offspring populations (**P+Q**) and selecting the best candidates. The candidates in **P+Q** are ranked according to both their *dominance* and *crowding.*

As is shown in Figure 8-1, points that are 'dominated' by better solutions should be rejected where possible. Also, solutions that are very close to each other (i.e. where the results are '*crowded'*) should also be deemed less important, therefore retaining diversity in the results. In NSGA-II, these objectives are achieved through *non-dominated sorting* and the calculation of the *crowding distance.* This process is shown graphically in Figure 8-7.

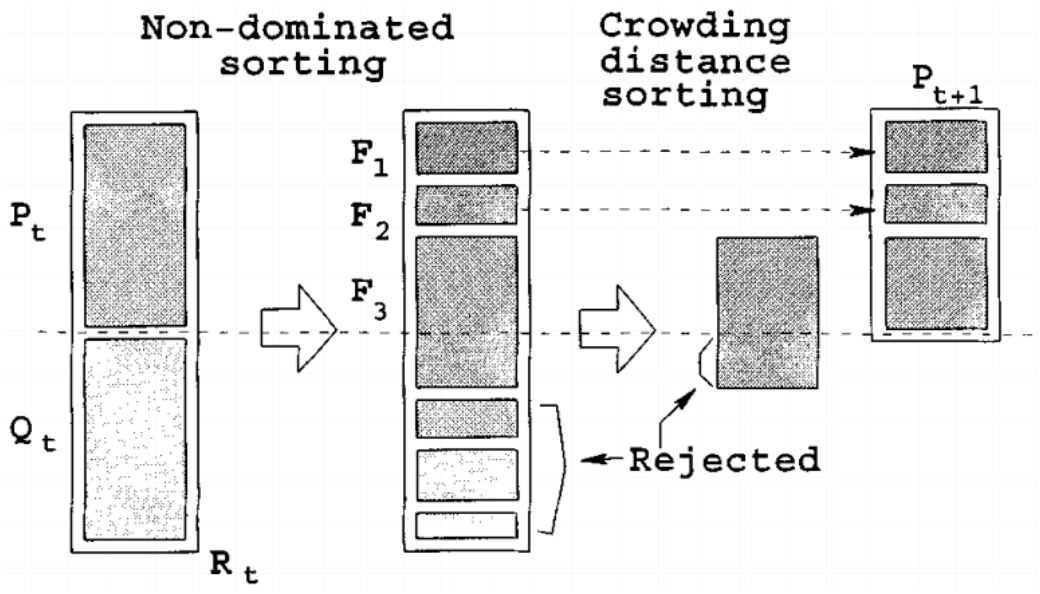


Figure 8-7: NSGA-II procedure (from Deb et al. [164]*)*

Figure 8-7 shows the parent population **P** and its offspring **Q** are combined and then undergo *non-dominated sorting*. The second column shows the combined population ranked by dominance (**F**). In the example above, F_3 (the population group ranked third for dominance) are sorted according to their crowding distance. The lowest portion of this group is rejected so that the subsequent population $\mathbf{P_{t+1}}$ retains the same population size as the initial population.

As shown in Figure 8-6, the whole process is repeated until a set number of generations is reached. The population surviving these generations is the solution.

8.3 Aims

The most computationally expensive aspect of NSGA-II is the evaluation of the function output for each new offspring population **Q**. I aim to improve the effectiveness of the NSGA-II by evaluating every alternate offspring population with an *emulator* instead of the *simulator* function. This builds on the work started

by Brownlee and Wright by using both simulation and emulation to create the parent population and is intended to speed up the rate of convergence. Brownlee and Wright have used radial basis functions to build the emulator, whereas we aim to test a similar solution using Kriging emulators.

8.4 Method

NSGA-II it is a multi-objective optimisation algorithm based on genetic algorithms. It is intended to provide a range of design solutions for competing objectives. We can express that problem mathematically as:

$$\min(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}) \dots f_M(\boldsymbol{x})) \qquad \text{Eq. 36}$$

Where $f_n()$ are the different objective functions to be minimised, $\boldsymbol{x}$ is a vector of parameters representing the building design, and M is the number of objectives to be minimised.

8.4.1 Augmenting NSGA-II with Kriging

In my work, I augment the NSGA-II algorithm with Kriging emulation to minimise two competing building objectives. Since the algorithm starts with a parent population **P**, which is immediately evaluated by the simulator, I can use this data to build an emulator. At the evaluation stage, I can also improve the emulator with the extra evaluations of $\mathbf{Q}$, which is required before the ranking and sorting of the combined parent-offspring population $\boldsymbol{R}$.

The emulator can then be used in the creation of the offspring population $\boldsymbol{Q}$. This creates an extra step in the analysis whereby I create an offspring population k-times bigger than $\boldsymbol{P}$, which I shall term $\mathbf{S}$[11]. I create the offspring population S using the same tournament selection, crossover and mutation process used in the original algorithm. However, instead of evaluating the $\mathbf{S}$ population with the simulator before I sort and rank the results, I evaluate them with the *emulator*. The results are still sorted and ranked in the same way, but this time the dominance and crowding values are based on the results of the emulator, not the simulator.

Following the sorting and ranking of S, I then select the best n results from S to form the *offspring* population $\boldsymbol{Q}$. It is here where I re-join the original NSGA-II algorithm at step 3. $\boldsymbol{Q}$ is evaluated with the simulator and joined with the original parent population $\boldsymbol{P}$ to create $\boldsymbol{R}$. The combined population $\boldsymbol{R}$ is then ranked and sorted, and the best n individuals are then selected by environmental selection. This concludes a single generation.

In addition to the steps described above, it is important to note that I update the Kriging meta-model every time I have to evaluate Q. In this way the emulator should improve in accuracy on each generation. This improvement should lead to the selection of a better Q the next time round, since the *emulated* evaluation of S should become increasingly accurate, and hence the sorting of the result of S should behave more and more like I am evaluating them with the simulator.

[11] In all the analyses we discuss in this chapter, we use $k = 3$

8.4.2 Algorithms tested

The final element to be considered is the selection of the initial population. We have seen that, in the creation of the Kriging emulator, the way of selecting the initial training set is important, with Latin Hypercube sampling being one of the most favoured methods. However, most NSGA-II algorithms typically use a random-start. I therefore test both these approaches when testing the augmented NSGA-II algorithm. I tested three types of NSGA-II, each with the following configurations:

- **NSGA-II** - This is the standard NSGA-II algorithm (including random start)
- **NSGA-II+KRS** - This the augmented algorithm with Kriging and a random start
- **NSGA-II+KLHS** - This the augmented algorithm with Kriging starting with *maximin* Latin hypercube sampling.

8.4.3 Approach to testing the NSGA-II algorithms

In order to test the effectiveness of the NSGA-II, NSGA-II+KRS and NSGA-II+KLHS, these algorithms are then applied to the building model, as well as a number of test functions. I first review the test functions used and the reasons for choosing them. This is followed by a brief description of the test setup and the results of the test modelling. I then go on to describe the building model used and show how the each of the NSGA-II algorithms were applied to it. I then show and discuss the results of the building model. Finally, I discuss the results of both the test functions and the building model and follow this with proposals for future work.

8.4.4 Test functions

The performance of NSGA-II was first compared to the performance of NSGA-II with Kriging using a number of test functions. The test functions used are multi-output functions that are in common use in the test literature [21],[165]. These functions were chosen for their diversity in their Pareto front solutions. The functions include those with connected and disconnected Pareto fronts (i.e. those where the solutions lie on one continuous Pareto front, and those that don't), as well as those with convex and non-convex solutions (i.e. those where the Pareto front solution is convex, and those where it is not).

Deb 3

Variable bounds (lower, upper): [0,1]

$g(x) = 4 - 3\exp(-50(x-0.2))^2$ for $x \leq 0.4$ or

$g(x) = 4 - 2\exp(-5(x-0.7))^2$ for $x > 0.4$

$h(a,b) = 1 - \left[\frac{a}{b}\right]^{\{(0.25 + 3.75(b-1))\}}$ for $a \leq b$ or

$h(a,b) = 0$ or $a > b$ Eq. 37

$f_1(x) = 4x_1$

$f_2(x) = g(x_2) \times h(f_1(\boldsymbol{x}), g(x_2))$

(number of inputs $n = 2$)

Fonseca II (non-convex)

Variable bounds (lower, upper): [-4,4]

$$f_1(\boldsymbol{x}) = 1 - \exp(-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2)$$

$$f_2(\boldsymbol{x}) = 1 - \exp(-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2) \qquad \text{Eq. 38}$$

where the solutions are $x_1 = x_2 = x_3$ $(n = 3)$

ZDT1 (convex)

Variable bounds (lower, upper): [0,1]

$$f_1(\boldsymbol{x}) = x_1$$

$$f_2(\boldsymbol{x}) = g(\boldsymbol{x})[1 - \sqrt{\frac{x_1}{g(x)}}] \qquad \text{Eq. 39}$$

$$g(\boldsymbol{x}) = 1 + 9\,\frac{\sum_{i=2}^{n} x_i}{n-1}$$

where the solutions are: $x_1 \in [0,1]$, $x_i = 0$, and $i = 2, \ldots, n$ $(n \leq 30)$

ZDT2 (non-convex)

Variable bounds (lower, upper): [0,1]

$$f_1(\boldsymbol{x}) = x_1$$

$$f_2(\boldsymbol{x}) = g(\boldsymbol{x})[1 - \left(\frac{x_1}{g(x)}\right)^2] \qquad \text{Eq. 40}$$

$$g(\boldsymbol{x}) = 1 + 9\,\frac{\sum_{i=2}^{n} x_i}{n-1}$$

where the solutions are: $x_1 \in [0,1]$, $x_i = 0$, and $i = 2, \ldots, n$ ($n \leq 30$)

ZDT3 (convex, disconnected)

Variable bounds (lower, upper): [0,1]

$$f_1(\boldsymbol{x}) = x_1$$

$$f_2(\boldsymbol{x}) = g(\boldsymbol{x})\,\{\,1 - \sqrt{\frac{x_1}{g(\boldsymbol{x})}} - \frac{x_1}{g(\boldsymbol{x})}\sin(10\,\pi\,x_1)\,\} \qquad \text{Eq. 41}$$

$$g(x) = 1 + 9\,\frac{\sum_{i=2}^{n} x_i}{n-1}$$

where the solutions are: $x_1 \in [0,1]$, $x_i = 0$, and $i = 2, \ldots, (n \leq 30)$

8.4.5 Analysis of the test functions

I used slightly different test setups for each of the test functions and for testing the building function. The setup of each of these functions is shown in Table 8-1.

Table 8-1: NSGA-II test setup

Method	Repetitions of algorithm	Generations	Kriging population ratio k
NGSA-II	10	20	-
NGSA-II+KRS	10	20	3
NGSA-II+RLHS	10	20	3

The Kriging model was set up to produce populations of size $|S|$ that are $k = 3$ times the original population size of the parent population. However, the initial population size is dependent on the function under test. This is because the *ZDT* family of functions each have 30 input parameters, whereas the *Deb3* and *Fonseca* functions only have two input parameters. Therefore, for the *ZDT* functions, I use a population of 60, and for the *Deb3* and *Fonseca* functions I use a population of only 20.

The evolutionary settings for the GA are the same for all test functions (as well as the building optimisation described later). Table 8-2 shows these settings which are taken from Tresidder [15].

Table 8-2: GA setup for the NSGA-II analysis

Setting	Value
Crossover probability	0.5
Mutation probability	0.4
Tournament size	5

8.4.6 Analysis of the building model

To test NSGA-II+KRS, NSGA-II+KLHS and NSGA-II on the building model, I looked at two competing performance factors; heating energy use (kWh) and lighting (kWh).

As with the building models in the previous chapters, the heating energy use of the building was calculated annually. To predict the lighting energy use, I added some simple lighting control logic to the building. The lighting was set to operate when;

- the house is occupied[12]; and
- when the lighting level at the control point falls below 300 lux (0700hrs – 2300hrs).

The location of the lighting control element is shown below.

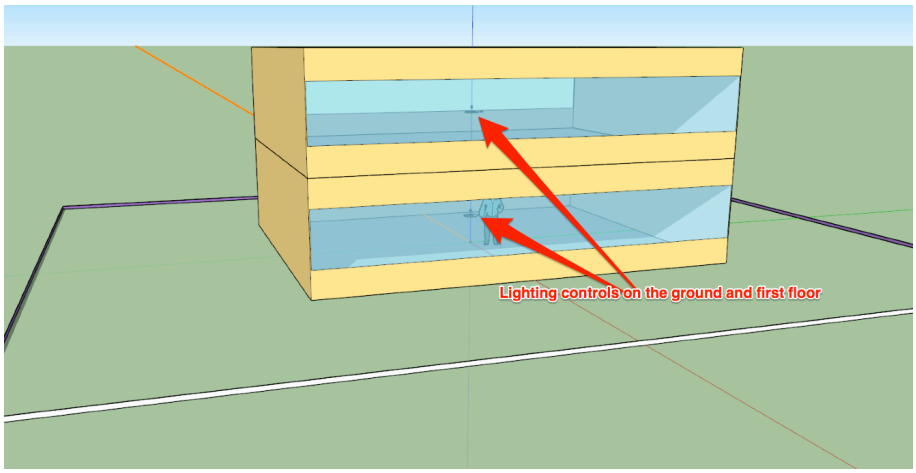


Figure 8-8: Building to be optimised in the NSGA-II-based algorithms

The construction and general set of the building is as described in Chapter 5.

[12] We assume in this case that the number of residents is constant at 4

The test regime for the building model was based on a simulator budget of 400 simulations for each routine, which was divided up as follows (Table 8-3). The setup for the GA parameters remains the same as that shown in Table 8-2.

Table 8-3: Test setup for the various NSGA-II test methods

Method	No. vars.	Max Gens.	Population
NSGA-II	9	20	20
NSGA-II+KLHS	9	20	20
NSGA-II+KRS	9	20	20

8.5 Results

8.5.1 Test functions results

The following figures (Figure 8-9 to Figure 8-20) how the results of the analysis. I am repeating each algorithm 10 times, since this allows us to statistically analyse the results. As described previously, one of the most useful indicators of the development of the Pareto front is the hypervolume. Typically, the hypervolume should increase as the algorithm progresses, representing an ever increasing dominated region. Since each iteration of the algorithm yields a different result, it is useful to statistically analyse the progression of the hypervolume on subsequent iterations. Furthermore, since I repeat each algorithm, I can statistically analyse the results at each iteration by examining the mean and the standard deviation of the hypervolume at each point. The following figures show the results for each test function in two parts. The first set of graphs illustrates how the hypervolume increases for each individual implementation of the algorithm. The second set of graphs illustrates the mean and standard

deviation of all the individual implementations, and therefore allows us to manage the overall performance and stability of each of the algorithms.

