Accelerated optimisation methods for low-carbon building design

Esmond Tresidder

Submitted in partial fulfilment of the requirements for the degree of DOCTOR OF PHILOSOPHY

Institute of Energy and Sustainable Development
De Montfort University, Leicester

Research funded by Technology Strategy Board and DesignBuilder Ltd.

March 2014
Abstract

This thesis presents an analysis of the performance of optimisation using Kriging surrogate models on low-carbon building design problems. Their performance is compared with established genetic algorithms operating without a surrogate on a range of different types of building-design problems. The advantages and disadvantages of a Kriging approach, and their particular relevance to low-carbon building design optimisation, are tested and discussed. Scenarios in which Kriging methods are most likely to be of use, and scenarios where, conversely, they may be disadvantageous compared to other methods for reducing the computational cost of optimisation, such as parallel computing, are highlighted.

Kriging is shown to be able, in some cases, to find designs of comparable performance in fewer main-model evaluations than a stand-alone genetic algorithm method. However, this improvement is not robust, and in several cases Kriging required many more main-model evaluations to find comparable designs, especially in the case of design problems with discrete variables, which are common in low-carbon building design. Furthermore, limitations regarding the extent to which Kriging optimisations can be accelerated using parallel computing resources mean that, even in the scenarios in which Kriging showed the greatest advantage, a stand-alone genetic algorithm implemented in parallel would be likely to find comparable designs more quickly. In light of this it is recommended that, for most low-carbon building design problems, a stand-alone genetic algorithm is the most suitable optimisation method.

Two novel methods are developed to improve the performance of optimisation algorithms on low-carbon building design problems. The first takes advantage of variables whose impact can be quickly calculated without re-running an expensive dynamic simulation, in order to dramatically increase the number of designs that can be explored within a given computing budget. The second takes advantage of objectives that can be calculated without a dynamic simulation in order to filter out designs that do not meet constraints in those objectives and focus the use of computationally-expensive dynamic simulations on feasible designs. Both of these methods show significant improvement over standard methods in terms of the quality of designs found within a given dynamic-simulation budget.
## Contents

List of Figures .................................................. x
List of Tables ...................................................... xiv
Abbreviations ...................................................... xvi
Glossary .......................................................... xvii
Author declaration .............................................. xxi
Acknowledgements ................................................ xxii

### I  Low-carbon building design and the role of optimisation  1

#### 1  Context ...................................................... 2

1.1 Motivations for reducing the environmental impact of buildings . . . 2
1.1.1 Global context .............................................. 2
1.1.2 Buildings and the role they can play in tackling climate change 4
1.2 The challenge facing designers of low-carbon buildings .............. 5

#### 2  Searching for optimal designs using optimisation algorithms 11

2.1 Formulation of the optimisation problem .......................... 11
2.2 Genetic algorithms ............................................. 12
2.3 Genetic operations driving change in a population of building designs 14
2.3.1 Replication .................................................. 14
2.3.2 Crossover .................................................... 14
2.3.3 Mutation ..................................................... 15
2.3.4 Improvements in genetic algorithm efficiency ............... 16
2.4 Efficiently exploring the design space ............................ 16
2.4.1 Models of models .......................................... 17
A note on the complexity of calculations involved in the Kriging optimisation process

Barriers to the implementation of Kriging optimisation methods in building-design

Discrete variables and discontinuities in building-design problems

Other limitations of Kriging optimisation

Limitations in the number of variables that can be considered

Limitations in the total number of simulations that can be run

Limitations in exploiting parallel-computing resources

Summary of challenges and rationale for further research

How should the comparison between Kriging optimisation and main-model optimisation be made?

Summary

Comparisons of performance between Kriging optimisation and stand-alone genetic algorithms on low-carbon building design problems

Methods for all experiments

Building models for EnergyPlus simulations

Calculating cost and CO₂ objectives

Choice of simulations to perform in each method

Optimisation using Kriging surrogate models

Interaction between Matlab, jEPlus, cost and CO₂ models

Algorithm settings used for optimisation using Kriging surrogate models

Coding of discrete variables in Kriging

Reducing the time overhead associated with Kriging

Optimisation using jEPlus+EA

Settings used for optimisation using jEPlus+EA

Determination of optima through brute-force analysis

Comparing the performance of different multi-objective optimisation methods

Calculation of mean Euclidian distance

Calculating the area dominated by estimates of the Pareto front
10.6.1.2 Finding the equivalence-point for the performance of Kriging and jEPlus+EA ........................................ 129
10.6.2 Moderately-discrete optimisation ..................................... 132
  10.6.2.1 Performance at the stopping point of the Kriging optimisation ........................................ 132
  10.6.2.2 Finding the equivalence-point for the performance of Kriging and jEPlus+EA ................................. 133
  10.6.2.3 Relative performance of the two methods at their respective stopping points ............................... 134
10.6.3 Moderately-continuous optimisation .................................. 135
  10.6.3.1 Performance at the stopping point of the Kriging optimisation ........................................ 135
  10.6.3.2 Relative performance of the two methods over the course of the optimisation ............................. 136
  10.6.3.3 Relative performance of the two methods at their respective stopping points ............................... 136
10.6.4 Highly-continuous optimisation ....................................... 137
  10.6.4.1 Performance at the stopping point of the Kriging optimisation ........................................ 137
  10.6.4.2 Relative performance in dominated area of the two methods over the course of the optimisation ..... 139
  10.6.4.3 Relative performance of the two methods at their respective stopping points ............................... 140
10.6.5 Comparing the advantage offered by Kriging on design problems with discretised continuous variables at different resolutions 140
10.6.6 Optimisation with variables that are discrete by nature .... 143

10.7 Discussion ........................................................................ 144
  10.7.1 Analysis of the fidelity of the Kriging models on different optimisation problems ............................. 146
  10.7.2 Worst and best-case scenarios for Kriging performance ................................................................. 148
10.8 Conclusions ...................................................................... 149

11 Defining the range of Kriging performance on low-carbon building design problems 151
  11.1 Overview of methods ....................................................... 151
  11.2 Introduction ..................................................................... 152
  11.3 Method ............................................................................ 154
  11.4 Results ............................................................................. 154
V Other investigations performed during the thesis 166

12 Man versus machine 168

12.1 Overview of methods 168
12.2 Introduction 169
12.3 Optimisation without optimisation algorithms 169
  12.3.1 Simplifying the design-problem 170
12.4 Results 172
12.5 Discussion 174
12.6 Conclusions 176

13 Applying the students’ insights to optimisation algorithms 177

13.1 Overview of methods 178
13.2 Demand-altering and non-demand-altering variables 178
13.3 Current approach to solving design-problems with a mix of demand-altering and non-demand-altering variables 179
13.4 Proposed method for handling non-demand-altering variables 181
13.5 Results 185
13.6 Discussion 187
13.7 Future work 190

14 Further exploitation of fast-to-calculate shortcuts 192

14.1 Overview of methods 193
14.2 Carbon and cost budgets 193
14.3 Constraint-handling methods 194
14.4 Proposed method for handling constraints in computationally-cheap objectives within NSGA-II 197
14.5 Experimental method 200
14.6 Results 201
14.7 Discussion 204
14.8 Conclusions 207
VI Whole thesis conclusions and future work 209

Context ................................................................. 210
Conclusions regarding aims and objectives .......................... 210
Future work ............................................................... 213

Appendix A Building models used for simulation in EnergyPlus 217
A.1 Building model used for experiments in Chapter 8 ............ 217
A.2 Building models used for experiments in Chapters 9, 12, 13 and 14 . 219
A.3 Building model used for experiments in Chapters 10 and 11 ... 224

Appendix B Cost and CO$_2$ models used throughout the thesis 227
B.1 Calculating annual CO$_2$ emissions from gas and electricity use . . 228
B.1.1 CO$_2$ factors used for the experiments described in Chapter 8 . 229
B.1.2 CO$_2$ factors used for the experiments described in Chapters 9, 12, 13 and 14 . 229
B.1.3 CO$_2$ factors used for the experiments described in Chapters 10 and 11 .............................................. 231
B.1.4 HVAC efficiency .................................................. 231
B.2 Calculating embodied CO$_2$ emissions ............................ 232
B.3 Cost models used in Chapter 8 .................................... 233
B.4 Cost models used in Chapters 9, 12, 13 and 14 .................. 234
B.5 Cost models used in Chapters 10 and 11 ........................ 236

Appendix Bibliography .................................................. 237

Appendix C Publications ................................................. 245
How to integrate optimization into building design practice: Lessons learnt from a design optimization competition ........................................ 245
Acceleration of building design optimisation through the use of Kriging surrogate models ................................................. 255
Optimisation of low-energy building design using surrogate models . . . 263
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>An example of a Pareto front in a two-objective solution space.</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>An outline of the process of a simple genetic algorithm.</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>An example of a simple crossover-operator.</td>
<td>15</td>
</tr>
<tr>
<td>2.3</td>
<td>An example of a simple mutation-operator.</td>
<td>15</td>
</tr>
<tr>
<td>6.1</td>
<td>Parameters of the Kriging model and how they affect its shape.</td>
<td>48</td>
</tr>
<tr>
<td>6.2</td>
<td>A workflow showing the different stages of the Kriging optimisation process as implemented in this thesis.</td>
<td>52</td>
</tr>
<tr>
<td>6.3</td>
<td>The time taken per loop of the Kriging optimisation increases with the total number of sample points.</td>
<td>54</td>
</tr>
<tr>
<td>7.1</td>
<td>An overview of the interaction between Matlab, jEPlus, cost and CO2 models used in the Kriging optimisation.</td>
<td>70</td>
</tr>
<tr>
<td>7.2</td>
<td>Calculation of the Euclidian distance to the nearest point on the true Pareto front.</td>
<td>77</td>
</tr>
<tr>
<td>7.3</td>
<td>An explanation of why Euclidian distance can be misleading in design spaces with large discontinuities.</td>
<td>78</td>
</tr>
<tr>
<td>7.4</td>
<td>A visual explanation of the calculation of the area dominated by a set of Pareto estimates.</td>
<td>79</td>
</tr>
<tr>
<td>7.5</td>
<td>Comparison of performance using dominated area is robust to discontinuities in the true Pareto set.</td>
<td>80</td>
</tr>
<tr>
<td>7.6</td>
<td>Improvements in Pareto-estimates at either end of the Pareto front contribute disproportionately to the dominated area.</td>
<td>81</td>
</tr>
<tr>
<td>8.1</td>
<td>Mean progression of the best-performing Kriging and stand-alone single-objective optimisations.</td>
<td>91</td>
</tr>
<tr>
<td>8.2</td>
<td>Progression of the Kriging and jEPlus+EA on a multi-objective design problem.</td>
<td>100</td>
</tr>
<tr>
<td>8.3</td>
<td>Pareto estimates made by all runs of both Kriging and jEPlus+EA.</td>
<td>101</td>
</tr>
<tr>
<td>9.1</td>
<td>Progression of Kriging and jEPlus+EA optimisations on a more complex design problem.</td>
<td>112</td>
</tr>
</tbody>
</table>
9.2 All Pareto estimates for both methods after 310 iterations.

10.1 All Pareto estimates made by jEPlus+EA and Kriging after 260 main-model evaluations on the highly-discrete design problem.

10.2 Progression of Kriging and jEPlus+EA optimisations on the highly-discrete design problem.

10.3 All Pareto estimates from Kriging after 260 main-model evaluations and jEPlus+EA after 600 on the highly-discrete experiment.

10.4 All Pareto estimates made by jEPlus+EA and Kriging after 260 main-model evaluations on the moderately-discrete design problem.

10.5 Progression of Kriging and jEPlus+EA on the moderately-discrete design problem.

10.6 All Pareto estimates from Kriging after 260 main-model evaluations and jEPlus+EA after 600 on the moderately-discrete experiment.


10.8 Progression of Kriging and jEPlus+EA on the moderately-continuous design problem.

10.9 All Pareto estimates made by jEPlus+EA and Kriging on the moderately-continuous design problem.


10.11 Progression of Kriging and jEPlus+EA on the highly-continuous design problem.


10.13 All Pareto estimates made by jEPlus+EA and Kriging after 260 main-model evaluations on the design problem with variables that were discrete by nature.

10.14 Progression of Kriging and jEPlus+EA on the design problem with variables that are discrete by nature.

10.15 Kriging predictions versus main-model predictions for the cost and CO₂ models for all 5 experiments in this chapter.

11.1 Extended progression of Kriging and jEPlus+EA on the highly-continuous design problem.

11.2 Advantage offered by Kriging at different points in the optimisation process on the highly-continuous design problem.

11.3 Estimated Pareto designs from all runs of Kriging and jEPlus+EA optimisations.
12.1 A comparison of estimates of the best designs at three different price points by Kriging, jEPlus+EA and students optimising without computational optimisation methods. ................................. 173

13.1 An outline of the methodology used to add in the effect of non-demand-altering variables at the end of the optimisation process. ................................. 183

13.2 The addition of the non-demand-altering variables increases the size and range of the Pareto front. ......................................................... 184

13.3 All Pareto estimates made by the old and new methods of handling non-demand-altering variables. ......................................................... 185

13.4 Each section of the Pareto front from Figure 13.3 examined in more detail. ......................................................... 186

13.5 The new method shows an improvement in terms of both the mean area dominated by Pareto estimates, and the mean number of Pareto estimates made. ......................................................... 188

13.6 Some of the designs on the Pareto front after the inclusion of non-demand-altering variables are not on the Pareto front being searched for by the algorithm. ......................................................... 190

14.1 Standard methods for handling constraints in NSGA-II can result in a reduction in the influence of crossover and mutation operators. .................. 196

14.2 Modification of population initiation to screen for individuals that do not meet the constraint criteria. ......................................................... 198

14.3 Modification of the crossover operator to screen for individuals that do not meet the constraint criteria. ......................................................... 199

14.4 Modification of the mutation operator to screen for individuals that do not meet the constraint criteria. ......................................................... 200

14.5 The proposed constraint-handling method appears to improve on the quality of Pareto estimates in the constrained region. ......................................................... 202

14.6 Progression of the two optimisations with different constraint handling mechanisms. ......................................................... 202

14.7 The new method finds more designs on the estimated Pareto-front than the standard NSGA-II constraint handling method. ......................................................... 204

14.8 Generating a proxy target in the computationally-cheap-to-assess objective from a target in the expensive-to-assess objective. ......................................................... 206

A.1 Building model used in experiments in Chapter 8 ................. 218

A.2 Construction elements used in experiments in Chapter 8 ................. 219

A.3 Square floor plan building model used in Chapters 9, 12, 13 and 14. ................. 220

A.4 Narrow floor plan building model used in Chapters 9, 12, 13 and 14. ................. 221
A.5 Wide floor plan building model used in Chapters 9, 12, 13 and 14. . . 222
A.6 Lightweight and straw-bale external wall construction types. . . . . . 223
A.7 External floor and roof constructions used in Chapters 9, 12, 13 and
14. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 224
A.8 Building model used in Chapters 10 and 11. . . . . . . . . . . . . . . 225

B.1 Future CO₂ emissions estimates used to calculate 21-year in use emis-
sions. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 230
## List of Tables

1.1 Different combinations of design variables and choices and their effect on the total number of possible designs. ........................................ 8

2.1 Representation of design choices as genes. ........................................ 13

3.1 Overview of thesis structure ............................................................ 21

7.1 Settings for the genetic algorithm used on the Kriging model and for the single-objective stand-alone GA. .............................................. 71

7.2 Genetic algorithm settings used in jEPlus+EA ..................................... 74

8.1 A summary of the experimental methods used in Chapter 8. ............... 84

8.2 Design variables used in the first set of experiments. .......................... 86

8.3 Results of single-objective optimisations with and without using a Kriging model ................................................................. 90

8.4 Results of different configurations of the Kriging optimisations. .......... 98

8.5 The effect of shortening the encoding length on the number of duplicate designs selected. .......................................................... 98

9.1 A summary of the experimental methods used in Chapter 9. ............... 107

9.2 Variable options available in the design problem. .............................. 108

9.3 Progression of Kriging and jEPlus optimisations along with statistical significance of the differences in performance. ..................... 113

10.1 A summary of the experimental methods used in Chapter 10. .......... 119

10.2 Variable choices available for the highly-discrete design problem. ...... 123

10.3 Variable choices available for the moderately-discrete design problem. 124

10.4 Design variables used in the optimisation with variables that were discrete by nature. ........................................................... 125

10.5 Variables that were fixed in the optimisation on the design problem with variables that were discrete by nature ......................... 126
10.6 Relative advantage oﬀered by Kriging after 260 main-model evaluations, compared to jEPlus+EA after 600 evaluations . . . . . . . . . 142
10.7 Mean values for the diﬀerence between Kriging predictions and mainmodel values for the design problems with discretised continuous variables at diﬀerent resolutions. . . . . . . . . . . . . . . . . . . . . . . 148
11.1 A summary of the experimental methods used in Chapter 11. . . . . . 152
11.2 The range of performance of Kriging compared to stand-alone methods on all experiments. . . . . . . . . . . . . . . . . . . . . . . . . . . 153
12.1 A summary of the experimental methods used in Chapter 12. . . . . . 168
13.1 A summary of the experimental methods used in Chapter 13. . . . . . 178
14.1 A summary of the experimental methods used in Chapter 14. . . . . . 193
B.1 Coeﬃcients of performance for diﬀerent HVAC plant used for experiments. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 231
B.2 Embodied CO2 in construction materials used in the experiments
described in Chapter 8. . . . . . . . . . . . . . . . . . . . . . . . . . . 233
B.3 Costs for diﬀerent construction elements used in the experiments described in Chapter 8. . . . . . . . . . . . . . . . . . . . . . . . . . . 234
B.4 Costs for the various materials used in the experiments in used in
Chapters 9, 12, 13 and 14 . . . . . . . . . . . . . . . . . . . . . . . . 235
B.5 Relevant materials costs for the optimisations described in Chapters
10 and 11. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 236

xv


### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR5</td>
<td>Fifth Assessment Report</td>
</tr>
<tr>
<td>ASHRAE</td>
<td>American Society of Heating, Refrigerating and Air Conditioning Engineers</td>
</tr>
<tr>
<td>ASHP</td>
<td>Air Source Heat Pump</td>
</tr>
<tr>
<td>CFL</td>
<td>Compact fluorescent</td>
</tr>
<tr>
<td>CO₂</td>
<td>Carbon Dioxide</td>
</tr>
<tr>
<td>DCLG</td>
<td>Department for Communities and Local Government</td>
</tr>
<tr>
<td>DECC</td>
<td>Department of Energy and Climate Change</td>
</tr>
<tr>
<td>DEFRA</td>
<td>Department for Environment, Food and Rural Affairs</td>
</tr>
<tr>
<td>EC</td>
<td>European Commission</td>
</tr>
<tr>
<td>EI</td>
<td>Expected Improvement</td>
</tr>
<tr>
<td>EPS</td>
<td>Expanded Polystyrene</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>HVAC</td>
<td>Heating, Ventilation and Air Conditioning</td>
</tr>
<tr>
<td>IDF</td>
<td>A type of EnergyPlus input file</td>
</tr>
<tr>
<td>IPCC</td>
<td>Intergovernmental Panel on Climate Change</td>
</tr>
<tr>
<td>kgCO₂</td>
<td>Kilograms of Carbon Dioxide</td>
</tr>
<tr>
<td>LED</td>
<td>Light emitting diode</td>
</tr>
<tr>
<td>LEED</td>
<td>Leadership in Energy and Environmental Design</td>
</tr>
<tr>
<td>Low-E</td>
<td>Low-emissivity</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>Non-dominated Sorting Genetic Algorithm 2</td>
</tr>
<tr>
<td>PV</td>
<td>Photovoltaic</td>
</tr>
<tr>
<td>RCP</td>
<td>Representative Concentration Pathways</td>
</tr>
<tr>
<td>SHW</td>
<td>Solar Hot Water</td>
</tr>
<tr>
<td>SVR</td>
<td>Support Vector Regression</td>
</tr>
<tr>
<td>TRNSYS</td>
<td>Transient System Simulation Tool</td>
</tr>
</tbody>
</table>
### Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area dominated</td>
<td>The area dominated by the Pareto front (or by estimates of it). Any design falling within this area is dominated by at least one design on the Pareto front.</td>
</tr>
<tr>
<td>Brute-force analysis</td>
<td>Testing every possible combination of design variables.</td>
</tr>
<tr>
<td>Chromosome</td>
<td>In the context of this thesis, a chromosome refers to a string of genes each representing a design variable.</td>
</tr>
<tr>
<td>CO(_2) factor</td>
<td>The amount of CO(_2) emitted in the production of a kWh of energy from a particular source (gas, electricity, coal etc.).</td>
</tr>
<tr>
<td>Design space</td>
<td>The total number of possible designs.</td>
</tr>
<tr>
<td>Design variables or decision variables</td>
<td>The variables that are to be changed in searching for the best designs.</td>
</tr>
<tr>
<td>Encoding length</td>
<td>The number of bits used to encode for numbers in the optimisation algorithms.</td>
</tr>
<tr>
<td>EnergyPlus</td>
<td>A BESTTESTed dynamic building-energy simulation engine developed by the US department of energy.</td>
</tr>
<tr>
<td>Epistatic</td>
<td>Two variables are said to be epistatic if the effect one variable has on the objective function depends, in part, on the value taken by the other variable.</td>
</tr>
<tr>
<td>Euclidian distance</td>
<td>The linear distance between two points in a chart.</td>
</tr>
<tr>
<td>Expected Improvement (also multi)</td>
<td>A value that can be obtained from the Kriging model. EI gives an estimate of the improvement expected over the current best design (or the current estimate of the Pareto front in the case of multi EI) at a location in the design space that has not already been tested on the main-model.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Exploration</td>
<td>Searching the design space in a way that maximises knowledge about the whole space.</td>
</tr>
<tr>
<td>Exploitation</td>
<td>Searching the design space in a way that maximises knowledge about a specific region of the design space.</td>
</tr>
<tr>
<td>Fitness function</td>
<td>See objective function.</td>
</tr>
<tr>
<td>Fixed-base models</td>
<td>A type of surrogate model in which the width parameter for each variable is fixed.</td>
</tr>
<tr>
<td>fmincon</td>
<td>A gradient-based, deterministic search algorithm used to search for superior designs in a small area of the design space.</td>
</tr>
<tr>
<td>Gaussian-basis function</td>
<td>A type of surrogate model in which the distribution of values around a sample point is determined by a gaussian probability distribution. Kriging is a type of Gaussian-basis function.</td>
</tr>
<tr>
<td>Genetic drift</td>
<td>The loss of genetic diversity over time due to random sampling, often exacerbated by small population sizes. In terms of genetic algorithms this can mean the loss of promising individuals (see Pareto archive).</td>
</tr>
<tr>
<td>jEplus</td>
<td>A batch processing tool for running EnergyPlus simulations.</td>
</tr>
<tr>
<td>jEplus+EA</td>
<td>jEPlus with an evolutionary algorithm (in this case based on NSGA-II) to enable optimisations to be run for EnergyPlus based building models.</td>
</tr>
<tr>
<td>Likelihood</td>
<td>The probability that the observed values could have come from a Kriging model with a given set of parameters. When searching for the best Kriging model to match the sample data, the genetic algorithm tries to maximise the likelihood of the data.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Main-model evaluations</td>
<td>Evaluations of the objective function using the main model, as opposed to using the Kriging surrogate model.</td>
</tr>
<tr>
<td>Morris-Mitchell Latin Hypercube</td>
<td>A sampling strategy that aims to sample as evenly as possible across a multi-dimensional design space. Defined as the set of designs that best cover the k-dimensional design space, achieved by maximising the mean Euclidian distance between designs.</td>
</tr>
<tr>
<td>Objective Function</td>
<td>A function that calculates the value of one of the objectives in an optimisation.</td>
</tr>
<tr>
<td>Pareto archive</td>
<td>An archive that stores Pareto-optimal solutions from one generation to the next so that they are not lost through genetic drift, a problem that can occur in genetic algorithms with small population sizes and without archiving strategies.</td>
</tr>
<tr>
<td>Passivhaus</td>
<td>A low-energy building design standard developed in Germany that has become popular in recent years.</td>
</tr>
<tr>
<td>Pareto front</td>
<td>The edge of the design space occupied by designs in the Pareto set.</td>
</tr>
<tr>
<td>Pareto set</td>
<td>The set of non-dominated designs. These designs represent the best trade-off between competing objectives.</td>
</tr>
<tr>
<td>Penalty function</td>
<td>A function used to penalise designs that fail to meet certain criteria, in order to guide the optimisation away from such designs.</td>
</tr>
<tr>
<td>Polynomial model</td>
<td>A type of linear regression model.</td>
</tr>
<tr>
<td>Solar heat gain coefficient</td>
<td>A measure of the solar transmittance of a material, typically used to describe solar transmittance characteristics of glazing.</td>
</tr>
</tbody>
</table>
U value A measure of the insulation value of a construction, defined as the heat transfer in Watts through a m² of that construction per degree Kelvin difference in temperature between either side of the construction. A low U value means that the construction is well insulated.
Author declaration

During the period of registered study leading to the preparation of this thesis the author has not been registered for any other academic award or qualification.

The material is the author’s own work and has not been submitted wholly or in part for any other academic award or qualification.

Name
Date
Acknowledgements

Thanks to the Technology Strategy Board for providing the funding for this research.
Design Builder for their generous additional funding, and to Andy Tindale and Gina Mann for their support and assistance throughout the project.
Richard Snape, Stephen Porritt and Ivan Koroliya at IESD for their invaluable assistance with technical problems.
My supervisors Yi Zhang and Simon Rees, for their guidance and critical eye.
Bobby Gilbert for setting the ball rolling.
Alexander Forrester for starting me on the path of optimisation methods, and for his continuing assistance throughout the thesis.
My wife Hilary for her unending support, words of wisdom and wisdom with words. My son Aaron for keeping me smiling and my unborn son for not arriving before the work was finished.
Part I

Low-carbon building design and the role of optimisation
Chapter 1

Context

This first chapter places low-carbon buildings in a global context and outlines the complexity of designing low-carbon buildings.