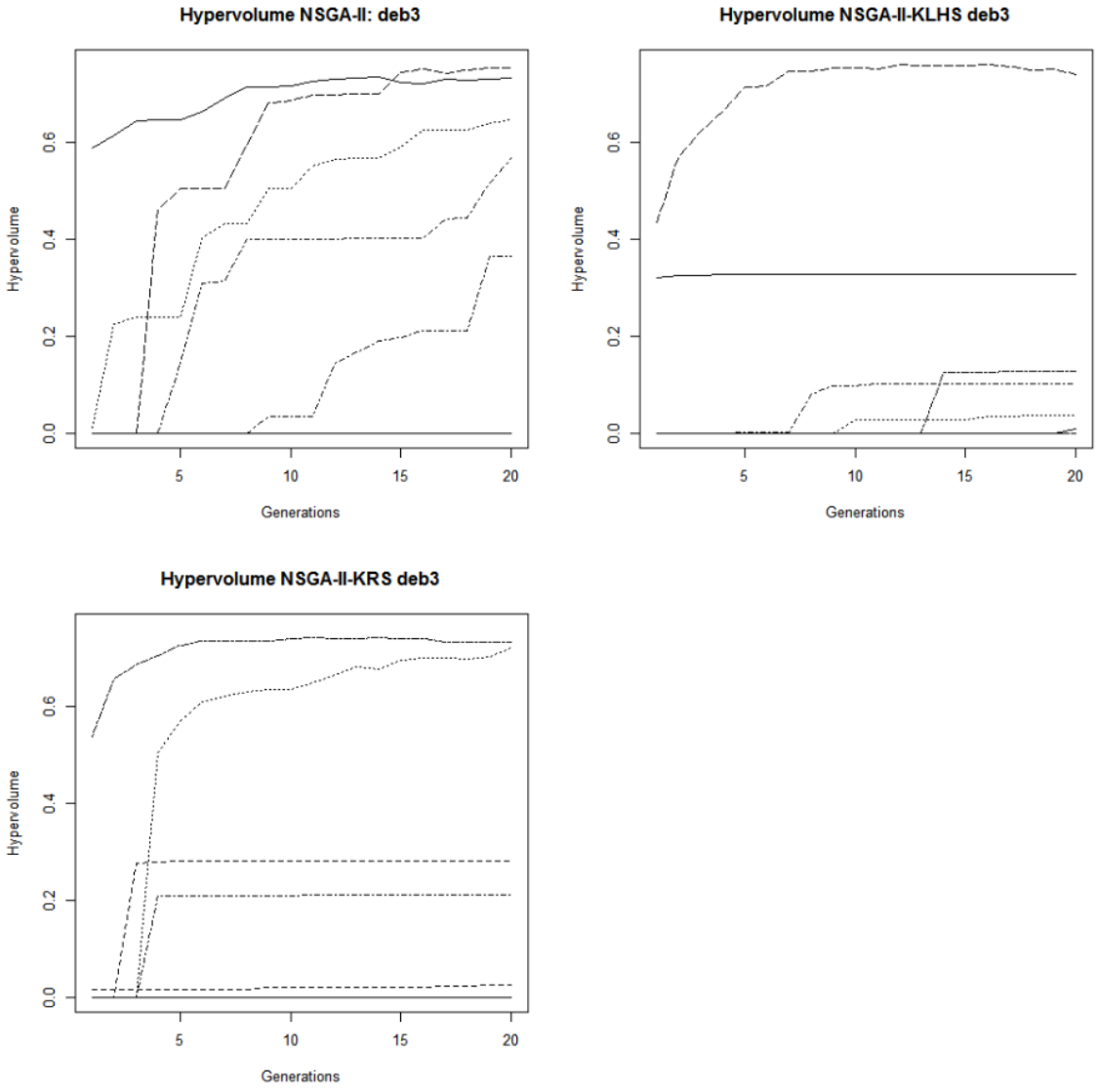
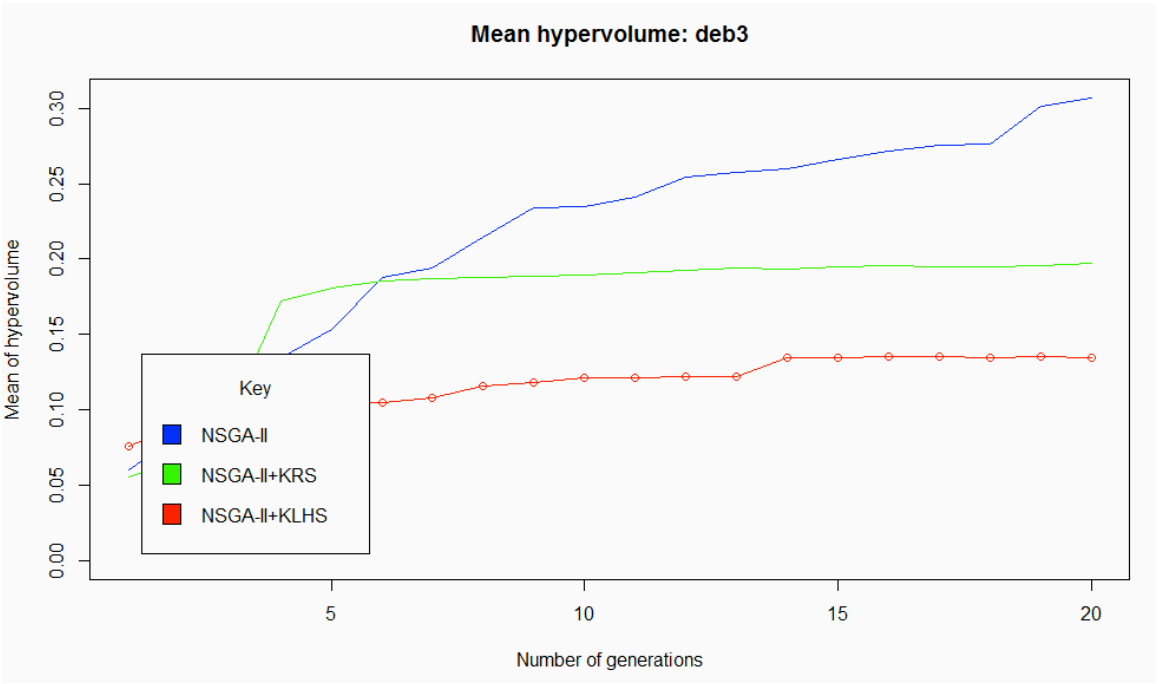


Figure 8-9: Progressions of the hypervolume of each evolution (Deb 3)



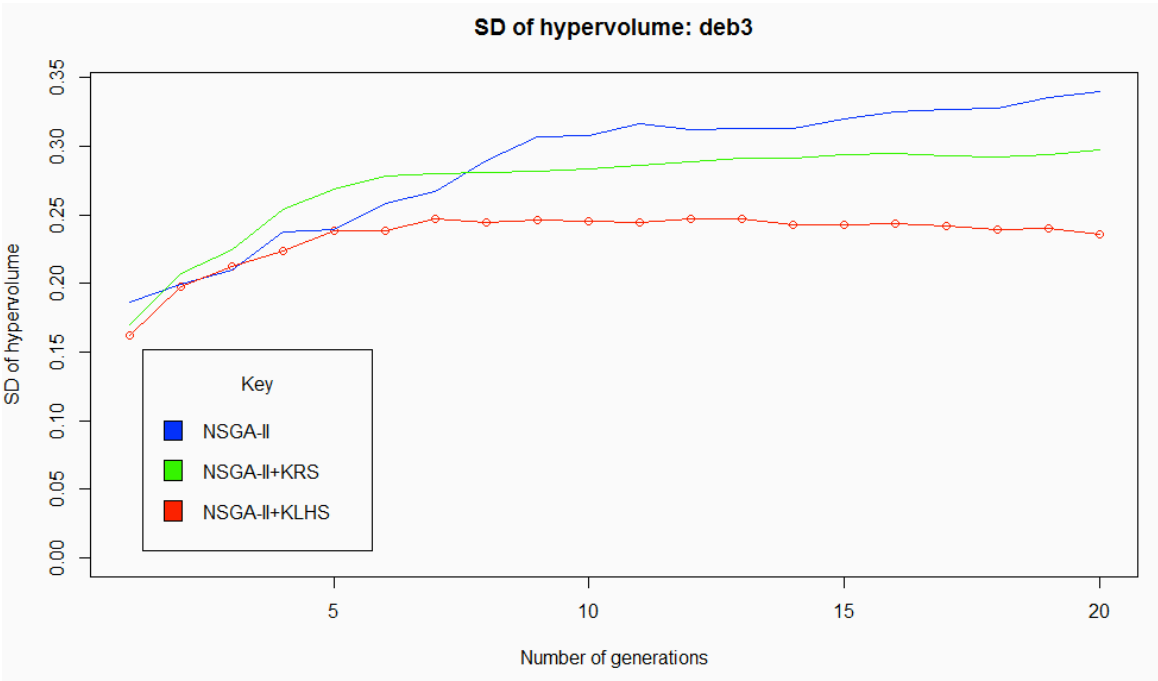


Figure 8-10: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for Deb3

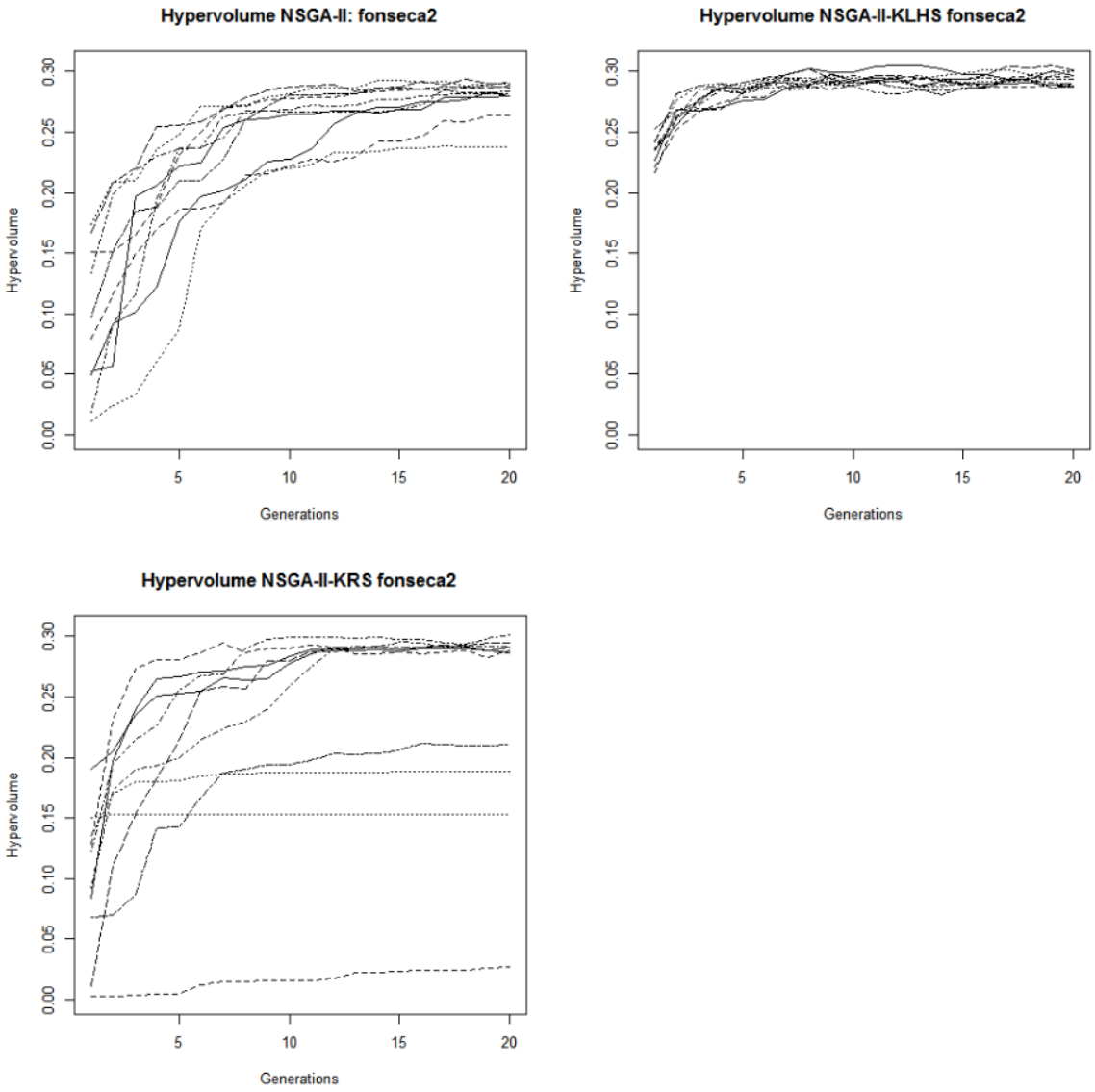
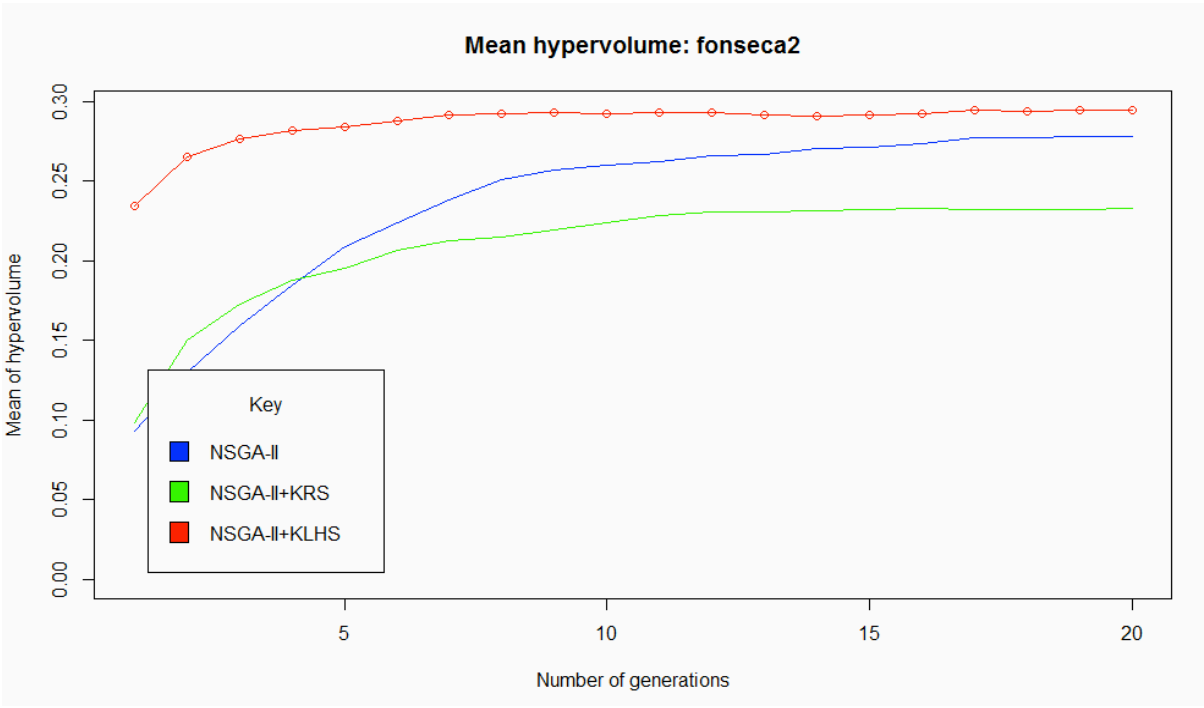