1.1 Motivations for reducing the environmental impact of buildings

1.1.1 Global context

The evidence that humankind’s emissions of greenhouse gases are causing a warming of the world’s climate is extremely strong. The most recent assessment report (AR5) from the Intergovernmental Panel on Climate Change (IPCC) issued a series of statements, each with an assessment of the scientific confidence associated with that statement (IPCC, 2013):

\[
\text{Each of the last three decades has been successively warmer at the Earth’s surface than any preceding decade since 1850. In the Northern Hemisphere, 1983–2012 was likely the warmest 30-year period of the last 1400 years.}
\]
And:

*It is extremely likely that human influence has been the dominant cause of the observed warming since the mid-20th century.*

Furthermore:

*Continued emissions of greenhouse gases will cause further warming and changes in all components of the climate system. Limiting climate change will require substantial and sustained reductions of greenhouse gas emissions.*

AR5 goes on to investigate the effect on the global climate of four future scenarios for greenhouse-gas concentrations, called Representative Concentration Pathways (RCPs 2.6, 4.5, 6.0 and 8.5). Of these four pathways, all except the scenario involving the most ambitious emissions reductions (RCP 2.6) are judged likely (defined as greater than 66% probability) to result in global surface temperature change in excess of 1.5°C by the end of the 21st century. The two higher-emissions scenarios (RCP 6.0 and 8.5) are likely to exceed 2°C and RCP 4.5 is more likely than not (greater than 50% probability) to exceed 2°C. Temperature changes above 1.5 or 2°C are widely thought to be likely to result in dangerous levels of destabilisation to the climate and earth systems.

The lower two of these emissions pathways are extremely ambitious; RCP 2.6 involves stabilisation of atmospheric greenhouse-gas concentrations in the 2040s, and falling concentrations thereafter, whereas RCP 4.5 sees a stabilisation of atmospheric concentrations in the late 21st century. These will require global annual emissions of greenhouse gases to drop to zero, and in the case of RCP 2.6, to become negative (i.e. for there to be a global net sequestration of CO₂). Furthermore, the authors note that:
Accounting for warming effects of increases in non-CO₂ greenhouse gases, reductions in aerosols, or the release of greenhouse gases from permafrost will also lower the cumulative CO₂ emissions for a specific warming target.

It should be noted that The IPCC assessment reports are a gauge of the current consensus on climate science and many climate scientists believe more ambitious emissions reductions are necessary (see for example Hansen et al. (2008)).

1.1.2 Buildings and the role they can play in tackling climate change

As the scientific consensus on climate change has become increasingly solid, governments around the world have issued legislation and targets for reductions in their own emissions of greenhouse gases. At the European level these include legally binding targets of a 20% reduction in EU greenhouse-gas emissions from 1990 levels by 2020 (EC, 2007), and a target of an 80% reduction by 2050 (EC, 2011). The UK Government is legally obliged to meet the 2020 target, but furthermore has made an 80% reduction in greenhouse-gas emissions by 2050 a legally binding target (DECC, 2008).

Improvements in the energy performance of both new and existing buildings can and should play an important role in meeting these targets. In the UK in 2009, energy use in buildings accounted for about 43% of UK CO₂ emissions (DCLG, 2009). Progressively more ambitious targets are in place for both domestic and non-domestic new buildings, and substantial work will also be required to improve the energy performance of existing building stock (DCLG, 2010).

Improvements in the energy efficiency of buildings above the standards of current building stock may come relatively easily. However, as the energy targets become
increasingly ambitious, meeting those targets in a cost-effective way also becomes more challenging.

It is with this background of increasingly firm scientific consensus and increasingly stringent building-energy targets that designers of today and the future must approach the task of designing their buildings.

1.2 The challenge facing designers of low-carbon buildings

The optimisation of low-carbon building design is a complex problem. If the designer is concerned only with a single performance objective, such as annual CO\textsubscript{2} emissions, and if the effect each design choice (each variable) has on the performance is independent of the values chosen for other variables, finding the optimum design may be relatively trivial. However, such a situation is rare. Even if there is only a single performance objective the optimum specification for each variable often depends on the choices made for other variables, that is to say the effect of each variable is epistatic. Furthermore, it is hard to imagine a situation in which the designer is concerned about improving energy performance or reducing CO\textsubscript{2} emissions but is unconcerned about the cost of the resulting design. Therefore in most low-carbon building design cases there will be at least two performance objectives; cost and energy performance.

The design of low-carbon buildings becomes more complicated still since the design may be subject to legal and project-specific constraints such as;

- performance specifications that must be met for different building elements, such as U values, building airtightness and glazing ratios

- comfort criteria
• building shape and layout

and to multiple objectives such as;

• in-use CO₂ emissions
• embodied CO₂ emissions
• life-cycle environmental impact
• energy use per unit of floor area (for example to meet Passivhaus standard)
• project cost
• building aesthetics

The consideration of future scenarios regarding climate change, deterioration of building fabric and changes in usage patterns further complicates the decision making process.

The best trade-off between competing objectives is defined by the Pareto front. For each design on the Pareto front the performance in one objective can be improved only by a decrease in the performance of the other objective(s). Designs that are not on the Pareto front are said to be dominated by those designs that are on the front, that is to say there is at least one design that is better on one objective and at least as good on the other objective(s) (Figure 1.1).
Figure 1.1 – A hypothetical solution space, with the Pareto designs marked in red. For both objectives minimisation is desired. The designs are grouped into three categories (A, B and C), according to which design dominates that portion of the solution space. Designs A1 and B1 dominate designs A2 and B2 respectively because they are superior on both objectives. Design C1 dominates both C2 and C3. Designs on the Pareto front are not dominated by any other designs and are said to be non-dominated.

For a given building project there may be several hundred variables which are important to the performance of the building. Many of these will be fixed early on in the design process through site constraints or other design considerations. The variables which are not fixed at this early stage become the focus of the optimisation. Different authors have referred to these variables using different terms (variables, design variables, decision variables). In common with several other studies (see for example Sasena et al. (2002), Hamdy et al. (2011) and Jin and Overend (2012)) this thesis uses the term *design variables* to refer to those variables that the designer can change with the objective of improving the performance of the design against one or more objectives. The more design variables there are, and the more choices available for each design variable, the bigger the design space (the set of all possible designs) that must be searched for good designs. The total number of possible designs (the design space) is given by the formula
where \( D \) is the total number of possible designs, \( k \) the number of design variables and \( C \) the number of choices for each variable (assuming each variable has the same number of possible choices). Putting some example numbers into Equation 1.1 illustrates how the design space can easily become extremely large (Table 1.1).

\[
D = C^k
\]  

(1.1)

<table>
<thead>
<tr>
<th>Number of design variables ((k))</th>
<th>Number of choices for each variable ((C))</th>
<th>Total number of possible designs ((D))</th>
<th>Time required for comprehensive evaluation if one dynamic simulation takes 60 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>243</td>
<td>4 hours</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>59,049</td>
<td>41 days</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>60,466,176</td>
<td>115 years</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
<td>(3.7 \times 10^{15})</td>
<td>(7.0 \times 10^9) years</td>
</tr>
</tbody>
</table>

**Table 1.1** – As either the number of design variables or the number of choices for each variable increases, the total number of possible designs increases exponentially in response according to Equation 1.1.

Linear changes in the number of design variables, or choices for each variable, drive exponential changes in the size of the design space because each additional variable adds another dimension to the design space. With regard to its effect on the difficulty of solving design problems, this has been called *The Curse of Dimensionality* (Forrester et al., 2008).

The evaluation of performance against one or more of the objectives (the evaluation of objective-functions) in building design is often time consuming. Dynamic simulation is typically used to estimate annual building energy demand (and from this \(\text{CO}_2\) emissions). These models involve a very large number of computations to determine the heat flows in and out of a building for each time step over the course of time the model is to be evaluated (often four or more time steps per hour and
a whole year of evaluation). The more complex a building the more surfaces these calculations have to be performed for, and the more time-consuming they become. Run times for the dynamic simulation of buildings vary enormously depending on the complexity of the model and the level of detail chosen for the simulation; from under a minute to many hours. Even in cases with building designs that are relatively quick to evaluate, a brute-force approach to optimisation, i.e. evaluating every possible design, may be infeasible (Naboni et al. (2013), Wang et al. (2005a) and Salminen et al. (2012)). This is illustrated by calculating the total time required to evaluate all possible designs in a hypothetical situation in which each dynamic simulation takes 60 seconds (Table 1.1).

The use of computers with multiple processor cores may make this task more feasible, but even access to a cluster capable of simultaneously running hundreds of simulations would not be enough to allow full exploration of the two largest design spaces in Table 1.1 in a timescale typical of a building-design project.

In these hypothetical examples the design space has already become too large to explore comprehensively. It should be noted that in real-world design situations (as opposed to the hypothetical situations described above), design spaces may be much larger, especially at the detailed-design stage, and evaluation by dynamic simulation can take considerably longer than one minute per model. In order to effectively search for the best trade-off between competing objectives, other methods must be used.

The larger the size of the design space the more challenging the task of searching for the optimum trade-off between objectives; of making good estimates of the Pareto front. This is the so-called curse of dimensionality. The implication of this is clear – design variables should be included in the optimisation only if they are both important in the determination of the performance of the building and if their optimum assignation cannot easily be determined outside of a multi-dimensional optimisation. Some existing research has focussed on how to choose which variables
to include in an optimisation in order that the design space is not too large (see for example Evins et al. (2012b), Eisenhower et al. (2012) and Wang et al. (2013)).

Traditionally, if optimisation of the building design has consciously been adopted as a goal at all, designers have approached complex questions such as these with a combination of tactics. These tactics may include experience of what has worked or not worked from previous projects, rules of thumb about a hierarchy of variables on which a limited budget should be spent, rules of thumb and regulations regarding the minimum performance of different elements of the building design, systematically investigating the effect of changing each variable in isolation, and trial and error (Coley and Schukat (2002), Wang et al. (2005b), Roy et al. (2008), Bichiou and Krarti (2011), Naboni et al. (2013) and Alsaadani and Souza (2012)).

Given the complex nature of many building design problems, these methods may be both very time consuming and ineffective at finding a good trade-off between competing design objectives (Wang et al. (2005b), Bichiou and Krarti (2011) and Evins (2012)). In order to effectively search for optimal designs in situations with large design spaces and time-consuming model evaluation, designers have turned to more sophisticated methods.
Chapter 2

Searching for optimal designs using optimisation algorithms

This chapter starts by presenting a generic representation of optimisation problems before describing methods to search for optimal designs using genetic algorithms and introducing surrogate models as a method to improve the search efficiency of optimisation methods.

2.1 Formulation of the optimisation problem

Optimisation algorithms can be applied to design problems with a wide variety of different design variables and objectives. A generic representation of optimisation problems is presented below.

A generic single-objective optimisation can be represented as follows:

\[
\text{minimise (or maximise) } f(\vec{X})
\]

where is \( \vec{X} \) is the vector of \( k \) design variables \([X_1, X_2, \ldots, X_k]\)

This single-objective case can be extended to a multi-objective (of \( t \) objectives) situation as follows:
minimise (or maximise) $\vec{f}(\vec{X}) = [f_1(\vec{X}), f_2(\vec{X}), ..., f_t(\vec{X})]$

Both the single and multi-objective optimisations can take place either unconstrained (as above) or subject to one or more constraints which can be represented as follows:

$g(\vec{X}) \leq 0$ (inequality constraint)

and $h(\vec{X}) = 0$ (equality constraint)

### 2.2 Genetic algorithms

Methods for finding the optimum trade-off between competing objectives have been the subject of much recent research and numerous methods exist. These include particle swarm, gradient-based searches, pattern searches, neural networks, simulated annealing and genetic algorithms. A comparison of these different methods is beyond the scope of this thesis, which focuses on genetic algorithms as a well-established and robust method. Genetic algorithms attempt to mimic the natural processes of evolution by natural selection in order to improve the designs from an initial starting population. Although many different types and configurations of genetic algorithm exist, they all follow a similar general methodology. This is summarised in Figure 2.1.

Each design variable (i.e. all those variables that are not already fixed) can be represented by a number, and that number can change according to the value chosen for that variable. So one variable might be heating system type, and that might be represented as shown in Table 2.1.

A design problem with 10 design variables would thus be represented by a string of 10 integers to form a chromosome. There are various different encoding methods, the one described here is integer encoding and is sufficient for understanding the basic
Figure 2.1 – Although different algorithms exist with minor differences (which may result in significant changes in performance), the basic framework for a genetic algorithm is as above. Each loop represents a new generation. The algorithm may be set to run for a given number of generations, or until other stopping criteria have been met.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Heating system type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Gas boiler</td>
</tr>
<tr>
<td>1</td>
<td>Air-source heat pump</td>
</tr>
<tr>
<td>2</td>
<td>Ground-source heat pump</td>
</tr>
</tbody>
</table>

Table 2.1 – Different choices for a variable can be represented by a different number, a gene.
processes of a genetic algorithm. The following section examines what happens at each stage of the algorithm.

Individuals from the initial population are selected to reproduce through a simple tournament system. Individuals are randomly paired, and from this pair the individual that shows the best performance against objectives (the greatest fitness) is chosen to progress to the next population. Various different methods exist for determining fitness, and these are discussed in more detail in later Sections (6.1.4.1 and 7.5.1).

2.3 Genetic operations driving change in a population of building designs

Individuals that have been selected to go forward to reproduce (through a tournament selection process) are then subject to either genetic replication, crossover or mutation. In each of these operations there is a probability that their genes will be either crossed over with the genes of a different parent, or that they will be mutated.

2.3.1 Replication

If neither crossover nor mutation takes place then the chromosome of the parent will simply be replicated. The offspring (the design going into the next population) will be the same as the parent.

2.3.2 Crossover

If crossover occurs then some genes from one parent are swapped with some genes from another parent. In Figure 2.2 a single-point simple crossover takes place on the third gene (the 4 of parent A is swapped for the 1 of parent B). The location
of the crossover point is usually random as is the selection of a second parent to be involved in the crossover.

![Diagram of crossover](image)

**Figure 2.2** – An example of a simple crossover-operator. The chromosome is eight genes long, and in this case the third gene (highlighted red) has been swapped between parent A and parent B, leading to two offspring that are similar, but not the same as, their parents.

### 2.3.3 Mutation

If mutation occurs then the genes of one parent are changed. The selection of the location for the change to take place, and the magnitude of the change, are chosen at random. In the example shown in Figure 2.3, a point mutation has taken place on the eighth (last) gene; 2 has been changed to 4.

![Diagram of mutation](image)

**Figure 2.3** – An example of a simple mutation-operator. In this case the eighth gene (highlighted red) has been randomly altered between the parent and the offspring.

Other types of crossover and mutation also exist which involve exchanging or changing more genetic material, or doing so in more sophisticated ways, but the purpose is the same – to produce offspring that are similar to the parents, but not the same.
Genetic algorithms may be set to run for a certain number of generations, or until other stopping criteria (such as whether the algorithm has found designs that meet pre-defined performance criteria) have been met.

The probabilities chosen for the genetic operators determine the behaviour of the algorithm. Algorithms with a very high chance of crossover or mutation will tend to perform more like a random search (because the offspring will tend to be very different from the parents) whereas algorithms with very low rates of crossover or mutation will take a long time to find optimum designs (because the population doesn’t change very much with each generation).

2.3.4 Improvements in genetic algorithm efficiency

Much research has been devoted to improving the efficiency of search algorithms, including genetic algorithms. The algorithm used in this thesis as a benchmark for comparisons of performance is the well known non-dominated sorting genetic algorithm II (NSGA-II). In addition to the general characteristics of genetic algorithms described above, NSGA-II employs a method for selecting parents from a mating pool that balances the fitness and diversity of solutions, meaning that the algorithm is less likely to become trapped in local optima (Deb et al. (2002)). The implementation used in this thesis also employs a pareto-archive similar to that used by Hamdy et al. (2012) to further improve the performance.

2.4 Efficiently exploring the design space

The use of genetic algorithms has been shown to allow good estimates of Pareto-optimal designs with far fewer design evaluations than would be required with a brute-force approach (see for example Wang et al. (2005a) and Verbeeck and Hens (2007)). Genetic algorithms are used in many fields where determination of the
best designs through brute-force computation is not feasible. Genetic algorithms such as NSGA-II (Deb et al., 2002) aim to balance exploitation of the design space (searching for the very best design in a given region) with exploitation (aiming to explore the whole design space and not get trapped in local optima).

However, the use of genetic algorithms to assist in building design is still far from widespread and is for the most part limited to academic research. Among the most important reasons for this is that, while evolutionary algorithms offer an efficient method for searching for optimum designs with fewer design evaluations than required by a brute-force approach, in many cases they can still require a prohibitively large number of design evaluations (Watson et al. (2013), Attia et al. (2013)). This is because the design evaluations tend to be time-consuming and, in optimisation problems with very large and complex design spaces even the use of a genetic algorithm may not reduce the number of evaluations required to make good estimates of optimal designs to the extent that such studies are feasible with the computing budget of a typical project.

2.4.1 Models of models

Even with the use of genetic algorithms, if the evaluation of performance against one or more objectives is very time consuming (as it is in optimisations requiring dynamic simulations) the number of evaluations required can still be prohibitive. A suggested method to reduce the number of main-model evaluations required to make good estimates of Pareto-optimal designs has been to build surrogate models of the main models and to run the genetic algorithm on these models in place of the main models. Surrogate models are mathematical models of the relationship between the design variables and the objective functions (in the cases described so far the objective functions are CO₂ and cost). Crucially surrogate models are
faster to interrogate than the main models they replace, meaning that many more
generations of genetic algorithm can be run within a given computing time budget.

The application of surrogate models to reduce the number of design evaluations
required to make good estimates of optimal designs has been proposed and imple-
mented by several studies (Wetter and Wright (2004), Hopfe et al. (2012), Kim et al.
(2013) and Kang et al. (2013)), including previous work by the author (Sobester
et al., 2012).

Several different types of surrogate model exist and have been applied to optim-
isation problems and these are discussed in more detail in Chapter 5. This thesis
focusses on one particular surrogate modelling technique that has already begun
to be applied in the field of building performance research, that of Kriging, and
investigates its applicability to low-carbon building design optimisation.
Chapter 3

Aims, objectives and major contributions

This chapter states the aims and objectives of this thesis and outlines its research methods, major contributions and structure.

Although the use of surrogate models to assist in building design optimisation has been applied by several studies, whether the use of such surrogates improves on standard evolutionary algorithms, and what the relative advantages and disadvantages of a surrogate-optimisation approach compared to a GA operating on the main models might be for low-carbon building design optimisation has not been thoroughly investigated. This investigation forms the main work of this thesis.

3.1 Aims and objectives

The main aim of this thesis is to establish the extent to which the use of Kriging-surrogate optimisation methods can accelerate the optimisation of low-carbon building design problems, compared to existing non-surrogate optimisation methods.

In addition to the main aim, the following objectives are pursued in this research:
• to establish the which types of low-carbon building optimisation problems Kriging surrogate-optimisation methods are suitable for, if not all problems.

• To establish the particular advantages and disadvantages of Kriging optimisation methods, when used on low-carbon building optimisation problems.

• To establish what degree of improvement in designs can be expected through the use of optimisation methods in general, compared to traditional design methods.

• To develop other methods to further accelerate or improve the optimisation of low-carbon building design, based on observations and lessons learnt throughout the research.

3.2 Research methods

The research methods used to answer the main aim of this thesis involve testing Kriging optimisation methods against non-surrogate optimisation methods on a range of low-carbon building design problems. The results are compared by the quality and number of designs found within a given budget of main-model samples. The methods used to determine the quality of the designs are discussed in more detail in later sections.

3.3 Major contributions

The major contributions made by this thesis are to;

• establish a likely range of performance advantage offered by Kriging, compared to optimisation without using surrogate models and discuss whether the advantage offered outweighs the benefits.
• introduce and test two new methods for accelerating the optimisation of low-carbon building designs.

• make a comparison between the quality of designs chosen by designers working without access to computational optimisation methods and those found by optimisation algorithms.

3.4 Structure of thesis

This thesis is organised as follows:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part I Low-carbon building design and the role of optimisation</td>
<td>Setting the scene, explaining the motivations behind the thesis and stating the aims and objectives</td>
</tr>
<tr>
<td>Part II Literature review</td>
<td>Reviewing major contributions in the field and identifying gaps in knowledge</td>
</tr>
<tr>
<td>Part III Optimisation using Kriging surrogate models</td>
<td>Introducing the Kriging methodology and discussing the potential advantages and disadvantages of this approach</td>
</tr>
<tr>
<td>Part IV Comparisons of performance between Kriging optimisation and stand-alone genetic algorithms on low-carbon building design problems</td>
<td>Establishing the range of performance offered by Kriging methods compared to stand-alone genetic algorithms on a variety of test cases</td>
</tr>
<tr>
<td>Part V Other investigations performed during the thesis</td>
<td>Comparing the results of optimisations by designers without optimisation algorithms with those found using optimisation algorithms. Introducing and testing two novel methods for accelerating the process of optimising low-carbon building designs</td>
</tr>
<tr>
<td>Part VI Whole thesis conclusions and future work</td>
<td>Making conclusions from the experiments performed with regard to the aims and objectives of the thesis.</td>
</tr>
</tbody>
</table>

Table 3.1 – Overview of thesis structure
Part II

Literature review
Chapter 4

Optimisation in low-carbon building design

Recognising that the design of low-carbon buildings is a complex optimisation problem, researchers have sought to apply various optimisation techniques to search for good designs. This chapter provides an overview of work that has been done in the field of building-design optimisation and comments on the limitations of current methods and how these might be addressed.

4.1 Iterative approaches

Early studies used iterative approaches to try and minimise the energy demand or CO₂ emissions from buildings, focussing on changing some variables initially, then others, and investigating the impact of these changes (see for example Rolfsman (2002), Peippo et al. (1991) and Gustafsson (2000)). These studies were limited in the total number of variables they could consider, and in their ability to handle many variables that were epistatic in nature.

Owing to these limitations, researchers began to be attracted to various optimisation methods that offered the prospect of handling many different, epistatic, variables
at once, and were able to effectively search for optimal designs in such complex design spaces. The problems of large design spaces, time-consuming performance evaluation and epistatic variables described in Section 1.2 are not, of course, limited to low-carbon building design optimisation, and optimisation techniques had been developed and implemented in other fields. Among the most effective and versatile of these methods are genetic algorithms. The use of genetic algorithms became widespread following the work of Holland (1975), and many examples of their use exist in other fields, for example in medicine (Sonka et al., 1996), automobile design (Baumal et al., 1998) and aerospace (Rauwolf and Coverstone-Carroll, 1996). The implementation of genetic algorithms has developed since their introduction, and more recent overviews of the methodology can be found in work by Bäck (1996) and Goldberg (2001).

4.2 Single-objective optimisation in low-carbon building design

In common with other fields, early applications of genetic algorithms used a single-objective approach. In instances where there was in fact more than one objective, these were often combined to form an aggregate objective that could be searched with a single-objective algorithm.

Coley and Schukat (2002) approached the multi-objective optimisation problem of minimising the annual energy use and maximising the architectural appeal of a village hall by combining a single-objective optimisation using a genetic algorithm on a thermal model of the building with a visual assessment of a selection of the resulting designs by a designer. The design variables were perimeter shape, roof pitch, construction details of walls, floor, roof and windows, window location, shading and building orientation (10 variables in all).
Wetter and Wright (2003) compared two different optimisation methods; the Hooke-Jeeves algorithm and a simple genetic algorithm on a single-objective optimisation of minimising the primary-energy consumption of an office building in three different locations with 13 different design variables. They found that the use of optimisation methods allowed them to reduce the primary-energy consumption of the building designs by between 7% and 32% compared to the base-case design. The range of possible reductions was due to different reductions being achievable in different locations. In a subsequent study (Wetter and Wright, 2004) they concluded that the simple GA was able to make very good estimates of the optimum with relatively few simulation calls, whereas a combination of the Hooke-Jeeves and particle-swarm methods achieved the best estimations of optimal designs but required significantly more simulations.

Wright and Alajmi (2005) used a building-energy optimisation problem as a test case to examine the sensitivity of a genetic algorithm to different parameter settings when the total number of design evaluations was limited (to 300). Their test case focussed on window geometry, shading and temperature set-point variables (13 design variables in all) and found that the algorithm was robust with regard to all parameter settings. A visual inspection of their results (Figure 3 of their paper) nevertheless suggests that the optimisations had still not reached an asymptote and therefore that 300 design evaluations was perhaps too small to make adequate estimates of optimum designs.

Hasan et al. (2008) optimised the design of a detached house for the single objective of lifecycle cost using a combination of a particle-swarm, population-based algorithm and a Hooke-Jeeves direct search algorithm. The design variables chosen were insulation thickness of the roof, walls and floor, U value of the windows and type of heat-recovery ventilation. Since all possible designs had been assessed in advance, the true optimum was known and the authors were able to conclude that this method was capable of finding or getting very close to the global minimum on
this test case. After the optimisation was run, the authors were able to compare the designs against annual energy demand and investment cost, to produce a Pareto front. The two solutions found to be optimal in the single-objective (lifecycle cost) case were very different in terms of initial investment cost, despite both being on the Pareto front. This short-falling illustrates the advantage of multi-objective optimisation; it is able to explore trade-offs between two or more performance objectives as part of the optimisation, rather than looking at these after the optimisation is finished (Wang et al. (2005a); Verbeeck and Hens (2007); Hopfe (2009)).

Simmons et al. (2013) compared the designs obtained by a single-objective optimisation of a combined metric of energy savings/cost premium to those obtained using prescriptive methodologies (building energy codes such as LEED, ASHRAE and Passivhaus) and found that optimisation methodologies resulted in much more cost-effective designs than those found using prescriptive methodologies.

4.3 Multi-objective optimisation in low-carbon building design

As multi-objective algorithms have become more commonplace in optimisation, and their benefits over single-objective algorithms recognised, they have also found their way into low-carbon building-design optimisation. The advantages of using a Pareto approach in instances of more than one objective, rather than having to decide a priori the relative importance of each objective in a composite single objective, means that the optimum trade-off between objectives can be searched for. Pareto approaches mean that multiple design-estimates of the Pareto front can be made in a single run of the optimisation, whereas single-objective approaches produce only a single best estimate of the optimum trade-off between objectives combined in the aggregate objective function from each optimisation run. A single-objective
algorithm, when used on a composite objective, is also unable to explore non-convex portions of the Pareto front between two or more design objectives.

Multi-objective approaches have most often been applied to explore the trade-off between two objectives, and this is the case for all the multi-objective optimisations performed in this thesis, but applications with three objectives are also possible. Beyond three objectives it becomes difficult to both visualise the resulting trade-off between objectives and to perform the computations necessary to assign each design a Pareto-rank.

Wright et al. (2002) used a simple genetic algorithm to perform a multi-objective optimisation on the HVAC system size and configuration in three different building types, with various constraints on the specification of the HVAC system. The two objectives (both to be minimised) were daily energy cost and occupant thermal discomfort. The model on which the optimisation was based was a simulation for a summer design day, a winter design day and an inter-seasonal design day. Although the number of variables of the HVAC system that were under investigation was relatively small (8 parameters needed to be specified), the control parameters amongst these were set on an hourly basis, resulting in a total of 200 control variables. The building type was not investigated as part of the optimisation, instead the optimisation was performed separately for each building type.