Figure 8-11: Progressions of the hypervolume of each evolution (Fonseca 2)



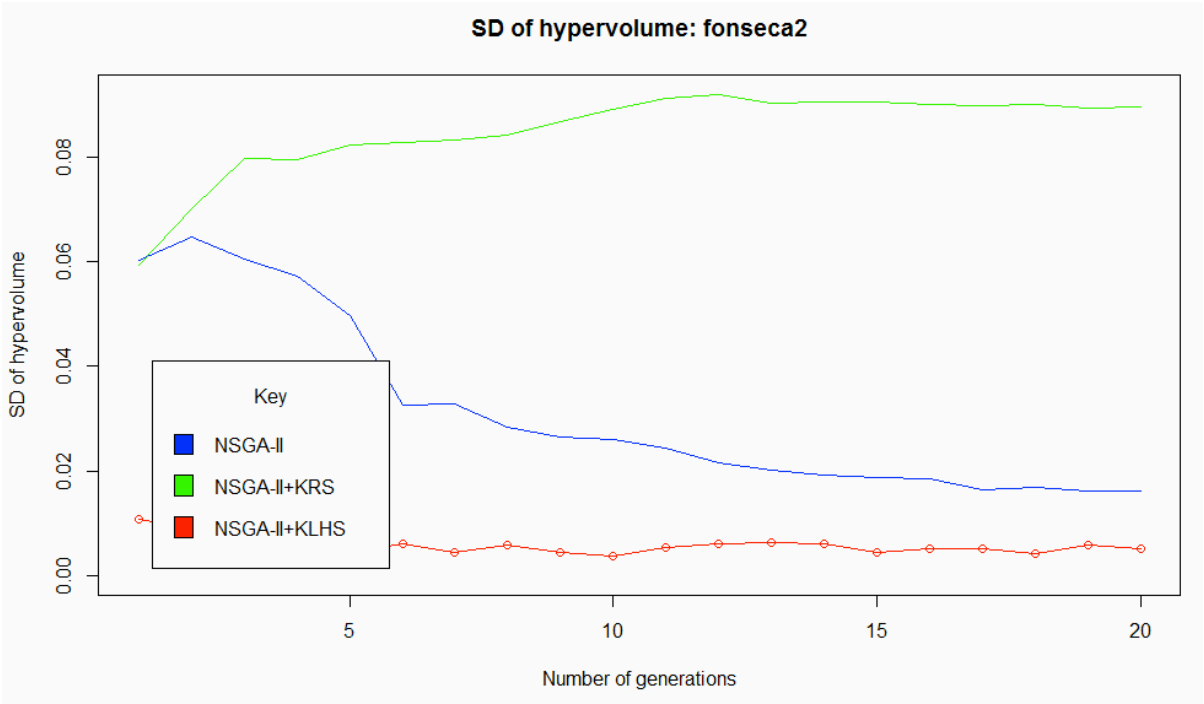


Figure 8-12: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for Fonseca2

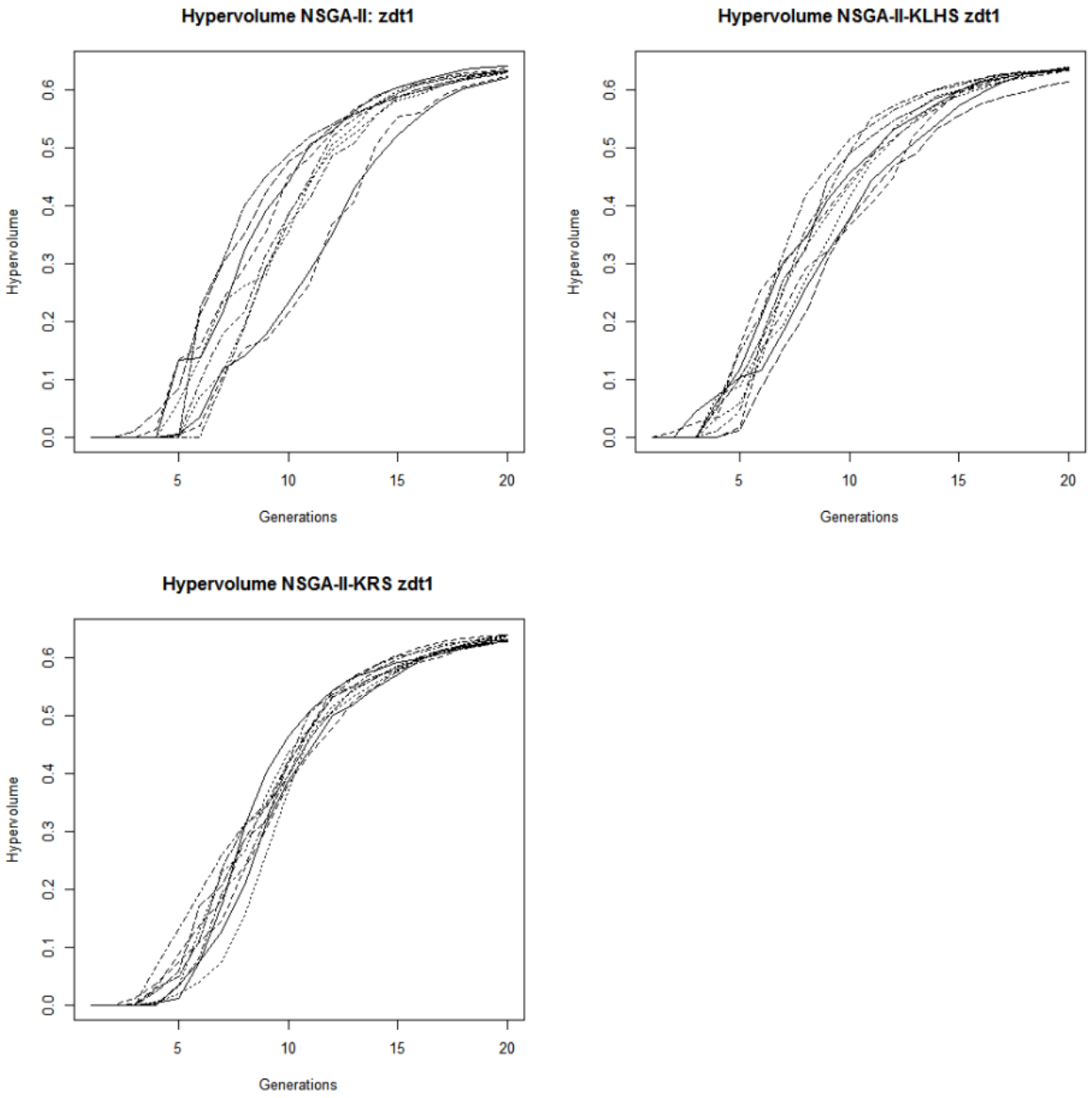
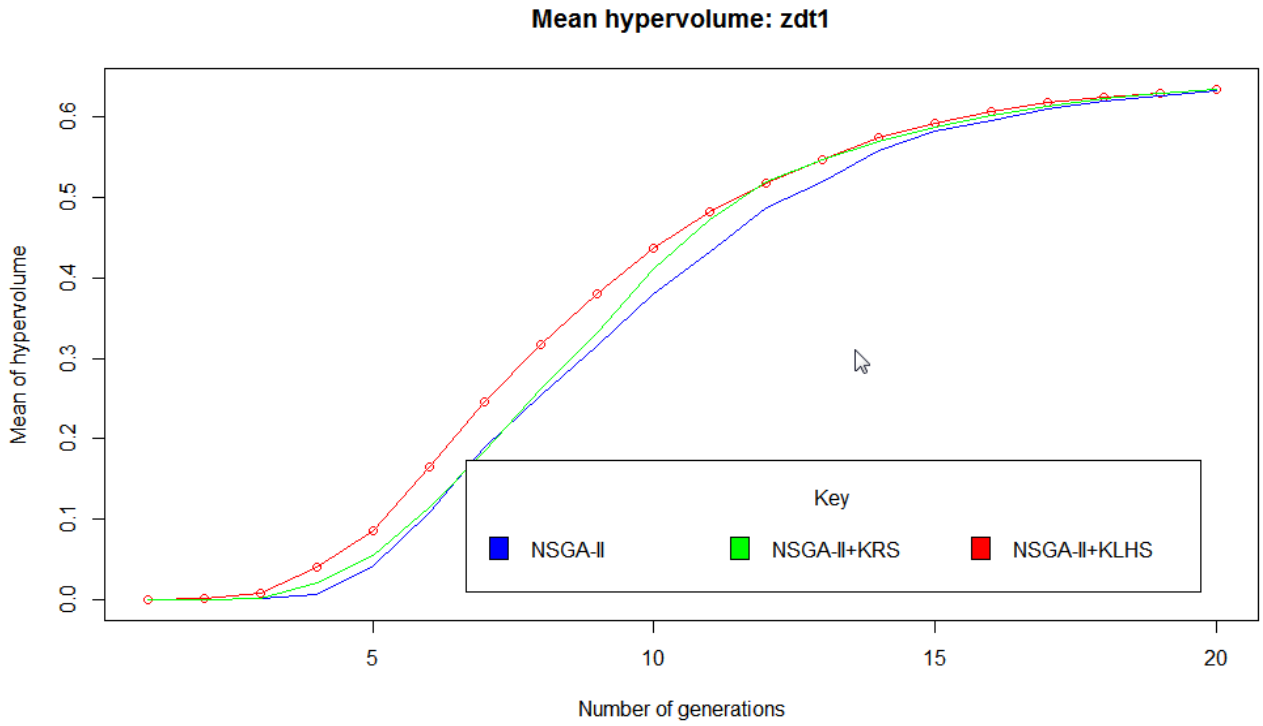


Figure 8-13: Progressions of the hypervolume of each evolution (zdt1)