Wang et al. (2005a,b) optimised a building for lifecycle impact and lifecycle cost. The variables included in the optimisation were building orientation, aspect ratio (depth of the building divided by width of the building), window to wall ratio for each facade, wall type, wall insulation and roof insulation (nine variables in total). In order that the algorithm could analyse enough individuals to effectively explore the design space, the dynamic simulations were run for only two days per month on each building, so as to reduce the time required to analyse each individual. The building optimisation was used as a test case to develop an algorithm that could handle the particular characteristics of building-design optimisation. These were
identified as 1. A mixture of discrete and continuous variables, 2. The ability to couple with building simulation tools and 3. The ability to handle variables whose feasibility depends on the choice made for other variables (hierarchical variables).

In order to effectively handle hierarchical variables, the algorithm was adapted to allow both dominant and recessive gene types to be represented on the chromosome.

Hamdy et al. (2011) performed a multi-objective optimisation to minimise the in-use carbon dioxide equivalent (CO$_2$-eq) emissions and the investment cost of a two-storey semi-detached house in Finland. Eight design variables were included in the optimisation. Three of these were classified as continuous (external wall insulation thickness, roof insulation thickness, floor insulation thickness) and five as discrete (window type, heat recovery ventilation type, shading type, building air-tightness and heating/cooling system type). An earlier paper by the authors (Hamdy et al., 2009) examined the effectiveness of combining deterministic and genetic-algorithm optimisation approaches with the goal of minimising the number of simulation runs required. They examined the efficacy of two different approaches. The first added a fmincon (a gradient-based, deterministic search algorithm used to search for superior designs in a small area of the design space) search of the design space in order to refine the initial population of the genetic algorithm. The second combined a sequential quadratic programming method at the end of the genetic algorithm run. They found that both methods were able to improve either the speed of the optimisation or the quality of the results obtained by the algorithms, compared to a genetic algorithm alone.

Verbeeck and Hens (2007) went one step further and included three objectives in their optimisation on the design of a residential building in Belgium, with the objectives being lifecycle energy use, lifecycle ecological impacts and lifecycle costs. They used a genetic algorithm coupled with the dynamic simulation model TRNSYS. They split the optimisation of their building design into two different phases, one for the building envelope and one for the subsequent HVAC system. In the first
stage of the optimisation the design variables were insulation thickness, insulation type, window frame, window area, shading and air tightness (a total of six variables). The optimum design found in this first step was then taken and the HVAC system was optimised for that design. In the second step the design variables were heating system, heating delivery, temperature control type, exit temperature of water and ventilation type (five variables).

Evins et al. (2012b) applied a design-of-experiments approach in order to identify the most influential and complex variables (14 in total) before optimising those variables using a genetic algorithm to simultaneously minimise carbon emissions, capital and running costs, with constraints on overheating and roof area.

Sequential search optimisations have also achieved good results in building-design problems. For example Griego et al. (2012) used a sequential search optimisation method to optimise energy efficiency and thermal comfort measures for residential buildings in Salamanca, Mexico. Given that this approach required the analysis of over 18,000 individual building designs it can be assumed that the approach is only really applicable in design problems with short evaluation times. Favre and Peuportier (2013) used dynamic programming (a sequential optimisation method) to optimise the ventilation strategy and solar protection of a highly insulated building to minimise summer overheating. Bichiou and Krarti (2011) compared the robustness of three different optimisation approaches; genetic algorithm, sequential search and particle swarm. They found that genetic-algorithm and particle-swarm algorithms typically require less computational time to obtain optimal solutions.

Carlucci and Pagliano (2013) used a multi-objective particle-swarm optimisation to search for the optimum trade-off between summer and winter discomfort hours in a free-floating (unconditioned) building, in order to find designs that required very little heating or cooling. Design variables were roof construction, wall construction, glazing type, shading on each aspect and percent window opening.
A study by Attia et al. (2013), which interviewed leading practitioners and experts in the field of building-design optimisation identified the most commonly optimised variables as controls, systems and building envelope. Energy, cost and thermal comfort were the most commonly used objectives, and, in cases where constraints were applied, thermal comfort was overwhelmingly the most frequently chosen.

4.4 Comparisons with traditional methods

Many studies have shown that optimisation methods are able to show significant improvements over base-case or typical designs (see for example Coley and Schukat (2002), Wetter and Wright (2003) and Ihm and Krarti (2012)) and traditional techniques are often criticised from a theoretical standpoint. For example Wetter (2000) described the traditional manual optimisation procedure as:

\[\text{extremely time consuming, often impractical for more than two or three independent variables and only a limited improvement can be achieved.}\]

While Wang et al. (2005a) noted that:

\[\text{Since the iterative process is ineffective and time consuming, only a few trials are usually made by designers with many promising solutions left unexplored.}\]

However, perhaps owing to the complexity of designing robust experiments to compare human designers with computational optimisation methods, there have been few studies comparing the designs from optimisation methods with those found by designers using more traditional techniques on the same design problem.

Naboni et al. (2013) made a comparison of conventional, parametric and evolutionary optimisation approaches for finding the optimal trade-off between yearly heating
energy and yearly cooling energy with the ultimate objective of a nearly-zero-energy building. In the traditional approach the designer simulated ten different designs, with the choice of designs to test being guided by the designers’ experience, intuition and expertise. For the same design problem both an exhaustive search of all possible designs and an optimisation using a genetic algorithm were performed. This showed that the designs found by the traditional method were all far from the true Pareto front, in fact in many cases these designs were amongst the worst Pareto-ranked designs in the entire design space. In contrast the genetic-algorithm approach was able to make very good estimates of the Pareto front within 60 generations (with a population size of ten). The improvement from here to termination (135 generations) was merely one of adding more designs to the estimated Pareto front. While it may not be fair to say that the traditional approach is categorically worse than the genetic algorithm (perhaps the designer would have done as well as the algorithm with the same simulation budget, rather than only doing ten simulations), it is perhaps a reflection of the real-world situation; even if traditional design techniques are capable of making good estimates of the Pareto front given the same number of simulations, designers are unlikely to ever do this many evaluations. Optimisation methods provide not only an efficient way of searching for optimum designs given a limited simulation budget, but a systematic and automated method for deciding which designs to evaluate, and how to handle the results. Without this, even if, in theory, traditional, iterative methods for optimising the design were as efficient as optimisation algorithms, they are unlikely to be employed to the same extent because choosing, running and collecting the results from the large number of simulations required is prohibitively time consuming and intellectually taxing. A subsidiary question of this thesis, addressed in Chapter 12, is the degree to which optimisation methods in general offer an improvement over traditional design methods.
4.5 Barriers to uptake of optimisation methods by practitioners

Although building-design optimisation has a relatively rich history of academic study, it is still rare for these techniques to be used in real-world building-design situations. Instead, designers tend to rely on iterative methods and previous experience to guide the design process (Watson et al. (2013) and Naboni et al. (2013)).

Currently, the use of building-simulation tools is much more common than the use of building-optimisation tools, and this can be seen by examining the total number of tools available to – as of 2011 a total of over 400 building-performance-simulation tools existed compared to just 18 building-optimisation tools (Attia et al., 2013). Besides simulation programs having been in existence for longer, several important barriers to the more widespread adoption of building optimisation tools account for this disparity. Amongst the most important reasons cited is the time required to complete optimisation studies, or the computational power required to do so (Watson et al. (2013), Attia et al. (2013) and Zhang et al. (2013b)). The challenge of accelerating optimisation methods to make them more applicable to real-world studies has received much attention in recent research, and is the subject of the following Section.

4.6 Searching better, or searching faster

The time required to optimise building designs has been identified as an important barrier to the adoption of optimisation methods by industry. On a fundamental level, implicit in the desire to use optimisation techniques in a real-world situation is a desire to have those techniques make good estimates of optimal designs within a restricted time frame. If time were not a limiting factor then a designer could simply evaluate every possible design and be sure of having found the optimal design or
trade-off between objectives. This is rarely possible in real-world building-design problems because the simulation models tend to be time consuming to run (Naboni et al. (2013), Wang et al. (2005a) and Salminen et al. (2012)). In order to enable good estimates of optimal designs to be made in reasonable time frames, building design optimisation studies have employed one or a combination of several different tactics:

- simplifying the simulation model to allow faster run-times, meaning more evaluations can be made within the time available (see for example Coley and Schukat (2002), Hamdy et al. (2011) and Marin et al. (2013))
- Evaluating simulation models for a limited number of days, rather than a full year (see for example Wright et al. (2002) and Wang et al. (2005b))
- Reducing the number of design variables (and thus the total size of the design space) to enable good estimates to be made with fewer simulation evaluations (see for example Evins et al. (2012b), Wystrcil and Kalz (2013), Eisenhower et al. (2012) and Wang et al. (2013))
- Splitting a large design space into two smaller design spaces and doing one optimisation after the other on these two design spaces (see for example Verbeeck and Hens (2007))
- “Seeding” the initial population with designs previously established as performing well (Hamdy et al., 2011).
- Improving the performance of the search algorithm to allow good estimates to be made with fewer simulation evaluations (see for example Hamdy et al. (2011) and Lee and Hensen (2013))
- Maximising the spread of designs in the initial population, for example through the use of a latin-hypercube sampling method (Lee and Hensen, 2013).
All but the last two tactics in the above list make compromises on either the accuracy of the simulation model or the number of design variables that can be studied concurrently (if those variables are epistatic then not studying them concurrently risks missing the true optima). Thus any methodology that is able to improve the speed with which good estimates of optimum designs can be made, without compromising on the accuracy of the simulation model, will be potentially of great value in improving the applicability of optimisation methods to building-design problems.

One such methodology that has been proposed and implemented in several studies is that of surrogate-model optimisation. Existing applications of these methods are discussed in Chapter 5.

4.7 Recent developments in building-simulation optimisation

This chapter has so far explored what has typically been done in building-optimisation studies and how researchers have overcome the difficulties in applying optimisation techniques to building-design problems. The following section looks at some more recent, innovative applications of optimisation techniques and discusses how these might be used in the future.

Glassman and Reinhart (2013) demonstrated that the optimal building facade for a given location changes with changes in the climate. One particularly important application of optimisation methodologies may be to design buildings the low-carbon status of which is robust with regard to future climate change. This approach of searching for designs that are optimal according to uncertain, future conditions could be tied in with the robust-optimisation approach similar to that developed by Hopfe et al. (2012) (see Section 5.1).
Trubiano et al. (2013) used a genetic algorithm to optimise the building form for a single objective that was a composite of minimising heating and cooling load subject to an average lighting level constraint. Although previous studies have also used geometric variables such as these as design variables, the method used in this study made it possible to generate more complex building shape configurations. The authors identified ten geometric variables (angles, lengths and widths) that, when varied, enabled the creation of complex shapes. This optimisation methodology was intended to assist in the early stages of building design, when all that is being decided upon is the building form, a stage at which energy efficiency is frequently ignored (Li et al., 2013).

Marin et al. (2013) developed an interactive genetic algorithm that allowed the designer to interact with the evolutionary process as it was running, guiding it in specific directions. This approach is interesting because it implicitly allows the incorporation of many more objectives or constraints than might be practical within the framework of a standard evolutionary approach. The designer can have a diverse and complex set of objectives (for example visual appeal, spatial layout and functionality of the building) in mind when reviewing the elite designs suggested by each generation of the evolutionary process.

Bucking et al. (2013) used a genetic algorithm to evaluate the effect of different incentive regimes on the energy efficiency versus total cost (costs of purchasing energy and building improvements) Pareto front for building designs. As governments look to reduce building sector CO₂ emissions by developing building-energy targets that are beyond cost-optimal levels, this sort of approach could be used to inform policy with regard to incentive schemes.

Taheri et al. (2013) used a particle-swarm optimisation to calibrate the performance of a building-energy model on a real building. Design variables in this case were solar transmittance, insulation thermal conductivity, concrete density, infiltration rate, ventilation rate and set-point air temperatures. These variables were altered
by the optimisation method with the objective of minimising the error between the predicted performance and the real performance.

4.8 Summary

Throughout the literature on building-design optimisation there is a repeated implicit or explicit desire for optimisation to require as few main-model evaluations (and therefore dynamic building simulations) as possible. This chapter has explored implementations of methods that search the main models of objective functions (CO₂, energy use, cost etc.) directly. Surrogate models have been suggested as a promising method by which the number of main-model evaluations might be reduced (Wetter and Wright, 2004). The following chapter explores surrogate methods, and their application to low-carbon building design, in more detail.
Chapter 5

Surrogate modelling optimisation

This chapter introduces surrogate models, reviews their application in low-carbon building design optimisation and evaluates existing comparisons with non-surrogate optimisation approaches.

Surrogate modelling methods replace the time-consuming main model with a quick-to-interrogate surrogate model for some or all of the design evaluations. Surrogate modelling can be useful in reducing the total number of main-model evaluations required to make good estimates of optimum designs in design problems with time-consuming objective functions (such as those requiring dynamic building simulations), compared to search algorithms operating directly on the main model (stand-alone optimisation algorithms).