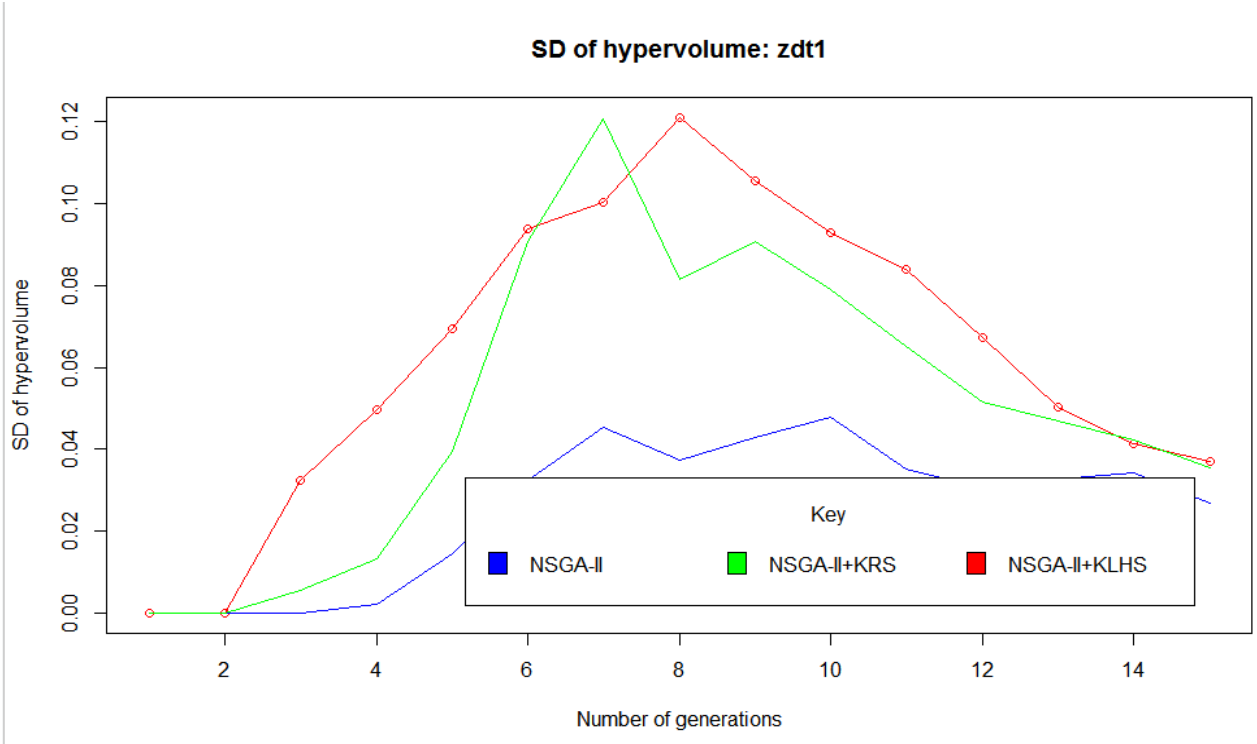


Figure 8-14: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for ZDT1

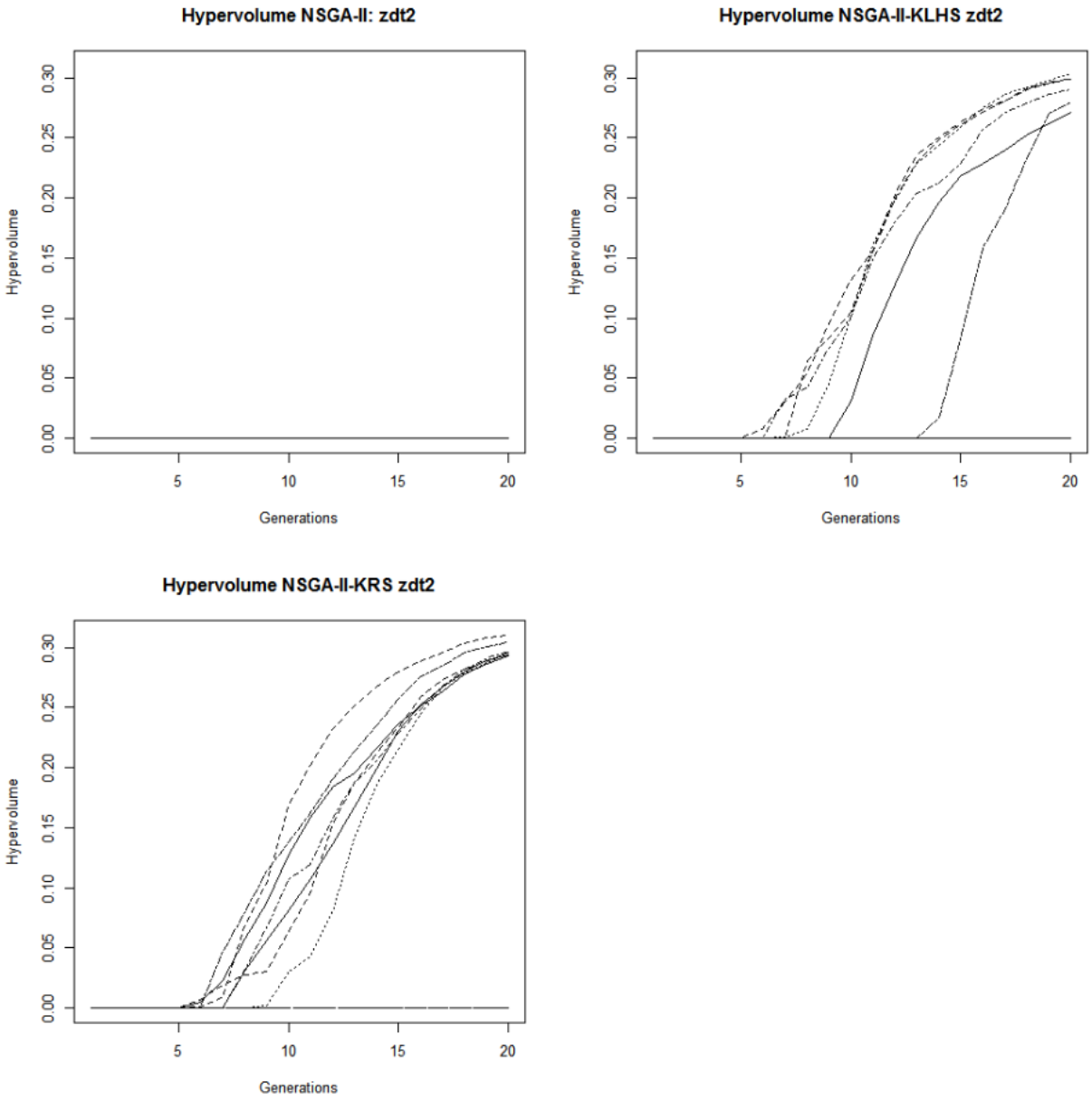
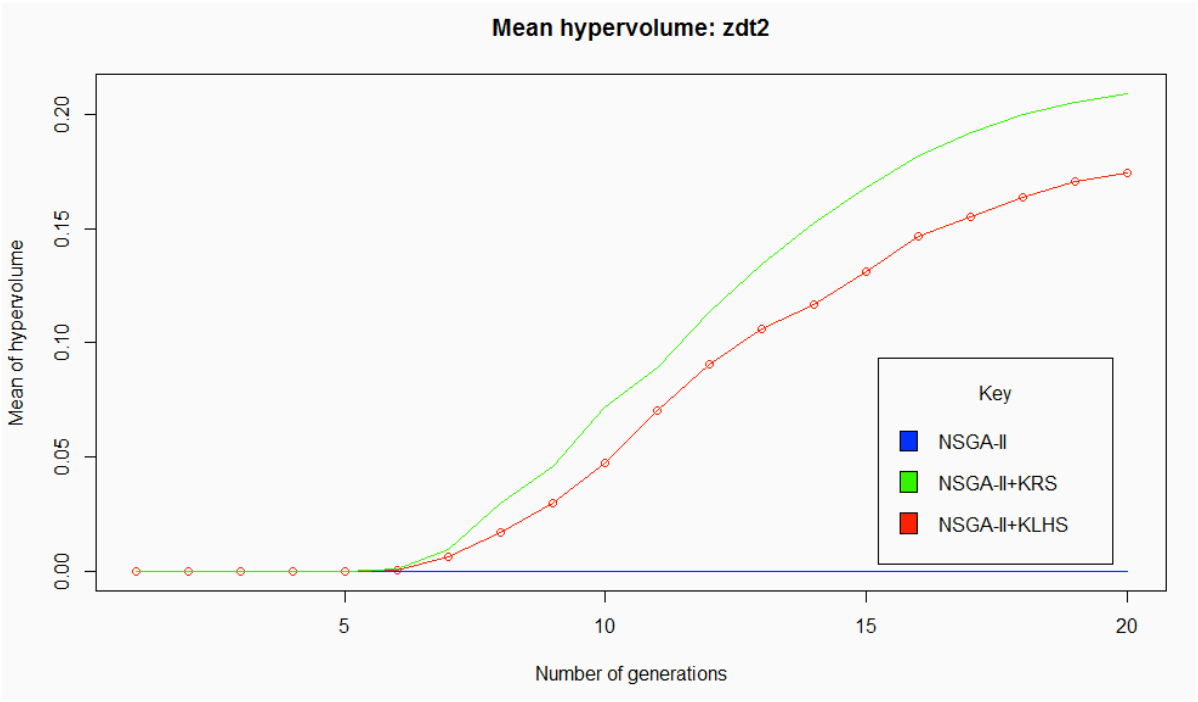


Figure 8-15: Progressions of the hypervolume of each evolution (zdt2)



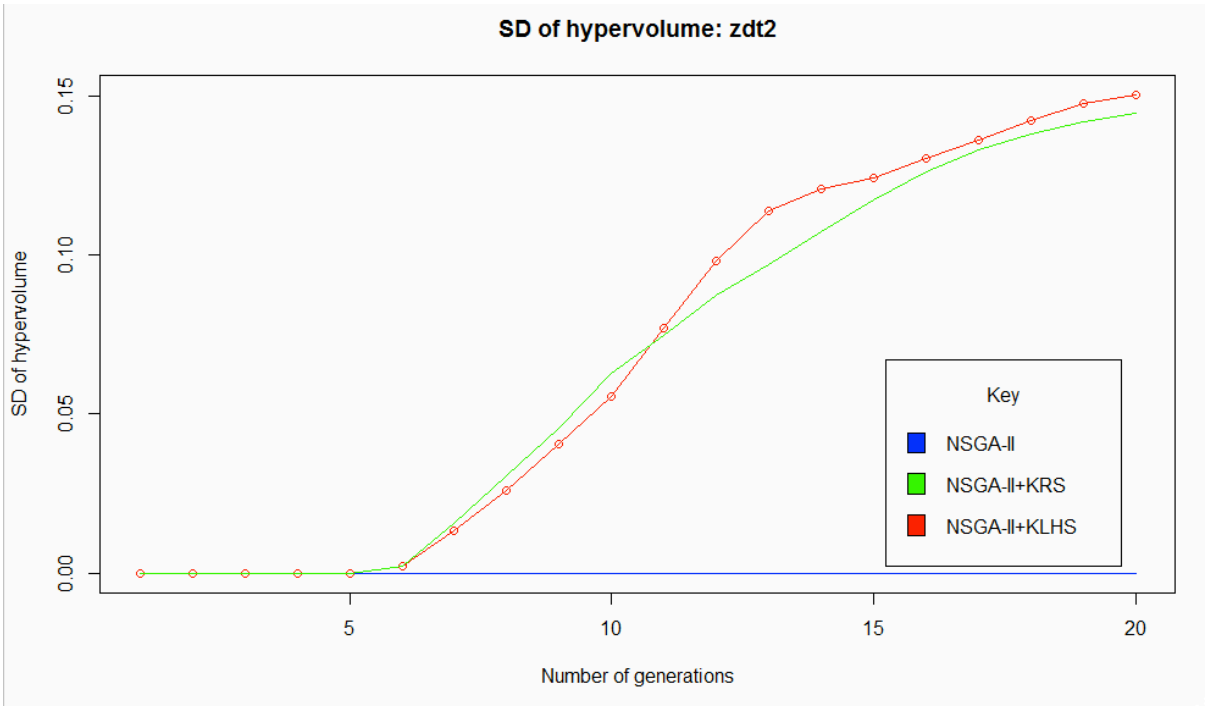


Figure 8-16: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for ZDT2

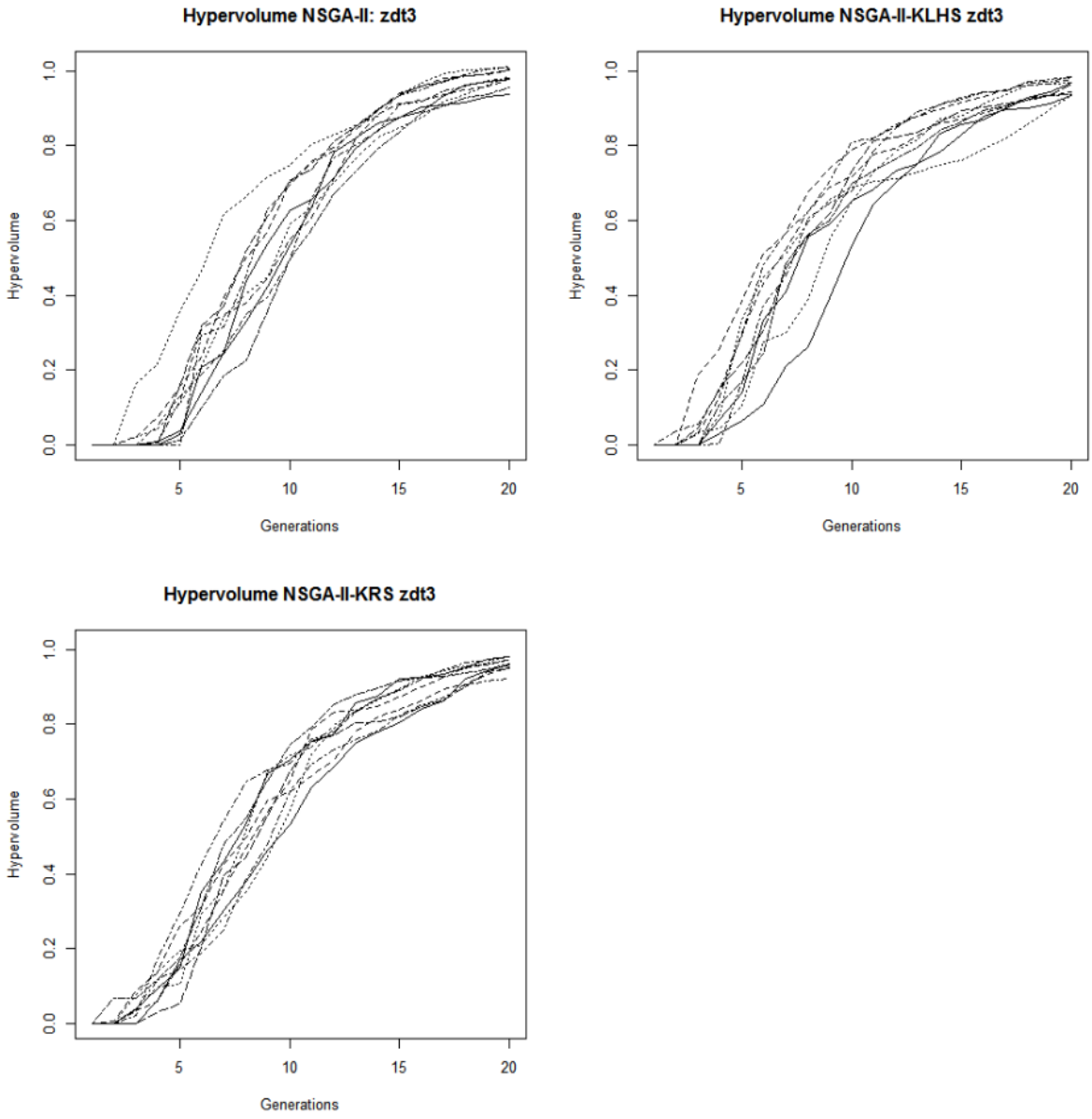
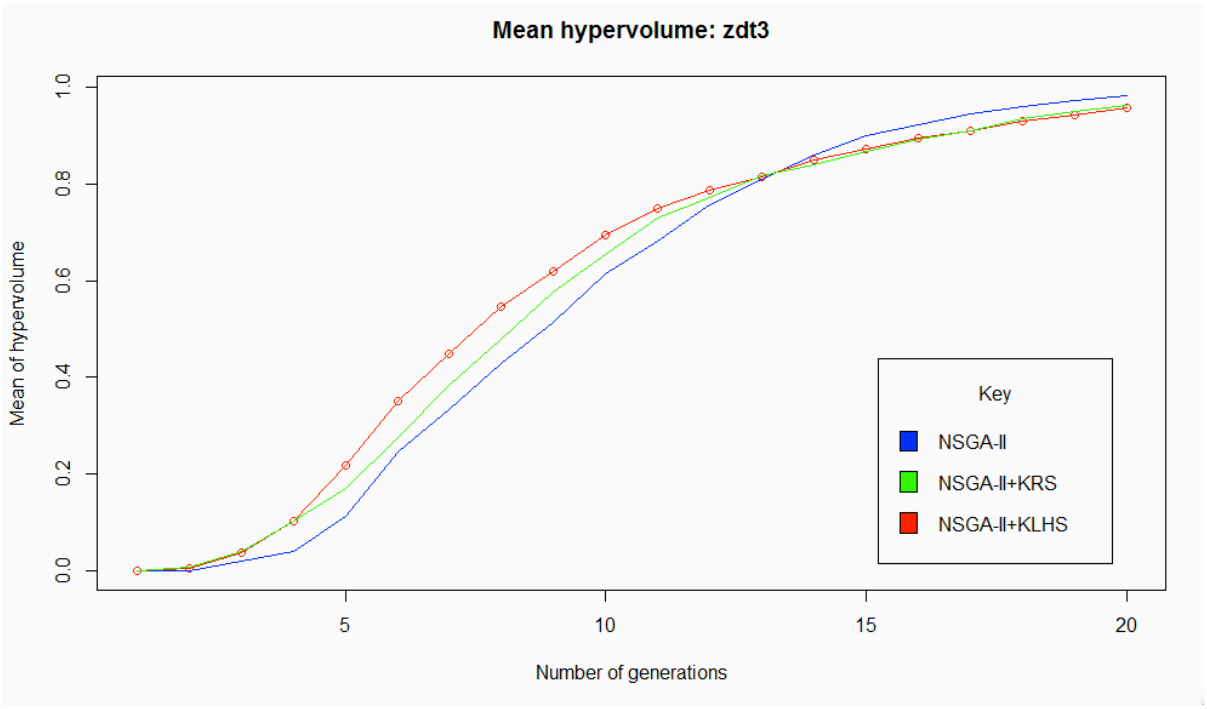


Figure 8-17: Progressions of the hypervolume of each evolution (zdt3)