Various different surrogate modelling techniques exist, these include:

- Support Vector Regression (SVR) models
- Fixed-base (cubic, thin plate) models
- Polynomial models
- Gaussian basis models including Kriging
• Neural networks including Markov models

Each method has its own particular advantages and disadvantages. Broadly speaking, SVR, fixed-base and polynomial models have the advantage that they can handle both large numbers of design variables and large numbers of main-model sample points (Forrester et al., 2008). Kriging offers better performance at modelling complex landscapes, and more sophisticated options for searching the model besides a simple prediction (for example, the model can predict the expected improvement, see Section 6.1.4.1), but with the drawback of being limited in both the number of variables that can be handled and the total number of main-model evaluations that can be modelled (Forrester et al., 2008).

5.1 Surrogate models in low-carbon building design optimisation

Surrogate models are another way in which researchers have tried to improve the performance of their optimisations when the number of main-model evaluations is limited by restrictions in time or computing budget. A surrogate modelling approach typically involves first performing a sample of the main model(s) from which a fast-to-interrogate mathematical surrogate can be constructed. The surrogate is then searched for optimal designs using an optimisation algorithm. Periodically the designs suggested by the surrogate optimisation may be tested on the main model, and the surrogate re-calibrated to include the new design (this is referred to as on-line learning, or update points).

Hopfe et al. (2012) used the time advantage afforded by replacing the main model with a Kriging surrogate to allow a thorough sensitivity analysis, and from this to enable optimisation incorporating uncertainty. This involved setting confidence intervals for the performance parameters of key variables and finding the designs that
perform best under a worst-case scenario, rather than those designs that perform best under an as-designed scenario, as is typically the case. This so-called robust optimisation could also be applied to other promising areas of research in building-performance optimisation, for example in designing buildings with performance that is robust to future changes in the climate, or variability in user behaviour.

Kang et al. (2013) used a Gaussian surrogate model (the standard Gaussian-process model, rather than a Kriging model) to enable real-time decisions to be made regarding the optimal operation of cooling-system variables for a multi-storey office building. Because the building was very large and complex, a single EnergyPlus run took nearly two hours to simulate one month of operation. With simulation times so long, trying different what-if scenarios on the cooling system parameters in real time was unfeasible. Instead a Gaussian surrogate model of the main EnergyPlus model was built from a training dataset of sample designs. This surrogate model was then used to run a multi-objective optimisation to simultaneously minimise the mean and standard deviation of electricity consumption. Once the initial surrogate model was built, there was no on-line learning.

Kim et al. (2013) developed and tested a surrogate modelling optimisation approach using a Gaussian-process regression model and Bayesian approach (this was the same model type as used in the Kang et al. (2013) study, and was not a Kriging model). This model was used as a surrogate of an EnergyPlus model and then employed to optimise glazing parameters (U factor and solar heat gain coefficient) on the four cardinal aspects. The training dataset was relatively large (500 samples) and there was no on-line learning used.

Zhang et al. (2013a) used a Kriging surrogate model of an EnergyPlus building model that was then searched using a sequential optimisation method with the objective of minimising energy-use intensity. At the end of each run of the sequential optimisation, a suggested point was tested on the main (EnergyPlus) model and this sample point used to update the Kriging surrogate. The chosen design problem
deliberately contained a combination of discrete and continuous variables. The authors concluded that the algorithm had converged when the maximum expected improvement dropped below 0.05, and this happened after a total of 195 iterations (main-model evaluations), of which 100 were in the initial training dataset, and the remaining 95 were sample points suggested at the end of each optimisation run. They concluded that this was an improvement on the 300-500 iterations required by Wetter and Wright (2004) for a genetic algorithm search without a surrogate model.

Previous work by the author of this thesis (Sobester et al., 2012) used a Kriging surrogate model to optimise the extent of glazing, glazing U value, insulation thickness and thermal mass thickness for a simple domestic building. The objectives were minimisation of construction cost and in-use CO$_2$ emissions. Optimisations were performed both with and without constraint criteria.

Different authors have chosen different surrogate modelling techniques, although Gaussian-based models (including Kriging) are popular choices. The remainder of this thesis focusses on the potential of Kriging surrogate methods to accelerate the optimisation of low-carbon building designs. Kriging is chosen over other surrogate modelling methods for several reasons:

- It is a method that has already been used in an attempt to accelerate building-design optimisations, although its capability to reliably find superior designs in a restricted time has not been established.

- It is better at representing complex landscapes than SVR, fixed-base, polynomial models and more simple Gaussian-basis functions.

- In addition to providing an estimate for the performance of an un-tested design, it can also output an estimated error for that prediction. This enables the Kriging model to be searched for the expected improvement, which effectively balances exploration and exploitation of the design space (Forrester et al., 2008).
5.2 Existing comparisons of building-design optimisations with and without surrogate models

Surrogate modelling optimisation has been suggested as a promising method for reducing the number of main-model evaluations required to make good estimates of optimal building designs (Wetter and Wright, 2004). However, it should certainly not be assumed that the methodology will offer an improvement over standard evolutionary algorithm approaches, or that it offers the same level of advantage in all situations.

That surrogate modelling offers an improvement over conventional optimisation methods is assumed in most of the instances of its use in building-design optimisation; because the surrogate is much faster to interrogate than the main model it is assumed that its use enables a reduction in main-model calls to achieve the same quality of estimations of the optimal designs. This is the case for those studies by Kang et al. (2013) and Kim et al. (2013).

Where comparisons have been made regarding the number of main-model evaluations required to make good estimates of optimal designs with and without surrogate models, they have often been based simply on the reduction achieved in main-model evaluations. Hopfe et al. (2012) state that the use of a Kriging model reduced the number of main-model evaluations to between 5 and 20% of the amount that would have been required without the surrogate model, but the robust-optimisation performed in this study was not duplicated without the use of the Kriging model, so it is not possible to draw conclusions about whether or not this reduction in main-model evaluations also affected the quality of the optimisation results. It is perhaps unfair to single out this study for not having done a comparison of the performance both with and without a surrogate model. It may well be that the main-model evaluations were so time consuming that the large number of additional runs required by
robust optimisation were only possible with a surrogate model given the computing resources available.

Zhang et al. (2013a) also drew comparisons between the number of main-model evaluations required to make estimates of the set of optimal designs, concluding that the use of a Gaussian model had allowed convergence within 195 iterations (runs of EnergyPlus), which compared favourably to the 300-500 iterations required for a similar optimisation without a surrogate model in a previous study by Wetter and Wright (2004). However, several limitations of the study mean that this conclusion is difficult to sustain. First of all the performance of all possible designs was not known, so although the optimisation converged according to their criteria (based on the expected improvement), it may have converged at a local optimum. Perhaps more importantly, the two optimisation methods (the surrogate method of Zhang et al. (2013a) and the genetic algorithm with no surrogate model used in Wetter and Wright (2004)) were searching two different design problems. Different authors have reported a wide range of iterations needed in order to make good estimates, for example Verbeeck and Hens (2007) used a total of 6000 iterations (a population of 100 run for 60 generations), Hamdy et al. (2011) required several thousand whereas Hamdy et al. (2012) showed that through improvements to the genetic algorithm good estimates could be made in as few as 180 main-model evaluations. It may be that the design problem chosen was relatively simple and a standard method might have achieved similar results as quickly or quicker than using a surrogate model. Finally, only one run of the optimisation was made. With any stochastic optimisation method a degree of chance is involved in the quality of the results obtained, and in order to draw more solid conclusions, repeated trials on the same design problem are necessary.
Part III

Optimisation using Kriging surrogate models
Chapter 6

Overview of the Kriging methodology

This chapter describes the Kriging methodology implemented in this thesis then discusses its drawbacks and potential barriers to the further use of Kriging models in low-carbon building design optimisation.

Kriging is a geostatistical method that was originally developed for predicting mineral-ore density at un-sampled locations based on the values obtained at sampled locations in the same area (Krige, 1951). Since Kriging is able to predict the value of an objective (in the original case mineral-ore density) at unknown sites in a 3-dimensional space (the ground in the area of interest) its extension to being able to predict the value of an objective in a multi-dimensional space means that it can be applied to engineering design problems with multiple variables. The application of Kriging to engineering design problems developed after the work of Sacks et al. (1989) and is now one of the more popular surrogate modelling approaches to use in cases of expensive main-model evaluations (Jin, 2003).

A Kriging model describes the correlation between \( n \) sample points under the influence of \( k \) design variables such that the value of the objective function can be predicted at unsampled locations in the design space (locations in which at least one of the variables in the \( k \) set is different from a previously sampled combination of variables).
Although, in the case of building-energy simulations, the data is deterministic (a building simulation with the exact same inputs, using the same simulation engine, will always produce the same result), the Kriging model instead views this data as stochastic (the result is subject to probability). Instead of the objective function having one value at any given location in the design space, it is viewed as a probability distribution with an estimated mean and error. For locations that have already been sampled on the main model the mean is the value obtained from the main model and the error is zero.

This viewing of the model as stochastic when it is in fact deterministic is a necessary fallacy that allows the building of a model that can output not only a prediction of the most likely objective-function value for an unsampled location in the design space, but also the estimated error in the model at that location. In any optimisation algorithm there is a balance to be struck between exploration (searching in a manner that gives a good idea of the shape of the design landscape across the whole design space) and exploitation (searching more intensively in a smaller region of the design space with the objective of finding local optima) of the model, and much work has been devoted to balancing these two aims within the field of multi-objective genetic algorithms (see for example Horn et al. (1994), Tan et al. (2001) and Deb et al. (2002)). Because the Kriging model has an estimated mean and error at unsampled locations this means that the model can be interrogated for a metric that balances exploration and exploitation of the model, the expected improvement. This is discussed in more detail in Section 6.1.4.1.
6.1 Generalised methodology for Kriging optimisation

6.1.1 Sampling the main models

Before the Kriging model is built, the objective function it is going to be a model of must first be sampled. In the case of the low-carbon building design optimisations described in this thesis the functions are either a CO\textsubscript{2} emissions model on its own (for single-objective optimisations) or with a construction cost model. The in-use CO\textsubscript{2} emissions were calculated from demand estimates made by a building-energy simulation model. The building-simulation program used throughout this thesis is EnergyPlus\textsuperscript{1}, a BESTTESTed dynamic simulation tool developed by the US department of Energy for the estimation of building energy demands, but the method could equally be applied to any building-simulation program.

When deciding on the total size of the initial sample, a balance must be struck between exploration and exploitation of the design space. Sőbester et al. (2005) advise allotting approximately one third of the simulation budget to the initial sample and the remaining two thirds as update points while searching the design space for optimal designs. However, it should be noted that this is by no-means a hard-and-fast rule and other studies have found that smaller sample sizes than this can also produce reliable results.

In order to make best use of the sampling budget the sampling plan should cover the design space as evenly as possible. This is far from trivial in the types of multi-dimensional design spaces that are typically encountered in building-design optimisation problems. The method employed in this thesis chooses sample points using a genetic algorithm to search for the Morris-Mitchell latin hypercube with the best space-filling characteristics (Forrester et al., 2008). This is achieved by

\textsuperscript{1}http://www.energyplus.gov
seeking to maximise the single objective of mean Euclidian distance between sample points in the multi-dimensional design space, with the design variables as the vectors being selected by the genetic algorithm. Once this latin hypercube has been found the sample points it suggests are run on the main models. The results from these samples are used to build the Kriging model.

6.1.2 Constructing the Kriging model

Kriging is a specific form of Gaussian radial basis function in which the width of the basis function, $\theta$ (which describes the strength of influence a particular variable has on the correlation between designs), is able to vary for each design variable as can the exponent, $p$ (which describes the smoothness of the correlations between sample points). This additional flexibility compared to a standard Gaussian radial basis function brings with it an improved ability to model complex design spaces, but an increased computational cost (Forrester et al., 2008). The basis function which describes how one sample point is correlated to another takes the form

$$\psi(i) = e^{\exp \left( -\sum_{j=1}^{k} \theta_j |x_j^{(i)} - x_j|^{p_j} \right)}$$

(6.1)

The correlation between two sample points under the influence of one variable depends on the values assigned to $\theta$ and $p$. This is illustrated in Figure 6.1.

---

2 Although in many implementations of Kriging, including the one presented here, the exponent is set at 2 to simplify the computation process.

3 In a standard Gaussian radial basis function the width ($\theta$) is the same for all variables, set at $1/\sigma^2$ and the exponent is fixed at 2.
Figure 6.1 – The values chosen for $\theta$ and $p$ affect the Kriging model of correlation between sample points. For a given value of $p$, increasing the value of $\theta$ causes the correlation to fall more quickly as the distance between sample points increases. For a given value of $\theta$, increasing the value of $p$ increases the smoothness of the correlation. The distance between two points is given by the formula $|x^{(i)}_j - x_j|$ and the correlation given by equation 6.1 with $k$ set to 1 (only one variable).

For a Kriging model that is a good fit to the sample data, the values of $\theta$ for each design variable will depend on the influence of that variable. If a variable has very little influence over the objective function then $\theta$ will be close to zero, meaning that even sample points which are far apart with respect to that variable are highly correlated – altering the value for that variable, even by a large amount, doesn’t affect the objective function very much\(^4\). On the other hand, if a variable is very important then the value of $\theta$ will be high, meaning that even a small change in that variable is expected to produce a large change in the objective function. In the method used in this thesis, the value of the exponent $p$ was set at 2 to simplify computations.