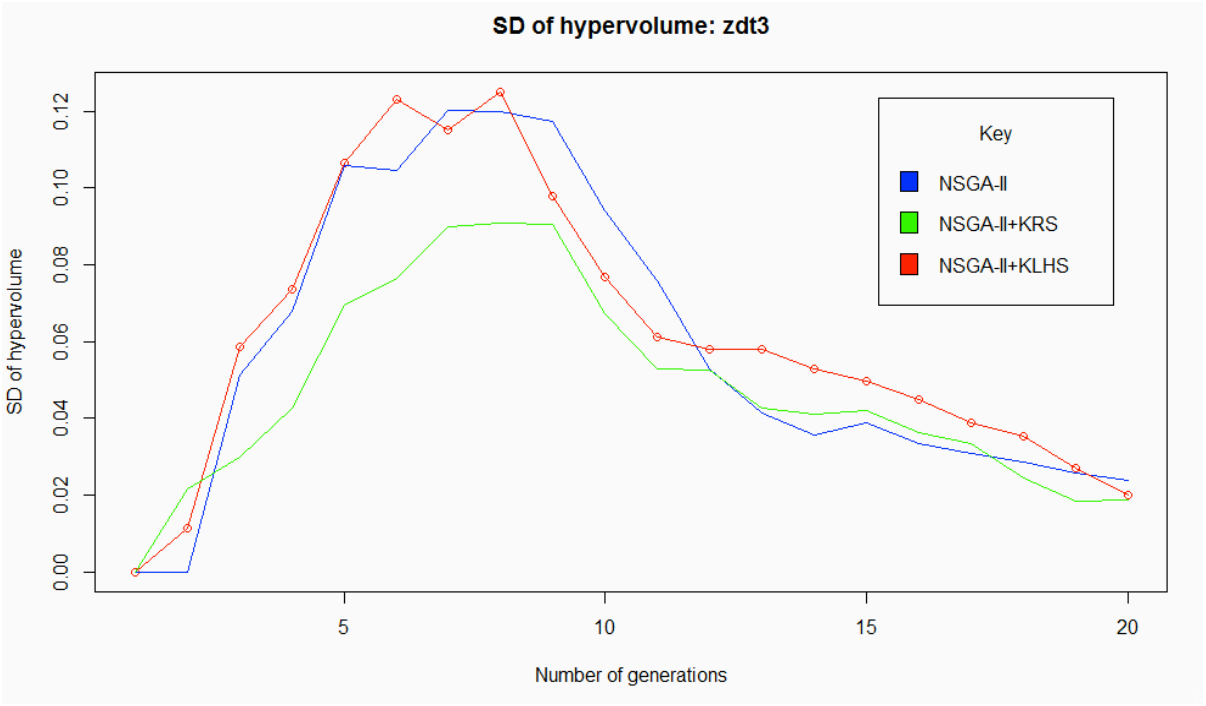


Figure 8-18: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for ZDT3

8.5.2 Building model Results

Figure 8-20 shows the hypervolume for each step in the evolution. The reference point for the hypervolume calculation is $x = 4000,\ y = 6000$.

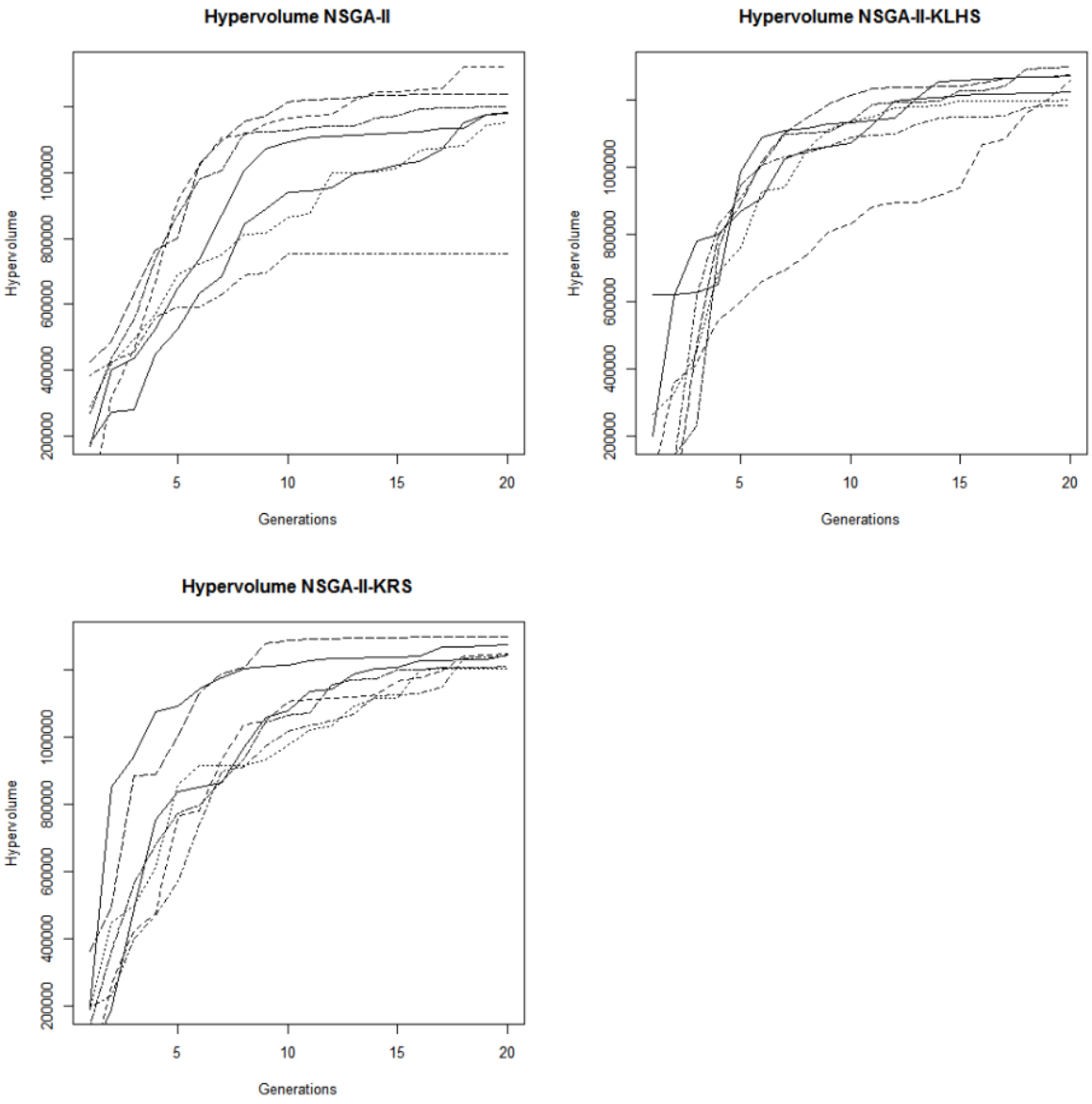
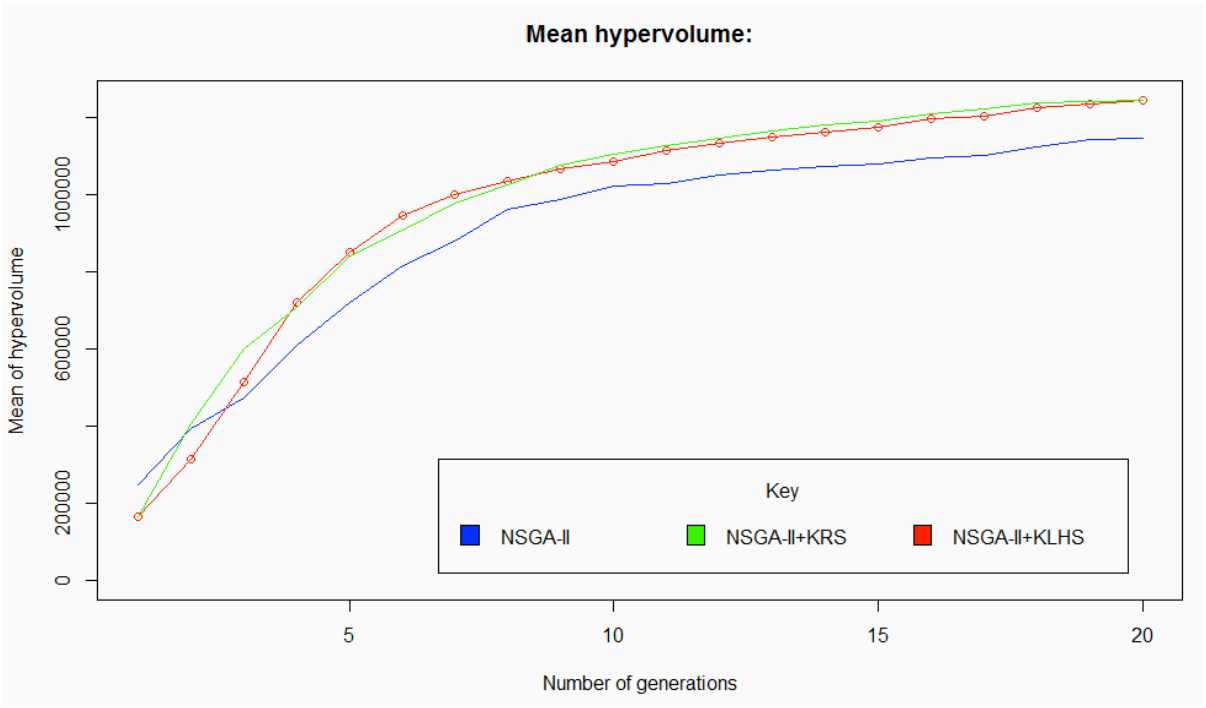


Figure 8-19: Progressions of the hypervolume of each evolution (Building model)



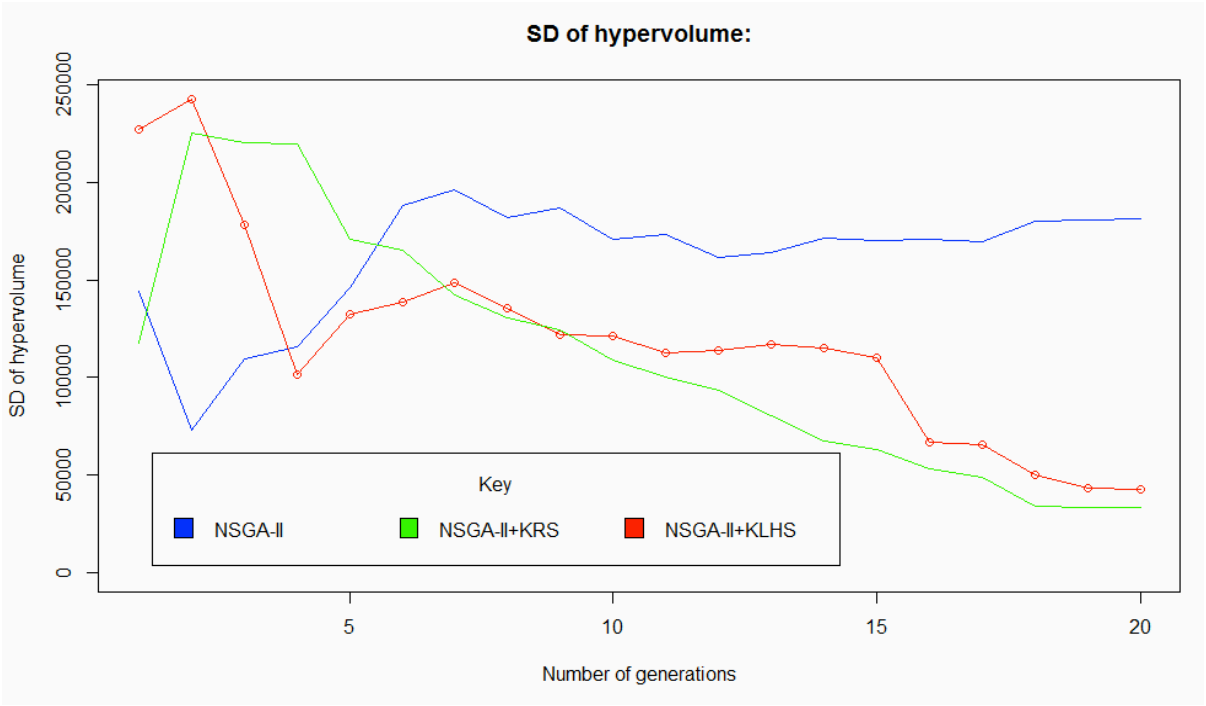


Figure 8-20: NSGA-II, NSGA-II+KRS, NSGA-II+KLHS results for the building model

8.6 Discussion

8.6.1 Test function results

The results of the test functions show that, in nearly all cases, the standard NSGA-II is outperformed by NSGA-II+KLHS and NSGA-II+KRS.

The results for the *deb3* did not fare so as well (see Figure 8-9 and Figure 8-10). Although good progress is made with this function, none of either the NSGA-II, NSGA-II+KRS or NSGA-II+KLHS functions resulted in a reliable convergence of the Pareto-optimal front. However, when performing the analysis on the *Fonseca2* function (Figure 8-11 and Figure 8-12), the NSGA-II+KLHS algorithm appears to be considerably more stable than both the NSGA-II and the NSGA-II+KRS algorithms. Of the NSGA-II and the NSGA-II+KRS algorithms, the NSGA-II appears to produce the most stable results.

For the test function ZDT1 (Figure 8-13 and Figure 8-14), the mean hypervolume of each of the 10 iterations of the test function show that, on average the NSGA-II+KLHS performs best of the three functions. I have shown that, for this function, the mean hypervolume is ahead of both NSGA-II and NSGA-II+KRS. It does however exhibit more variation around the 'midpoint' number of generations, as does NSGA-II+KRS. The NSGA-II produces the least variation in the results, but overall takes longer to reach the asymptote.

The results for the test function ZDT2 are more interesting (Figure 8-15 and Figure 8-16). During the 10 test iterations, the NSGA-II function in this case has a zero mean and standard deviation for the hypervolume. This is due to the location of the reference point used for the hypervolume calculation. Any

hyperspaces where $y_1 > 1$ and/or $y_2 > 1$do not encroach on the target space, and therefore have a hypervolume of 0. The reason that NSGA-II resulted in a zero mean and standard deviation is that it did not have any results in the target space. We can see this from Figure 8-21 below.

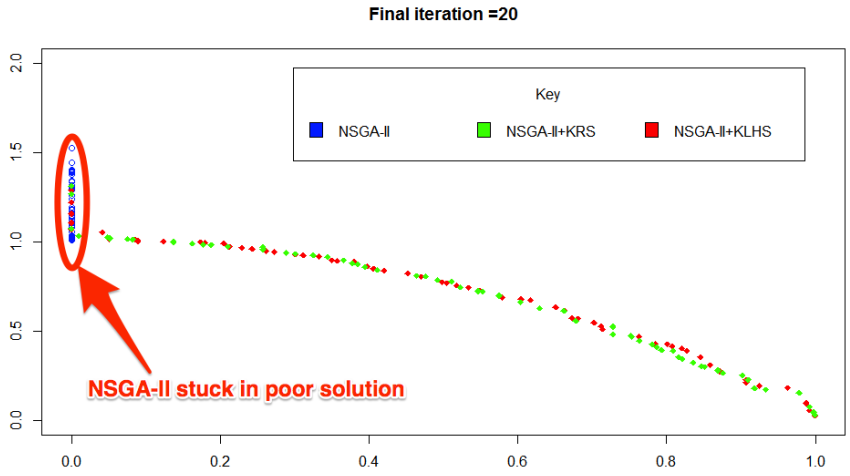


Figure 8-21: Illustration of the NSGA-II algorithm getting 'stuck' during an algorithm generation

The results show that the NSGA-II algorithm has become stuck in the area highlighted. This also happened occasionally for NSGA-II+KLHS and NSGA-II+KRS as we can see from Figure 8-15. Further work is required to investigate why the Kriging augmented solutions outperform the original NSGA-II in this instance.