Building a Kriging model involves building a collection of the basis functions shown in equation 6.1 to describe the correlation between each sample point and every other

\(^4\)If possible, unimportant variables should be excluded from the optimisation problem altogether – if they are not influential then their inclusion needlessly adds another dimension to the design space, making the search for good designs more difficult.
sample point in the design space, with values of \( \theta \) chosen to describe the influence of each variable such that the correlations best fit the data. This collection of basis functions forms a \( n \times n \) correlation matrix, denoted \( \Psi \), from which a covariance matrix can be calculated as follows

\[
\text{Cov}(Y, Y) = \sigma^2 \Psi
\]  

(6.2)

This covariance matrix is the starting component of the Kriging model. It is now necessary to tune the Kriging model to ensure that it is a good fit to the observed data before using it to make recommendations about regions of the design space that are likely to be worth further exploration.

### 6.1.3 Optimising \( \theta \) to maximise the likelihood of the data

As described in Section 6.1.2, the value chosen for \( \theta \) and \( p \) for each variable describes the influence that variable has over the Kriging model. In the implementation of Kriging optimisation used in this thesis the value of \( p \) is fixed, but the values for \( \theta \) are chosen for each variable in order to construct a Kriging model that fits the data as well as possible. Once the model fits the observed data well it can be used to make predictions about designs that have not yet been sampled on the main model. In order to assign the values of \( \theta \) for each variable a search is made for the vector of \( \theta \)s that are most likely to have come from a model with the particular set of sample points. Specifically, the search aims to **maximise the likelihood of the parameters given the data**. The calculation of this likelihood cannot be differentiated so instead the vector of \( \theta \)s is chosen by a single-objective genetic algorithm treating each value of \( \theta \) as a variable and seeking to maximise the likelihood of that vector of \( \theta \)s given the sample data. Each string of \( \theta \) values (one for each variable in the building design) forms an individual in the population and the first population is chosen at random. The performance of each individual is assessed against the
objective function, which in this case is the likelihood, and is to be maximised. The population of individuals evolves over the generations by means of mutation, crossover, replication and selection (described in Figure 2.1) towards individuals with a high likelihood and thus values of $\theta$ that best fit the sample data.

6.1.4 Optimising design variables to find good locations for update points

It is possible to simply search the Kriging model for its global optimum (minimum or maximum objective function, depending on whether the optimisation problem is one of maximisation or minimisation) after the values of $\theta$ have been chosen to fine-tune the model. Indeed, this is the approach taken by some studies (see for example Kang et al. 2013 and Kim et al. (2013)) and may be appropriate when decisions need to be made very rapidly and in cases where the surrogate is a good approximation of the main-model. However, since the Kriging model is itself a model of a model, it is likely that it contains some inaccuracies, and the global optimum found through such a search may not accurately reflect the global optimum in the main model. Instead, the Kriging model is used as a guide to choose where next to search the main model – an update point. An evaluation is then made on the main model using this design, and the value fed back into the Kriging model which is recalibrated using the method described in Section 6.1.3 and then searched again for promising update points. The method for choosing update points is described below.

6.1.4.1 Criteria for choosing update points

The Kriging model can be interrogated for a prediction of the objective function at an unsampled location (a combination of design variables that has not already been tested). This is done in a similar way to how the parameter $\theta$ is tuned to maximise
the likelihood of the sample data described in Section 6.1.3. Instead of searching for the vector of $\theta$s that maximises the likelihood of the parameters given the data, as was done when trying to fit the model to the sample data, the search is now for the value that maximises the likelihood of the parameters given the previously sampled data and the new prediction. Unlike the search for the vector of $\theta$s that maximise the data, this can be achieved through differentiation so requires only a single calculation.

In addition to the capability of making a prediction of performance at an unsampled location, Gaussian basis models (of which Kriging is one) are also able to provide an estimate of the error of the model at unsampled locations. This offers two other potential metrics that can be used to choose update points, the probability of improvement and the expected improvement. The methodology used in this thesis uses the expected improvement (EI) as the guide for promising designs to test on the main model(s), and a genetic algorithm is employed to search for those designs with the greatest expected improvement. The genetic algorithm uses the EI (or the multi-EI) as the objective function, and each individual in the population is represented by a chromosome formed from a string of design variables. The initial population in each search of the Kriging model is chosen randomly. The mathematical derivations and explanations for the methods described in this chapter are explained in more detail in Forrester et al. (2008). The use of expected improvement balances exploitation and exploration of the Kriging model and means that it is not necessary to use a complicated genetic algorithm in order to achieve this balance. For multi-objective optimisation problems, the expected improvement can be relatively easily adapted to provide an estimate of the probability that a design will improve on the current estimate of the Pareto front using a separate Kriging model for each objective. The multi-EI for a given design is a single value, enabling the use of single-objective optimisation algorithms on multi-objective optimisation problems, while still using
a Pareto-approach and without the usual problems entailed in combining multiple objectives into a single objective described in Section 4.3.

### 6.1.5 Putting it all together

An overview of the Kriging optimisation process has been presented in the preceding sections. This process is summarised in Figure 6.2.

![Figure 6.2 – A workflow showing the different stages of the Kriging optimisation process as implemented in this thesis](image)

1. Sample main model using latin hypercube
2. Construct $n \times n$ correlation and covariance matrix
3. Using a genetic algorithm, search for the vector of $\Theta$ values which maximises the likelihood of the data
4. Using a genetic algorithm, search for the design that maximises the expected improvement
5. Test the suggested design on the main model
6. Stopping criteria met?
   - Yes
   - No
5a. Add update point to $n$ sample points
7. End optimisation
6.1.6 A note on the complexity of calculations involved in the Kriging optimisation process

The calculation of *likelihood* (the measure of how well the Kriging model fits the observed data, used to search for the optimal vector of $\theta$ values) and *expected improvement* (the measure used to assess different designs, used to search for optimal update points) are repeated many hundreds of times in each loop of the optimisation process (see Figure 6.2), guided by the genetic algorithm, as part of the optimisation process. Because so many calculations are made the speed of these calculations is important. $\Psi$ is used in the calculation of both likelihood and expected improvement and so its size has an important effect on the speed and complexity of these calculations. Since $\Psi$ is a $n \times n$ matrix, where $n$ is the number of sample points included in the Kriging model, an increase in the number of sample designs included in the Kriging model leads to an exponential increase in the size of $\Psi$, and thus the complexity and time taken to perform both the likelihood and expected improvement calculations. In fact the complexity increases even more than this for the calculation of the likelihood; the matrix inversions involved mean that the number of calculations required scales according to $n^3$, rather than $n^2$. Because of this exponentially increasing complexity, in their book on surrogate modelling optimisation Forrester et al. (2008) recommend a maximum of 500 sample points. This recommendation is based on the computing power available at the time the book was written, and may rise with increases in computing power, or with parallel implementations of the calculations involved in Kriging (Forrester, 2013). The effect of the number of sample points on the time taken to run the two principal genetic algorithms (one to search for the optimal vector of $\theta$ values and one to search for the optimal combination of design variables) is shown in Figure 6.3.
Figure 6.3 – The time taken for each loop increases as the number of sample-points included in the Kriging model increases. In the case of the GA searching for the optimum vector of $\theta$ values this increase is exponential, whereas for the GA searching the Kriging model for optimum building designs the increase is linear.

As can be seen in Figure 6.3, the time taken per loop (per sample point added to the Kriging model) to search for the optimum model parameters (values of $\theta$) increases exponentially, while the time taken to search the resulting Kriging model for optimal designs increases linearly. However, since the times shown in in Figure 6.3 are per loop, and since each additional loop adds to the total time taken for all previous loops, the total time taken by the genetic algorithm searching the Kriging model for optimum designs increases exponentially, rather than linearly, as more samples are added, and the total time taken by the genetic algorithm searching for optimum values of $\theta$ increases exponentially at a greater rate than shown in Figure 6.3. The total time taken for the example shown in Figure 6.3 is 26 hours, 23.5 of which were for the genetic algorithm searching for optimal values of $\theta$, and 2.5 of which were for the genetic algorithm searching the Kriging model for optimal building designs. These figures are longer than the Kriging time overheads quoted elsewhere in this thesis for two reasons. The first (and most important) is that this example is for a 500-sample Kriging model, whereas most of the Kriging optimisations described
in this thesis used fewer sample points than this. The second is that optimisations tend to do a much larger initial sample than the one shown in Figure 6.3, which uses an initial sample size of ten, to illustrate how the time-per-loop increases. A larger initial sample means fewer time-consuming loops in total.

The time overhead associated with Kriging means that even if it can reduce the number of main-model evaluations required to make good estimates of optimal designs, it is not always worth using. Whether or not it is worth using depends on the size of the time overhead, the reduction in main-model evaluations enabled by the use of Kriging and how time-consuming the evaluation of the main models is.

Increasing the number of variables included in the optimisation does not increase the complexity of the calculations in an exponential fashion, but in a linear one (each additional variable included means that formula 6.1 must be run an additional time for each value in \( \Psi \)). However, more variables still poses a problem to Kriging since each additional variable makes constructing a Kriging model with a good degree of fidelity increasingly challenging, and because the more dimensions there are in the design space the less likely it is that 500 sample points can adequately search that space.

6.2 Barriers to the implementation of Kriging optimisation methods in building-design

6.2.1 Discrete variables and discontinuities in building-design problems

The Kriging methodology assumes that the variables are continuous, and that the relationship between the variables and the performance is also smooth (Forrester et al., 2008). In building-design problems this is frequently not the case; many
design problems are a mixture of continuous and discrete variables, and there may be many discontinuities in the relationships between variables and performance (Wetter and Wright (2004), Wang et al. (2005b) and Hamdy et al. (2011)). This mismatch between the Kriging model and the main model may introduce inaccuracies in the predictions made by the Kriging model. If these inaccuracies are large enough then the advantage offered by Kriging optimisation, in terms of allowing a reduction in the number of main-model evaluations or total time required for the optimisation, may not apply in all situations.

6.2.2 Other limitations of Kriging optimisation

In addition to the concern that there is a mismatch between the discontinuous nature of typical building-design problems and the smooth nature of Kriging models, there are other drawbacks to Kriging optimisation methods that apply more widely, rather than just in optimisation of design problems with discrete variables.

- The Kriging modelling method is limited in the number of variables it can take into consideration and still make good predictions about the main model, since each additional variable makes the model more difficult to fit. This limitation may restrict the use of Kriging methods on building design optimisation problems, as compared to standard genetic algorithms, which can handle very large numbers of variables\(^5\).

- The total number of main-model evaluations that can be run using a Kriging approach is limited since with each additional sample point the fitting and interrogating of the Kriging model becomes exponentially more time consuming.

---

\(^5\)Although it should always be borne in mind that the addition of each variable makes the design space exponentially larger and thus more difficult to search for good designs. Striking a balance between including enough variables to make good estimates of the global optimum and not including so many that the search becomes too challenging for the algorithm will always be important.
The use of a surrogate-modelling approach makes taking advantage of parallel-computing capability more complicated.

Each of these potential drawbacks is expanded upon in the following sections.

### 6.2.3 Limitations in the number of variables that can be considered

In order to fit a Kriging model to the sample points, the value of $\theta$ must be tuned for each variable. As described in Section 6.1.6, as the number of variables increases it becomes increasingly difficult to fit a Kriging model to the data with a high enough degree of fidelity. The literature generally refers to 20 variables being the maximum number that can be handled by a Kriging surrogate model while still maintaining good predictive capabilities for single-objective optimisation problems (Forrester et al., 2008; Hopfe et al., 2012). This should be reduced further still for multi-objective design problems because of the extra complexity introduced by the additional objective (Forrester, 2009).

This may not be such a problem – although it may be possible to include a very large number of variables in an optimisation using genetic algorithms without a surrogate model, it may not be desirable since each additional variable makes the design space exponentially larger (see equation 1.1) and thus harder to search. For an optimisation in which the main model(s) is very expensive to interrogate, it may be unlikely that an optimisation involving more than 20 influential variables will get close to convergence within the simulation budget. Regardless of whether a Kriging model is used, this restriction in the number of variables included, or one similar to it, may well be sensible.
6.2.4 Limitations in the total number of simulations that can be run

Because the calculations that are run over and over again during the Kriging optimisation process all rely on the $n \times n$ matrix $\Psi$, they become more time consuming and complex the more sample points that are added to the Kriging model. This places an upper limit on the number of sample points that can be included in the Kriging model (and therefore the number of building-energy simulations that can be performed) of around 500. This is described in more detail in Section 6.1.6.

In a similar way to the limitation on the number of design variables described in Section 6.2.3, the limitation on the total number of building-energy simulations that can be run using this Kriging optimisation method may well exist whether or not a Kriging model is employed. For any design problem with simulations that are computationally expensive enough to make considering the use of a surrogate model worthwhile, the total number of simulations that can be performed within the computing budget will be limited in any case.

6.2.5 Limitations in exploiting parallel-computing resources

While the implementation of stand-alone genetic algorithms using parallel computing is relatively straightforward (each member of a given population can be simulated in parallel, computing resources permitting), the implementation of surrogate-modelling optimisation in parallel is less straightforward.

The genetic algorithms that fine-tune the Kriging model and then search it for promising designs could be run in parallel much as they can in the standard approach, but if the time-consuming portion of the optimisation is the evaluation of one or more of the main models (and it should be, or else the extra computing cost of using
a surrogate approach would not be worth paying) then it is these that should be run in parallel.