Finally, all of the algorithms appear to perform well on the *zdt3* function (Figure 8-17 and Figure 8-18). From these results, there is little to distinguish the results, although it could be argued that over 20 generations, the original NSGA-II algorithm marginally outperforms NSGA-II+KRS and NSGA-II+KLHS. This is shown in the slightly larger hypervolume (on average) after 20 generations (Figure 8-18).

8.6.2 Building model results

The results of the building model show that both the NSGA-II+KRS and NSGA-II+KLHS outperform the standard NSGA-II (Figure 8-19 and Figure 8-20). We can see from the graph of the mean hypervolume that the NSGA-II+KLHS and NSGA-II+KRS quickly reach the asymptote of the mean hyper volume, whereas NSGA-II still falls behind. I also find that the SD of the hyper volume remains high for NSGA-II, which indicates that there is significant variance in the result, even after 15-20 iterations of the algorithm.

Digging deeper, I find that the variance in output of the NSGA-II is due to the fact that there is a tendency for this algorithm to get 'stuck' in no Pareto optimal solutions. Figure 8-20 shows the development of the hypervolume for NSGA-II over each of the n repetitions.

It can been seen from Figure 8-20 that the NSGA-II algorithm as applied to the building model gets stuck in solutions that represent a lower hypervolume. The lowest line at the 20th generation appears to not be able to improve on the solution obtained at generation number 10. In contrast, the evolution of the hypervolumes for NSGA-II+KLHS and NSGA-II+KRS do not seem to have this problem. The results show that NSGA-II+KRS and NSGA-II+KLHS all seem to evolve to solutions that are close to the asymptote, even when there is potential to get stuck in the same place as the standard NSGA-II.

If I take an individual example; taking the bottom line at the 10th generation of NSGA-II and NSGA-II+KLHS, we see that the hypervolume is about the same. However, whereas the NSGA-II algorithm tends to be unable to improve on this

solution, the NSGA-II+KLHS on the other hand appears to be able to continue to improve.

A sample of one of the evolution processes is shown in Figure 8-22. Each of the steps shows the output population after 5, 10, 15 and 20 generations for each of NSGA-II, NSGA-II+KLHS and NSGA-II+KRS. We can see that, by the 5th generation, the NSGA-II+KLHS and NSGA-II+KRS algorithms already have a significant advantage over the NSGA-II algorithm, with the NSGA-II+KRS significantly outperforming both. By the 15th generation, all the algorithms are operating in the same area, with the NSGA-II+KLHS showing slightly more diversity in the results than either NSGA-II or NSGA-II+KRS. By the 20th generation, the results of NSGA-II+KRS (shown in yellow) appear to dominate the results of the other algorithms.

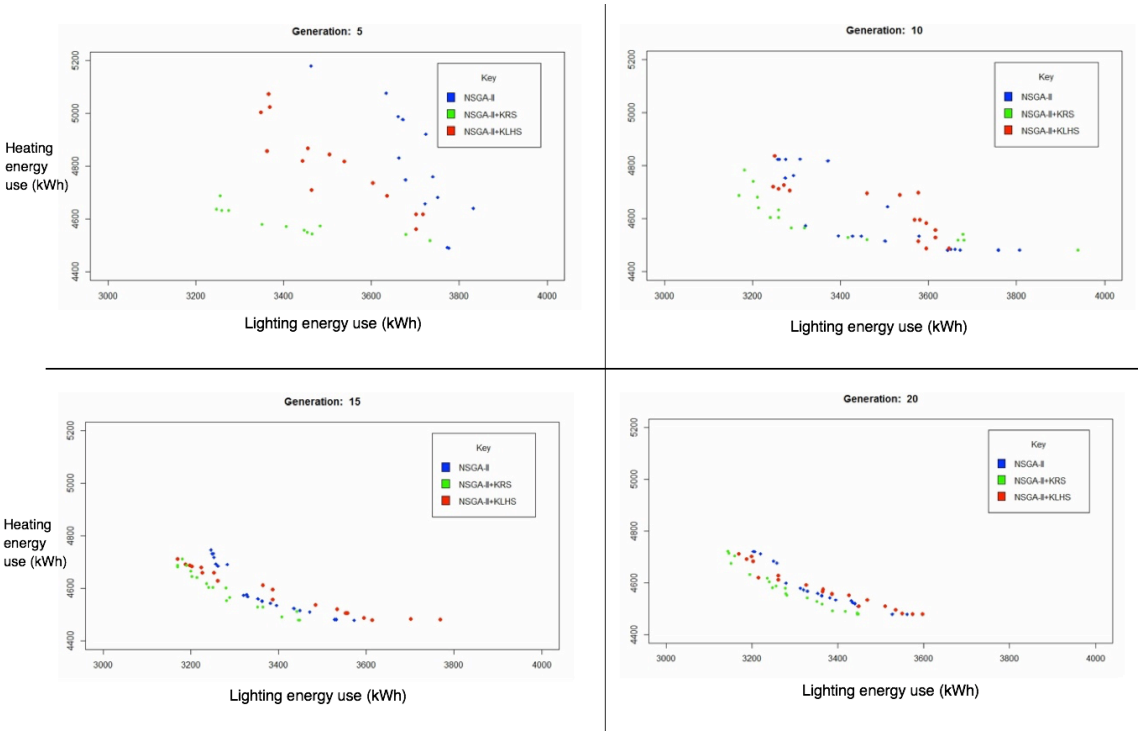


Figure 8-22: Typical evolution of the NSGA-II algorithm for the building model

8.7 Conclusions

The results show that the performance of the algorithm depends strongly on the type of test function. The NSGA-II+KLHS algorithm appears to work much better on the non-convex Pareto fronts (i.e. *Fonseca2* and *zdt2*). For the test functions with convex solutions all three of the algorithms work well for the most part, but NSGA-II+KRS and NSGA-II appear to work equally well. This is also reflected in the results of the building model, which has been shown to have a convex solution, and where the NSGA-II+KLHS works well. However, in this instance, it is matched in performance by the NSGA-II+KRS algorithm. For the building model tested, NSGA-II appears to work less well than the Kriging-augmented algorithms.

8.8 Summary

There are a number of reasons why the algorithms NSGA-II+KLHS and NSGA-II+KRS appear to outperform the standard NSGA-II. This is likely to be because, each time the algorithm for NSGA-II+KRS and NSGA-II+KLHS produced a new parent population, three-times the number of potential solutions are evaluated before the environmental selection. Even though these parameters are only evaluated by the emulator, the ranking algorithm is likely to produce a more diverse population, even after the final population selection.

To date, there has been little research into the use of Kriging to improve pre-existing optimisation procedures, and therefore there are many potential avenues for future work. This initial research has shown that the method has potential, but

there are a number of areas in which future work should be focussed. These areas include;

- testing this approach on a variety of building modelling problems (including those with 3 or more output objectives);
- developing formalised methods for matching the algorithm settings to the optimisation problem; and
- researching methods for determining the optimum limit on the size of the Kriging model.

The first point will require extensive analysis of a variety of problem types. It will also require iterative testing of the various ‘tuning’ parameters used in the Kriging augmented algorithms and the NSGA-II algorithm. The analysis of such parameters includes looking at factors such as the initial Kriging population size, limitations on the size of the Kriging emulator and changing the various parameters that control the offspring population of the GA.

Chapter 9 Discussion and conclusions

In this thesis, a variety of uses for Kriging have been presented. These range from simple meta-modelling and optimisation problems and the management of uncertainty through to the development of robust optimisation algorithms and multi-objective optimisation procedures.

The simple meta-modelling problems show that it is possible to provide an accurate representation of a building simulator with a Kriging meta-model. Although this is not necessarily a new or particularly important finding, it does show that the objective space of building simulator models is amenable to this kind of analysis. Since I have tended to focus the modelling efforts in this thesis on building energy use, as well as briefly touching on lighting levels, further work will therefore be needed to determine whether or not the Kriging methodology is amenable to objective spaces that could (potentially) produce very different kinds of output.

I have also examined the EGO optimisation technique with a building model and have shown that, for the problem considered, the Kriging approach is much more efficient at converging on the global optimum. However, as with the findings of the simple meta-modelling problem, further work is required to determine whether or not this is universally true, both in terms of other GA approaches and in the different problems that we might be optimising for.

The results in Chapter 4, which examine the application of Kriging meta-models in modelling the distribution of energy use under parameter uncertainty, show that Kriging meta-models can be useful as a tool for producing these output

distributions more rapidly than so-called brute force MC simulations. However, these results also highlight some of the pitfalls in this approach. For example, when determining the *difference* between two potential solutions, it appears to be more reliable to emulate the difference, than simply subtract the difference between two emulators. The choice of the correlation function used in the emulator also has a significant effect.

In the final two chapters, I tested a Kriging-based robust optimisation algorithm and an enhanced NSGA-II algorithm for multi-objective optimisation. The robust optimisation algorithm was used to provide a robust design of a building based on pre-defined uncertainty parameters. This approach worked well and uncovered solutions that were more robust to the parameter changes than the standard Kriging EGO.

9.1 Limitations to the work

My work has introduced a number of ways in which Kriging can be applied to building design. However, in attempting to demonstrate a range of applications, it has not provided an extensive appraisal of each method described.

My work has focussed on building energy models. As such, further consideration needs to be given to examining the application of Kriging to other aspects of building design. For example, if Kriging-based optimisation is to be applied in commercial design practices, it will need to consider many different ‘outputs’, such overheating, acoustics, air quality, construction costs, life cycle analyses of construction materials and many others. The use of Kriging in analysing these building performance variables needs to be fully explored.

9.2 Further work

The approaches to the application of Kriging detailed in this thesis would benefit from testing across a wide variety of building types and problems in order to understand their efficacy. There is also the potential for further development of the algorithms themselves, and I have already highlighted potential work in this area too. The algorithms themselves would also benefit from comparisons with alternative methods, for example, in the case of uncertainty analysis, I might compare Kriging to other regression methods. Notwithstanding these potential avenues of further exploration, there is clearly significant potential for developing Kriging as a useful tool for solving building design problems.

9.3 Recommendations to future researchers

The work in this thesis has shown significant for potential for applying Kriging methods for optimisation, uncertainty analysis, robust optimisation and multi-objective optimisation. For future researchers I would recommend beginning with a more building-centric approach. This means focussing on building problems that need solving, rather than simply testing Kriging methods of building-type problems as I have done.

The most promising areas for future research in this area are uncertainty analysis and robust optimisation. Both these areas are clearly closely related. UA looks at the potential uncertainty distribution for a particular design option and RO gives us a method for optimisation under that uncertainty. I believe that finding an efficient way to combine these methods with multi-objective optimisation will be a fruitful area of further research.

In particular, I would recommend that future research focusses on using Kriging to research the problems related to balancing overheating, acoustics, air quality, construction costs, life cycle analyses of construction materials as well as the examining the effect of uncertainty introduced by a changing climate.

Buildings have many performance measures that need to be optimised together. Any useful design algorithm will therefore use a multi-objective approach, but will also need to consider uncertainty and robustness. Based on the research in this thesis, I believe that Kriging techniques could be an effective to achieve this goal.

References

[1] P. de Wilde, “The gap between predicted and measured energy performance of buildings: A framework for investigation,” *Autom. Constr.*, vol. 41, pp. 40–49, May 2014.

[2] “Communication from the European Commission 2020: A strategy for smart, sustainable and inclusive growth,” 2010.

[3] “Communication from the commission to the European Parliament, the Council, the European economic and social committee and the commitee of the regions: A Roadmap for moving to a competitive low carbon economy in 2050,” 2011.

[4] U. K. Government, “Climate Change Act 2008.” 2008.

[5] L. De Boeck, S. Verbeke, A. Audenaert, and L. De Mesmaeker, “Improving the energy performance of residential buildings: A literature review,” *Renew. Sustain. Energy Rev.*, vol. 52, pp. 960–975, Dec. 2015.

[6] W. Tian and P. de Wilde, “Uncertainty and sensitivity analysis of building performance using probabilistic climate projections: A UK case study,” *Autom. Constr.*, vol. 20, no. 8, pp. 1096–1109, Dec. 2011.

[7] A.-T. Nguyen, S. Reiter, and P. Rigo, “A review on simulation-based optimization methods applied to building performance analysis,” *Appl. Energy*, vol. 113, pp. 1043–1058, Jan. 2014.

[8] R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[9] K. F. Fong, V. I. Hanby, and T. T. Chow, “System optimization for {HVAC} energy management using the robust evolutionary algorithm,” *Appl. Therm. Eng.*, vol. 29, no. 11--12, pp. 2327–2334, Aug. 2009.

[10] a. M. Rysanek and R. Choudhary, “Optimum building energy retrofits under technical and economic uncertainty,” *Energy Build.*, vol. 57, pp. 324–337, Feb. 2013.

[11] A. P. Ramallo-González, T. S. Blight, and D. A. Coley, “New optimisation methodology to uncover robust low energy designs that accounts for occupant behaviour or other unknowns,” *J. Build. Eng.*, vol. 2, pp. 59–68, Jun. 2015.

[12] I. Tsoukalas and C. Makropoulos, “Multiobjective optimisation on a budget: Exploring surrogate modelling for robust multi-reservoir rules generation under hydrological uncertainty,” *Environ. Model. Softw.*, vol. 69, pp. 396–413, Jul. 2015.

[13] R. Marijt, “Multi-objective Robust Optimization Algorithms for Improving Energy Consumption and Thermal Comfort of Buildings,” University of Leiden, 2009.

[14] J. L. Loeppky, J. Sacks, W. Welch, and W. J. Welch, “Choosing the Sample

Size of a Computer Experiment: a Practical Guide," *Technometrics*, vol. 51, no. 4, pp. 366–376, 2009.

[15] E. Tresidder, "Accelerated optimisation methods for low-carbon building design," De Montfort University, 2014.

[16] S. Rehman, M. Langelaar, and F. van Keulen, "Efficient Kriging-based robust optimization of unconstrained problems," *J. Comput. Sci.*, vol. 5, no. 6, pp. 872–881, Nov. 2014.

[17] V. Watson, E. Jones, E. Murphy, P. J. Wright, A. Brownlee, and G. Aird, "Industry challenges in using optimisation tools with IES Optimise as a case study."

[18] "TAS Thermal Analysis Simulation Software," *www.edsl.net*, 2016. .

[19] D. Cockcroft, "DesignBuilder building simulation," *www.designbuilder.co.uk*, 2016. .

[20] "Ecotect Analysis - Sustainable Building Design Software - Autodesk." .

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[22] US Department of Energy, "Energy Plus," 2015. [Online]. Available: http://apps1.eere.energy.gov/buildings/energyplus/energyplus_about.cfm.

[23] S. Azhar, "Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry," *Leadersh. Manag. Eng.*, vol. 11, no. 3, pp. 241–252, 2011.

[24] T. H. Beach, O. F. Rana, Y. Rezgui, and M. Parashar, "Cloud computing for the architecture, engineering & construction sector: requirements, prototype & experience," *J Cloud Comput Adv Syst Appl*, vol. 2, no. 1, p. 8, 2013.

[25] Onuma Inc, "Onuma," *www.onuma.com*, 2016. .

[26] Autodesk Inc., "Revit Server," *www.autodesk.co.uk*, 2016. .

[27] Bentley Systems Inc., "ProjectWise - Collaboration and Content Management Software," *www.bentley.com/en/products/brands/projectwise*, 2016. .

[28] Bentley Systems Inc., "AssetWise - Asset Information Management Software," *www.bentley.com/en/products/brands/assetwise*, 2016. .

[29] Graphisoft, "BIMcloud," *www.graphisoft.com*, 2016. .

[30] Jotne IT, "EDM Model Server," *www.epmtech.jotne.com/edmmodelserver-ifc*, 2016. .

[31] H. Kim and M. Parashar, "CometCloud: An Autonomic Cloud Engine," in *Cloud Computing*, John Wiley & Sons, Inc., 2011, pp. 275–297.

[32] Cabinet Office, “Government Construction Strategy,” May 2011.

[33] T. Cerovsek, “A review and outlook for a ‘Building Information Model’ ({BIM)}: A multi-standpoint framework for technological development,” *Adv. Eng. Informatics*, vol. 25, no. 2, pp. 224–244, Apr. 2011.

[34] O. T. Ogunsola and L. Song, “Application of a simplified thermal network model for real-time thermal load estimation,” *Energy Build.*, vol. 96, pp. 309–318, 2015.

[35] L. Van Gelder, P. Das, H. Janssen, and S. Roels, “Comparative study of metamodelling techniques in building energy simulation: Guidelines for practitioners,” *Simul. Model. Pract. Theory*, vol. 49, pp. 245–257, Dec. 2014.

[36] J. Clarke, “A vision for building performance simulation: a position paper prepared on behalf of the IBPSA Board,” *J. Build. Perform. Simul.*, vol. 8, no. 2, pp. 39–43, 2015.

[37] University of Strathclyde, “ESP-r,” *www.esru.strath.ac.uk/Programs/ESP-r*, 2011. .

[38] The University of Wisconsin, “A TRaNsient SYstems Simulation Program (TRNSYS),” *sel.me.wisc.edu/trnsys*, 2016. .

[39] A. Ramallo-Gonzalez, “Modelling, Simulation and Optimisation Methods for Low Energy Buildings,” 2013.

[40] X. Xu and S. Wang, “Optimal simplified thermal models of building envelope based on frequency domain regression using genetic algorithm,” *Energy Build.*, vol. 39, no. 5, pp. 525–536, May 2007.

[41] N. Baker and K. Steemers, “LT Method 3.0---a strategic energy-design tool for Southern Europe,” *Energy Build.*, vol. 23, pp. 251–256, 1996.

[42] S. Wang and X. Xu, “Simplified building model for transient thermal performance estimation using GA-based parameter identification,” *Int. J. Therm. Sci.*, vol. 45, no. 4, pp. 419–432, Apr. 2006.

[43] M. Picco, R. Lollini, and M. Marengo, “Towards energy performance evaluation in early stage building design: A simplification methodology for commercial building models,” *Energy Build.*, vol. 76, pp. 497–505, Jun. 2014.

[44] M. De Rosa, V. Bianco, F. Scarpa, and L. A. Tagliafico, “Heating and cooling building energy demand evaluation; a simplified model and a modified degree days approach,” *Appl. Energy*, vol. 128, pp. 217–229, Sep. 2014.

[45] A. P. Ramallo-González, M. E. Eames, and D. a. Coley, “Lumped parameter models for building thermal modelling: An analytic approach to simplifying complex multi-layered constructions,” *Energy Build.*, vol. 60, pp. 174–184, May 2013.

[46] A. Gasparella, G. Pernigotto, F. Cappelletti, P. Romagnoni, and P. Baggio,

"Analysis and modelling of window and glazing systems energy performance for a well insulated residential building," *Energy Build.*, vol. 43, no. 4, pp. 1030–1037, Apr. 2011.

[47] H. Ren, W. Gao, and Y. Ruan, "Economic optimization and sensitivity analysis of photovoltaic system in residential buildings," *Renew. Energy*, vol. 34, no. 3, pp. 883–889, Mar. 2009.

[48] J. Morrissey and R. E. Horne, "Life cycle cost implications of energy efficiency measures in new residential buildings," *Energy Build.*, vol. 43, no. 4, pp. 915–924, Apr. 2011.

[49] Z. Szalay, "Modelling building stock geometry for energy, emission and mass calculations," *Build. Res. Inf.*, vol. 36, no. 6, pp. 557–567, 2008.

[50] J. Yu, C. Yang, L. Tian, and D. Liao, "Evaluation on energy and thermal performance for residential envelopes in hot summer and cold winter zone of China," *Appl. Energy*, vol. 86, no. 10, pp. 1970–1985, Oct. 2009.

[51] B. Eisenhower, Z. O'Neill, S. Narayanan, V. A. Fonoberov, and I. Mezić, "A methodology for meta-model based optimization in building energy models," *Energy Build.*, vol. 47, pp. 292–301, Apr. 2012.

[52] P. M. Ferreira, A. E. Ruano, S. Silva, and E. Z. E. Conceição, "Neural networks based predictive control for thermal comfort and energy savings in public buildings," *Energy Build.*, vol. 55, pp. 238–251, Dec. 2012.

[53] J.-T. Jin and J.-W. Jeong, "Optimization of a free-form building shape to minimize external thermal load using genetic algorithm," *Energy Build.*, vol. 85, pp. 473–482, Dec. 2014.

[54] E. Znouda, N. Ghrab-Morcos, and A. Hadj-Alouane, "Optimization of Mediterranean building design using genetic algorithms," *Energy Build.*, vol. 39, no. 2, pp. 148–153, Feb. 2007.

[55] M. Palonen, A. Hasan, and K. Siren, "A genetic algorithm for optimization of building envelope and {HVAC} system parameters," in *Proc. Of the 11th IBPSA Conference, Glasgow, Scotland*, 2009.

[56] V. Siddharth, P. V. Ramakrishna, T. Geetha, and A. Sivasubramaniam, "Automatic generation of energy conservation measures in buildings using genetic algorithms," *Energy Build.*, vol. 43, no. 10, pp. 2718–2726, Oct. 2011.

[57] Y. Bichiou and M. Krarti, "Optimization of envelope and {HVAC} systems selection for residential buildings," *Energy Build.*, vol. 43, no. 12, pp. 3373–3382, Dec. 2011.

[58] L. Magnier and F. Haghighat, "Multiobjective optimization of building design using TRNSYS simulations, genetic algorithm, and Artificial Neural Network," *Build. Environ.*, vol. 45, no. 3, pp. 739–746, Mar. 2010.

[59] J. A. Wright, H. A. Loosemore, and R. Farmani, "Optimization of building thermal design and control by multi-criterion genetic algorithm," *Energy Build.*, vol. 34, no. 9, pp. 959–972, Oct. 2002.

[60] F. Pernodet, H. Lahmidi, and P. Michel, “Use of genetic algorithms for multicriteria optimization of building refurbishment,” in *11th International IBPSA Conference*, 2009.

[61] A. Alajmi and J. Wright, “Selecting the Most Efficient Genetic Algorithm Sets in Solving Unconstrained Building Optimization Problem,” *Int. J. Sustain. Built Environ.*, vol. 3, no. 1, pp. 18–26, 2014.

[62] R. L. Haupt, “Optimum population size and mutation rate for a simple real genetic algorithm that optimizes array factors,” in *IEEE Antennas and Propagation Society International Symposium. Transmitting Waves of Progress to the Next Millennium. 2000 Digest. Held in conjunction with: USNC/URSI National Radio Science Meeting (Cat. No.00CH37118)*, vol. 2, pp. 1034–1037.

[63] P. Reed, B. S. Minsker, and D. E. Goldberg, “Simplifying multiobjective optimization: An automated design methodology for the nondominated sorted genetic algorithm-II,” *Water Resour. Res.*, vol. 39, no. 7, 2003.

[64] F. P. Chantrelle, H. Lahmidi, W. Keilholz, M. El Mankibi, and P. Michel, “Development of a multicriteria tool for optimizing the renovation of buildings,” *Appl. Energy*, vol. 88, no. 4, pp. 1386–1394, Apr. 2011.

[65] M. Hamdy, M. Palonen, and A. Hasan, “Implementation of pareto-archive NSGA-II algorithms to a nearly-zero-energy building optimisation problem,” *BSO12 Ibpsa-engl.*, pp. 10–11, 2012.

[66] A. E. I. Brownlee and J. A. Wright, “Constrained, mixed-integer and multi-objective optimisation of building designs by NSGA-II with fitness approximation,” *Appl. Soft Comput. J.*, vol. 33, pp. 114–126.

[67] V. Dixit, N. Seshadrinath, and M. K. Tiwari, “Performance measures based optimization of supply chain network resilience: A NSGA-II + Co-Kriging approach,” *Comput. Ind. Eng.*

[68] J. Knowles and D. Corne, “The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation,” in *Evolutionary Computation, 1999. {CEC} 99. Proceedings of the 1999 Congress on*, 1999, vol. 1, p. 105 Vol. 1.

[69] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach,” *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[70] Design Builder Software Ltd., “Design Builder.” 2015.

[71] R. Evins, “A review of computational optimisation methods applied to sustainable building design,” *Renew. Sustain. Energy Rev.*, vol. 22, pp. 230–245, 2013.

[72] V. Machairas, A. Tsangrassoulis, and K. Axarli, “Algorithms for optimization of building design: A review,” *Renew. Sustain. Energy Rev.*, vol. 31, pp. 101–112, 2014.

[73] R. Evins, P. Pointer, R. Vaidyanathan, and S. Burgess, “A case study

exploring regulated energy use in domestic buildings using design-of-experiments and multi-objective optimisation," *Build. Environ.*, vol. 54, pp. 126–136, 2012.

[74] M. Palonen, A. Hasan, and K. Siren, "A Genetic Algorithm for Optimization of Building Envelope and HVAC System Parameters," in *Proceedings of the 11th International IBPSA Conference*, 2009.

[75] S. Carlucci, G. Cattarin, F. Causone, and L. Pagliano, "Multi-objective optimization of a nearly zero-energy building based on thermal and visual discomfort minimization using a non-dominated sorting genetic algorithm (NSGA-II)," *Energy Build.*, vol. 104, pp. 378–394, 2015.

[76] P. O. Fanger and Others, "Thermal comfort. Analysis and applications in environmental engineering," *Therm. Comf. Anal. Appl. Environ. Eng.*, 1970.

[77] J. K. Day and D. E. Gunderson, "Understanding high performance buildings: The link between occupant knowledge of passive design systems, corresponding behaviors, occupant comfort and environmental satisfaction," *Build. Environ.*, vol. 84, pp. 114–124, Jan. 2015.

[78] P. C. da Silva, V. Leal, and M. Andersen, "Occupants' behaviour in energy simulation tools: lessons from a field monitoring campaign regarding lighting and shading control," *J. Build. Perform. Simul.*, vol. 8, no. 5, pp. 338–358, 2015.

[79] J. Heinonen and S. Junnila, "Residential energy consumption patterns and the overall housing energy requirements of urban and rural households in Finland," *Energy Build.*, vol. 76, pp. 295–303, Jun. 2014.

[80] I. Richardson, M. Thomson, and D. Infield, "A high-resolution domestic building occupancy model for energy demand simulations," *Energy Build.*, vol. 40, no. 8, pp. 1560–1566, Jan. 2008.

[81] T. S. Blight and D. A. Coley, "Sensitivity analysis of the effect of occupant behaviour on the energy consumption of passive house dwellings," *Energy Build.*, vol. 66, pp. 183–192, Nov. 2013.

[82] A. Ioannou and L. C. M. Itard, "Energy performance and comfort in residential buildings: Sensitivity for building parameters and occupancy," *Energy Build.*, vol. 92, pp. 216–233, 2015.

[83] Z. M. Gill, M. J. Tierney, I. M. Pegg, and N. Allan, "Low-energy dwellings: the contribution of behaviours to actual performance," *Build. Res. Inf.*, vol. 38, no. 5, pp. 491–508, 2010.

[84] D. R. F Haldi, "The impact of occupants' behaviour on building energy demand," *J. Build. Perform. Simul.*, vol. 4, no. 4, pp. 323–338, 2011.

[85] C. Demanuele, T. Tweddell, and M. Davies, "Bridging the gap between predicted and actual energy performance in schools," in *Proceedings of the World Renewable Energy Congress XI*, 2010, no. September, pp. 1–6.

[86] R. Galvin, "Making the 'rebound effect' more useful for performance evaluation of thermal retrofits of existing homes: Defining the 'energy

savings deficit' and the 'energy performance gap,'" *Energy Build.*, vol. 69, pp. 515–524, Feb. 2014.

[87] A. C. Menezes, A. Cripps, D. Bouchlaghem, and R. Buswell, "Predicted vs. actual energy performance of non-domestic buildings: Using post-occupancy evaluation data to reduce the performance gap," *Appl. Energy*, vol. 97, pp. 355–364, Sep. 2012.

[88] F. Haldi and D. Robinson, "The impact of occupants' behaviour on building energy demand," *J. Build. Perform. Simul.*, vol. 4, no. 4, pp. 323–338, Dec. 2011.

[89] E. Azar and C. C. Menassa, "A comprehensive analysis of the impact of occupancy parameters in energy simulation of office buildings," *Energy Build.*, vol. 55, pp. 841–853, Dec. 2012.

[90] S. de Wit and G. Augenbroe, "Analysis of uncertainty in building design evaluations and its implications," *Energy Build.*, vol. 34, no. 9, pp. 951–958, 2002.

[91] F. Nicol, "TM52: The limits of thermal comfort: avoiding overheating in European Buildings." 2013.

[92] a. D. Peacock, D. P. Jenkins, and D. Kane, "Investigating the potential of overheating in UK dwellings as a consequence of extant climate change," *Energy Policy*, vol. 38, no. 7, pp. 3277–3288, Jul. 2010.

[93] C. Lodge, S. M. Porritt, L. Shao, P. C. Cropper, and C. I. Goodier, "Occupancy patterns and their effect on interventions to reduce overheating in dwellings during heat waves," in *Adapting to Change: New Thinking on Comfort*, 2010.

[94] D. P. Jenkins, S. Patidar, P. F. G. Banfill, and G. J. Gibson, "Probabilistic climate projections with dynamic building simulation: Predicting overheating in dwellings," *Energy Build.*, vol. 43, no. 7, pp. 1723–1731, Jul. 2011.

[95] P. F. G. Banfill, D. P. Jenkins, S. Patidar, M. Gul, G. F. Menzies, and G. J. Gibson, "The risk of buildings overheating in a low-carbon climate change future," in *Proceedings of the International Conference for Enhanced Building Operations*, 2012.

[96] D. P. Jenkins, M. Gul, S. Patidar, P. F. G. Banfill, G. Gibson, and G. Menzies, "Designing a methodology for integrating industry practice into a probabilistic overheating tool for future building performance," *Energy Build.*, vol. 54, pp. 73–80, Nov. 2012.

[97] V. M. Nik and A. Sasic Kalagasidis, "Impact study of the climate change on the energy performance of the building stock in Stockholm considering four climate uncertainties," *Build. Environ.*, vol. 60, pp. 291–304, Feb. 2013.

[98] Met Office, "Using Climate Projections," *ukclimateprojections.metoffice.gov.uk*, 2012. .

[99] Y. Ji, R. Fitton, W. Swan, and P. Webster, "Assessing overheating of the

UK existing dwellings -- A case study of replica Victorian end terrace house," *Build. Environ.*, vol. 77, pp. 1–11, Jul. 2014.

[100] "CIBSE - Guide A Environmental Design - 2015 Edition." .

[101] X. Wang and Y. Li, "Predicting urban heat island circulation using CFD," *Build. Environ.*, vol. 99, pp. 82–97, Apr. 2016.

[102] B. J. Hatchett, D. Koračin, J. F. Mejía, and D. P. Boyle, "Assimilating urban heat island effects into climate projections," *J. Arid Environ.*, vol. 128, pp. 59–64, May 2016.

[103] D. Lauwaet, K. De Ridder, S. Saeed, E. Brisson, F. Chatterjee, N. P. M. van Lipzig, B. Maiheu, and H. Hooyberghs, "Assessing the current and future urban heat island of Brussels," *Urban Clim.*, vol. 15, pp. 1–15, Mar. 2016.

[104] Z. Tan, K. K.-L. Lau, and E. Ng, "Urban tree design approaches for mitigating daytime urban heat island effects in a high-density urban environment," *Energy Build.*, vol. 114, pp. 265–274, 2016.

[105] D. Zhou, L. Zhang, L. Hao, G. Sun, Y. Liu, and C. Zhu, "Spatiotemporal trends of urban heat island effect along the urban development intensity gradient in China," *Sci. Total Environ.*, vol. 544, pp. 617–626, 2016.

[106] H. S. Bagiorgas and G. Mihalakakou, "On the influence of the urban heat island on the cooling load of a school building in Athens, Greece," *J. Atmos. Sol. Terr. Phys.*, vol. 138--139, pp. 179–186, Feb. 2016.

[107] E. Oikonomou, M. Davies, A. Mavrogianni, P. Biddulph, P. Wilkinson, and M. Kolokotroni, "Modelling the relative importance of the urban heat island and the thermal quality of dwellings for overheating in London," *Build. Environ.*, vol. 57, pp. 223–238, Nov. 2012.

[108] C. J. Hopfe and J. L. M. Hensen, "Uncertainty analysis in building performance simulation for design support," *Energy Build.*, vol. 43, no. 10, pp. 2798–2805, Oct. 2011.

[109] H.-G. Beyer and B. Sendhoff, "Robust Optimization – A Comprehensive Survey."

[110] Z. Yu, B. C. M. Fung, F. Haghighat, H. Yoshino, and E. Morofsky, "A systematic procedure to study the influence of occupant behavior on building energy consumption," *Energy Build.*, vol. 43, no. 6, pp. 1409–1417, Jun. 2011.

[111] "Passivhaus: The world's leading fabric first approach to low energy buildings." .

[112] S. de Wit and G. Augenbroe, "Analysis of uncertainty in building design evaluations and its implications," *Energy Build.*, vol. 34, no. 9, pp. 951–958, Oct. 2002.

[113] T. Walter, P. N. Price, and M. D. Sohn, "Uncertainty estimation improves energy measurement and verification procedures," *Appl. Energy*, vol. 130,

pp. 230–236, 2014.

[114] M. C. Georgiadou, T. Hacking, and P. Guthrie, “A conceptual framework for future-proofing the energy performance of buildings,” *Energy Policy*, vol. 47, pp. 145–155, Aug. 2012.

[115] N. Eyre and P. Baruah, “Uncertainties in future energy demand in UK residential heating,” *Energy Policy*, vol. 87, pp. 641–653, Dec. 2015.

[116] B. van Ruijven, B. de Vries, D. P. van Vuuren, and J. P. van der Sluijs, “A global model for residential energy use: Uncertainty in calibration to regional data,” *Energy*, vol. 35, no. 1, pp. 269–282, Jan. 2010.

[117] M. Kavgic, A. Mavrogianni, D. Mumovic, A. Summerfield, Z. Stevanovic, and M. Djurovic-Petrovic, “A review of bottom-up building stock models for energy consumption in the residential sector,” *Build. Environ.*, vol. 45, no. 7, pp. 1683–1697, Jul. 2010.

[118] A. T. Booth and R. Choudhary, “Decision making under uncertainty in the retrofit analysis of the UK housing stock: Implications for the Green Deal,” *Energy Build.*, vol. 64, pp. 292–308, Sep. 2013.

[119] G. B. Dantzig, “Linear Programming under Uncertainty,” *Manag. Sci.*, vol. 1, no. 3–4, pp. 197–206, 1955.

[120] D. Bertsimas, D. B. Brown, and C. Caramanis, “Theory and Applications of Robust Optimization,” *SIAM Rev.*, vol. 53, no. 3, pp. 464–501, 2011.

[121] A. L. Soyster, “Convex Programming with {Set-Inclusive} Constraints and Applications to Inexact Linear Programming,” *Oper. Res.*, vol. 21, no. 5, pp. 1154–1157, 1973.

[122] J. E. Falk, “Exact Solutions of Inexact Linear Programs,” *Oper. Res.*, vol. 24, no. 4, pp. 783–787, 1976.

[123] A. Ben-Tal and A. Nemirovski, “Robust Convex Optimization,” *Math. Oper. Res.*, vol. 23, no. 4, pp. 769–805, 1998.

[124] A. Ben-Tal and A. Nemirovski, “Robust Solutions of Uncertain Linear Programs,” *Oper. Res. Lett.*, vol. 25, no. 1, pp. 1–13, 1999.

[125] A. Ben-Tal and A. Nemirovski, “Robust solutions of Linear Programming problems contaminated with uncertain data,” *Math. Program.*, vol. 88, pp. 411–421, 2000.

[126] L. El Ghaoui, F. Oustry, and H. Lebret, “Robust Solutions to Uncertain Semidefinite Programs,” *SIAM J. Optim.*, vol. 9, no. 1, pp. 33–52, 1998.

[127] L. El Ghaoui and H. Lebret, “Robust Solutions to {Least-Squares} Problems with Uncertain Data,” *SIAM J. Matrix Anal. Appl.*, vol. 18, no. 4, pp. 1035–1064, Oct. 1997.

[128] C. J. Hopfe, M. T. M. Emmerich, R. Marijt, and J. Hensen, “Robust multi-criteria design optimisation in building design,” *Proc. Build. Simul. Optim. Loughborough, UK*, pp. 118–125, 2012.

[129] D. G. Krige, “A statistical approach to some basic mine valuation problems on the Witwatersrand,” *J.Chem.Met.Min.Soc.S.Afr.*, pp. 119–139, 1951.

[130] T. W. Simpson, J. J. Korte, T. M. Mauery, and F. Mistree, “Comparison of Response Surface and Kriging Models for Multidisciplinary Design Optimization,” 1998.

[131] E. Bernardini, S. M. J. Spence, D. Wei, and A. Kareem, “Aerodynamic shape optimization of civil structures: A CFD-enabled Kriging-based approach,” *J. Wind Eng. Ind. Aerodyn.*, vol. 144, pp. 154–164, Sep. 2015.

[132] F. Carrat and A. J. Valleron, “Epidemiologic mapping using the "kriging" method: application to an influenza-like illness epidemic in France.,” *Am. J. Epidemiol.*, vol. 135, no. 11, pp. 1293–300, Jun. 1992.

[133] I. Andrianakis, I. R. Vernon, N. McCreesh, T. J. McKinley, J. E. Oakley, R. N. Nsubuga, M. Goldstein, and R. G. White, “Bayesian history matching of complex infectious disease models using emulation: a tutorial and a case study on {HIV} in Uganda,” *PLoS Comput. Biol.*, vol. 11, no. 1, p. e1003968, Jan. 2015.

[134] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, vol. 14. 2006.

[135] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning.*, vol. 14, no. 2. 2006.

[136] S. Conti, J. P. Gosling, J. E. Oakley, and a. O’Hagan, “Gaussian process emulation of dynamic computer codes,” *Biometrika*, vol. 96, no. 3, pp. 663–676, Jun. 2009.

[137] O. Roustant, D. Ginsbourger, and Y. Deville, “DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization,” *J. Stat. Softw.*, vol. 51, no. 1, 2012.

[138] W. R. Mebane Jr, J. S. Sekhon, and Others, “Genetic optimization using derivatives: the rgenoud package for R,” *J. Stat. Softw.*, vol. 42, no. 11, pp. 1–26, 2011.

[139] A. O’Hagan, “Bayesian analysis of computer code outputs: A tutorial,” *Reliab. Eng. Syst. Saf.*, vol. 91, no. 10–11, pp. 1290–1300, Oct. 2006.

[140] P. C. Tabares-Velascoa and B. Griffith, “Diagnostic test cases for verifying surface heat transfer algorithms and boundary conditions in building energy simulation programs,” *J. Build. Perform. Simul.*, vol. 5, no. 5, pp. 329–346, 2012.

[141] CIBSE, “CIBSE Weather Data Sets: Weather Data and Climate Change InformationTitle,” *www.cibse.org*, 2016. .

[142] L. S. Bastos and A. O’Hagan, “Diagnostics for Gaussian Process Emulators,” *Technometrics*, vol. 51, no. 4, pp. 425–438, Nov. 2009.

[143] J. E. Oakley and A. O’Hagan, “Probabilistic sensitivity analysis of complex

models: a Bayesian approach," *J. R. Stat. Soc. Ser. B (Statistical Methodol.*, vol. 66, no. 3, pp. 751–769, Aug. 2004.

[144] Anthony O'Hagan, J. M. Bernardo, J. O. Berger, A. P. Dawid, A. F. M. Smith (eds, Marc C. Kennedy, Jeremy E. Oakley, "Uncertainty Analysis and other Inference Tools for Complex Computer Codes," 1998.

[145] S. Wilcox and W. Marion, *Users' Manual for TMY3 Data Sets Users Manual for TMY3 Data Sets.* 2008.

[146] BSi, "BS EN ISO 15927 - Hygrothermal performance of buildings - Calculation and presentation of climatic data - Part 4 : Hourly data for assessing the annual energy use for heating and cooling. (2005)," 2005.

[147] T. Kershaw, M. Eames, and D. Coley, "Assessing the risk of climate change for buildings: A comparison between multi-year and probabilistic reference year simulations," *Build. Environ.*, vol. 46, no. 6, pp. 1303–1308, Jun. 2011.

[148] T. Kershaw, M. Eames, and D. Coley, "Comparison of multi-year and reference year building simulations," *Build. Serv. Eng. Res. Technol.*, vol. 31, no. 4, pp. 357–369, Aug. 2010.

[149] M. F. Jentsch, G. J. Levermore, J. B. Parkinson, and M. E. Eames, "Limitations of the CIBSE design summer year approach for delivering representative near-extreme summer weather conditions," *Build. Serv. Eng. Res. Technol.*, 2013.

[150] M. E. Eames, A. P. Ramallo-Gonzalez, and M. J. Wood, "An update of the UK's test reference year: The implications of a revised climate on building design," *Build. Serv. Eng. Res. Technol.*, 2015.

[151] Department for Communities and Local Government, "English Housing Survey 2011," Jul-2013. .

[152] J. V Roshan and H. Ying, "Orthogonal-maximim Latin hypercube designs," *Stat. Sin.*, vol. 18, no. 1, pp. 171–186, 2008.

[153] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *J. Glob. Optim.*, vol. 13, no. 4, pp. 455–492, 1998.

[154] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, Feb. 2006.

[155] ASHRAE and US Green Building Council, *ASHRAE 189.1-2014: Standard 189.1-2014 -- Standard for the Design of High-Performance Green Buildings*. ASHRAE, 2014.

[156] "UKCP09: Gridded observation data sets," *The Met Office*. Met Office, FitzRoy Road, Exeter, EX1 3PB., 2008.

[157] L. Scrucca, "GA: A Package for Genetic Algorithms in R," *J. Stat. Softw.*, vol. 53, no. 4, pp. 1–37, 2013.

[158] E. M. Malatji, J. Zhang, and X. Xia, “A multiple objective optimisation model for building energy efficiency investment decision,” *Energy Build.*, vol. 61, pp. 81–87, Jun. 2013.

[159] S. Short, “Review of the UK 2000 Time Use Survey,” Jul. 2006.

[160] W. K. Hastings, “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[161] O. B. Augusto, F. Bennis, and S. Caro, “Multiobjective engineering design optimization problems: a sensitivity analysis approach,” *Pesqui. Oper.*, vol. 32, no. 3, pp. 575–596, 2012.

[162] J. Wang, Z. (John) Zhai, Y. Jing, and C. Zhang, “Particle swarm optimization for redundant building cooling heating and power system,” *Appl. Energy*, vol. 87, no. 12, pp. 3668–3679, 2010.

[163] M. Wetter, “GenOpt - A Generic Optimisation Program,” in *Proc. IBPSA’s Building Simulator*, 2001.

[164] K. Deb, S. Agrawal, and A. Pratap, “A fast Elitist Non-dominated Sorting Genetic Algorithm for multi-objective Optimization (NSGA II),” 2002.

[165] N. Chakraborti, B. Siva Kumar, V. Satish Babu, S. Moitra, and A. Mukhopadhyay, “A new multi-objective genetic algorithm applied to hot-rolling process,” *Appl. Math. Model.*, vol. 32, no. 9, pp. 1781–1789, 2008.

[166] US Department of Energy, “EnergyPlus Engineering Reference The Reference to EnergyPlus Calculations.” 2011.