While parallelisation of the main-model samples used to build the initial Kriging model may be straightforward (simulate all individuals in parallel, or in batches of the maximum number that can be simulated at once), parallelisation of the designs tested on the main model during the on-line learning phase of the optimisation is much more complicated. Methods have been proposed for demanding multiple update points from the Kriging model on each optimisation run (Parr (2012) and Chevalier et al. (2013)), and these multiple updates could be run on the main models in parallel. However, this parallelisation can be expected to generally result in inferior performance, for the same number of sample points\(^6\), than when choosing the update points one at a time (Chevalier et al., 2013). This is because, if multiple update points are being selected, some of them are selected based on a Kriging model that has been built from fewer main-model samples than would have been the case if the updates were being chosen one at a time. For example, after a training-data set of 70, if the update points are chosen one at a time, and the Kriging model re-tuned in between each main-model sample, then the 75th main-model sample would be chosen based on a model containing 74 sample points. If, after the same size training-data set, the update points are chosen five at a time, then the 75th main-model sample would be chosen based on a model built around 70 sample points. The effect of this on the performance can be expected to be larger the more update points are chosen per loop, and there is likely to be a balance to be struck between making the most of opportunities for parallelisation and improving the fidelity of the Kriging model on which optimisations are performed.

The development of Kriging optimisation methods that can be run in parallel is very much in its infancy. Initial results are promising, with Chevalier et al. (2013)\(^6\)This does not mean that the parallelisation is a bad idea, just that its effect, in Kriging, is unlikely to be a simple matter of accelerating the optimisation by a factor equal to the number of main-model evaluations that can be run simultaneously.
finding that on two test-cases, selecting four update points at a time, rather than one, resulted in comparable performance, for the same number of total sample points (i.e. the general expectation of inferior performance through parallelisation explained above did not apply). However, the two test-problems were single-objective and only contained two and six variables, respectively, and the same may not apply on more complex optimisation problems. Furthermore, the more update points are selected at once the more likely a drop in performance, as explained above.

In contrast, the acceleration of stand-alone genetic algorithms using parallel computing is relatively straightforward and well established (by, for example, jEPlus+EA (Zhang and Korolija, 2010)).

6.2.6 Summary of challenges and rationale for further research

It is certainly the case that surrogate models can be constructed using Kriging that are much faster to interrogate than objective-function models involving dynamic building simulation, and this has made the method attractive as a way to reduce the computational burden of building-design optimisation. That these fast-to-interrogate surrogates result in an acceleration of the process of optimisation (arriving at a comparable set of designs more quickly, or a superior set of designs in the same amount of time) is implied but not tested in those studies discussed in Section 5.1. Simply because the surrogate is fast to interrogate it does not necessarily follow that optimisation using these surrogates in lieu of the main models will accelerate the optimisation process. There are several reasons to suspect that it may not. The most important of these is a concern about the fidelity of Kriging models, which assume a smooth and continuous design space, on building design problems with highly discrete variables and discontinuities in the objective functions. There are also important limitations in terms of the number of variables that can be con-
sidered using a Kriging approach, the total number of main-model interrogations that can be included in the model and the degree to which the methods can take advantage of parallel computing resources.

In light of both the potential benefits of Kriging methods, and the potential problems, the main aim of this thesis is to establish the extent to which Kriging-surrogate optimisation methods can accelerate the optimisation of low-carbon building design problems, compared to existing non-surrogate optimisation methods.

### 6.3 How should the comparison between Kriging optimisation and main-model optimisation be made?

The aim of using a surrogate model (in this case a Kriging model) is to enable good estimates of optimal designs to be made with fewer calls to the main model than would be required for an optimisation working without a surrogate model (a stand-alone optimisation method). In order to draw fair comparisons between the two methodologies, both methods should be compared on the same optimisation problem, using the number of main-model evaluations required to make equivalent-quality estimates of optimum designs as the metric for comparison. Sufficient runs of each experimental set-up should be performed to enable the average performance of each method to be estimated and the statistical significance of any differences in performance of the two methods to be determined.

However, the situation is more complicated than is implied by a simple comparison of the number of main-model evaluations required by different methodologies. There is, typically, an additional time-cost associated with the use of the Kriging model due to the time requirements discussed in Section 6.1.6. In order to be worth using, the Kriging model must reduce the number of main-model evaluations to such an
extent that more time is saved in this way than is added through the use of the surrogate model. This will depend, in part, on how time consuming the main model is to run. In instances of building-design optimisation with simple building-energy models, main-model evaluations may be so fast that it may never be a good idea to use a surrogate-modelling approach.

6.4 Summary

The theory behind using surrogate models to reduce the computing cost of optimisation is simple – by replacing a time-consuming model with one that is fast to interrogate, more interrogations can be made in a given computing budget and therefore the design space can be explored more thoroughly. One particularly promising surrogate modelling approach is that of Kriging. There are, however, reasons to suspect that the implementation of Kriging on building-design optimisation problems faces several obstacles and that it should not be assumed that these methods offer an advantage over the use of search algorithms without surrogate models. Several studies have implemented a surrogate optimisation approach for building design problems, and in some cases direct comparisons have been drawn between the performance of the optimisation with and without the surrogate model. These comparisons are not currently robust enough to conclude that Kriging, or other surrogate modelling optimisation approaches, offer an advantage over optimisation without surrogate models, or indeed in which situations the use of a surrogate model may or may not be appropriate.

The main aim of this thesis is to establish the typical performance improvement offered by using the Kriging surrogate-modelling optimisation technique over methods in which the optimisation runs on the main model directly (stand-alone optimisation methods). In addition to this main aim, this thesis will attempt to establish
some of the advantages and disadvantages of each method and to identify situations which should lead a designer to choose one method over the other.
Part IV

Comparisons of performance between Kriging optimisation and stand-alone genetic algorithms on low-carbon building design problems
Part I introduced the need for low-carbon buildings and the challenges facing designers of those buildings, before describing methods to assist in the search for optimal designs. It was noted that, while optimisation methods are effective at finding good designs in complex, multi-dimensional design spaces, their use outside of academic research remains limited, in part due to the time-consuming nature of the optimisation process. Surrogate-modelling methods have been suggested as a method for reducing the time required for optimisations to make good estimates of optimal designs. This Part describes experimental comparisons of one particular surrogate-modelling method, that of Kriging, against established stand-alone genetic algorithm methods, with the aim of establishing whether or not its use can accelerate low-carbon building design optimisations.

The methods were tested on a variety of single and multi-objective low-carbon building design problems, in some cases searching for known optima and in others for unknown optima. Design problems based around building models were chosen, instead of generic test-problems, because of a desire to ensure the particular characteristics of low-carbon building-design optimisation problems were represented.
Chapter 7

Methods for all experiments

This chapter describes methods that are common to all the experiments performed in this thesis, and includes a discussion of methods for comparing results obtained by different optimisation methods.

The experiments described throughout this thesis involve the comparison of different methods for searching for optimal low-carbon building designs. For the experiments described in Chapters 8, 9, 10 and 11, this comparison was between a method which used genetic algorithms to search Kriging surrogate models for promising designs to be tested on the main models, and methods which used genetic algorithms to search the main objective-function models directly, without the use of a surrogate model. The methodology for each of these two methods is illustrated, in general terms, in Figures 2.1 and 6.2. A description of the set-up of these experiments is provided in this chapter.

Throughout this thesis, low-carbon building design problems are used as test-cases for the optimisation methods in an attempt to accurately reflect the likely performance of these methods on such problems. However, the actual optimisation results – the resulting building designs, are not the principal focus. The important thing is the relative performance of the different methods at finding optimal designs. As
such, time is not spent describing the optimal designs, or exploring why some designs perform better than others.

7.1 Building models for EnergyPlus simulations

All building energy models were constructed in DesignBuilder\(^1\), a user-friendly front end to the BESTTESTed building-simulation program EnergyPlus. After the models had been built in DesignBuilder, they were exported as an *.idf file that could be further modified to allow variables to be altered during the optimisation. The simulations were then managed by the batch-simulation tool jEPlus (Zhang and Korolija, 2010). The modifications necessary to allow jEPlus to alter variables within the building models are explained on the jEPlus website\(^2\). The building models used in each optimisation are described in detail in Appendix A. EnergyPlus simulations provided the estimates of energy demand for the different building designs.

Both domestic and commercial building models were used as test-cases for the optimisation methods, and the models became more detailed, and time consuming, in each subsequent experiment. The models used in the first experiments were deliberately simplistic in order to reduce run-times and allow many iterations of different experimental set-ups, and each subsequent experiment used more complicated models due to a desire to test the optimisations on more realistic and challenging problems.

7.2 Calculating cost and CO\(_2\) objectives

The two objective functions used for all the experiments described in this thesis were construction cost and CO\(_2\) emissions, although different models are used in different experiments. The choice of these two objectives was based on several factors. Firstly,

---

\(^1\)DesignBuilder Software Ltd. Available online at http://www.designbuilder.co.uk

\(^2\)http://www.jeplus.org
these objectives are popular choices in existing environmental building design optimisation studies (see Section 4.3 and Attia et al. (2013)). Secondly, an energy or CO$_2$ emissions target is a main (often the most challenging) component of many voluntary and compulsory building standards. CO$_2$ was chosen as the principle objective in the single-objective tests and one of two objectives in the multi-objective tests. Construction cost was chosen as the other objective in the multi-objective tests since cost is an important factor in any building project. These two objective functions took the place of $f_1$ and $f_2$ in the multi-objective optimisation formula in Section 2.1.

The CO$_2$ emissions were calculated from the energy-use estimates provided by the EnergyPlus simulations (and, in the case of the experiments described in Chapters 9 and 12, models for the embodied CO$_2$ emissions of different construction materials). Details of the cost and CO$_2$ models used to calculate performance against objectives are provided in Appendix B.

### 7.3 Choice of simulations to perform in each method

Both optimisation methods used jEPlus to handle individual and batch simulations. The choice of which simulations to run was made by the optimisation methodologies. In the case of jEPlus+EA, each design in a population was simulated. In the case of the Kriging method, an initial batch of simulations was made (the training data-set) and subsequent individual simulations were made in order to establish the performance of update points suggested by the genetic algorithm searching the Kriging model.

In the case of the Kriging optimisation, the genetic algorithm and Kriging modelling functions were run in Matlab, and simulations called from Matlab via jEPlus (Figure 7.1). The results from the simulations were then read back into Matlab and used in the models for CO$_2$ and cost. This was also the method used in the single-objective
optimisation without a surrogate model described in Chapter 8. In the case of 
jEPlus+EA optimisations, the genetic algorithm, the EnergyPlus simulations and 
the CO$_2$ and cost models were all run from within jEPlus+EA.

7.4 Optimisation using Kriging surrogate models

The method used throughout this thesis is that described in Forrester et al. (2008) 
and summarised in Chapter 6. After the initial sample, this method tests just one 
update point at a time on the main model(s).

In addition to a genetic algorithm searching for promising update points, at the 
end of each GA run the gradient-based search method fmincon was used. This was 
intended to search the location around the design suggested by the GA in more 
detail, to try and improve upon that design.

The size of the initial sample used to build the first Kriging model, with sample 
points chosen using an optimal Morris-Mitchell latin hypercube, was not kept the 
same for all experiments, and is mentioned in the relevant methods section for each 
experiment.

7.4.1 Interaction between Matlab, jEPlus, cost and CO$_2$ models

The Kriging modelling and optimisation took place within Matlab, but it was ne-
cessary for this process to call EnergyPlus simulations, and to use the demand 
data from these simulations to generate estimates for CO$_2$ emissions using the CO$_2$ 
models. A similar process took place for the cost modelling, with Matlab calling 
the cost models to provide cost estimates for different designs. This interaction is 
summarised in Figure 7.1.
1. Choose size of initial sample \( n \) and total number of update points \( z \)
2. Input number of variables \( k \)
3. Use GA to search for latin hypercube of \( n \) designs that best cover the \( k \)-dimensional design space
4. Write \( n \) design vectors to text file to be tested on main models
5. Compile dataset of designs with corresponding cost and CO\(_2\) emissions
6. for \( z \) update points
7. Calculate Pareto front from current dataset
8. for each objective
9. Use GA to search for optimal \( \theta \) values for each design variable
10. Build \((n+z) \times (n+z)\) correlation and covariance matrix
11. end for
12. Use GA to search for optimal combination of \( k \) design variables
13. Write design vector to text file to be tested on main models
14. Add update point to dataset
15. end for

Figure 7.1 – An overview of the interaction between Matlab, jEPlus, cost and CO\(_2\) models used in the Kriging optimisation. The Kriging optimisation was run within Matlab and designs to be simulated on the main models were exported to jEPlus (for simulation on EnergyPlus), then to cost and CO\(_2\) models. In the experiments described in Chapters 8, 9 and 13, the cost and CO\(_2\) models were also built within Matlab. For the experiments described in Chapters 10 and 11, the cost and CO\(_2\) models were built into the EnergyPlus simulation and were thus output by jEPlus directly.

7.4.2 Algorithm settings used for optimisation using Kriging surrogate models

The settings for the genetic algorithm used both to search for the optimum values of \( \theta \) and to search the Kriging model for promising update points employed the default settings provided with the Kriging optimisation package\(^4\) accompanying the book Engineering Design via Surrogate Modelling (Forrester et al., 2008). These are shown in Table 7.1. The genetic algorithm used to search for the optimal Morris-Mitchell latin hypercube (in order to choose sample points) had a population size of 20 and ran for 10 generations.

\( ^3 \)http://www.jeplus.org  
\( ^4 \)Available at http://www.wiley.com/go/forrester
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value chosen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tournament size</td>
<td>5</td>
</tr>
<tr>
<td>Number of bits per variable</td>
<td>20</td>
</tr>
<tr>
<td>Probability of reproduction (copying)</td>
<td>0.1</td>
</tr>
<tr>
<td>Probability of crossover</td>
<td>0.5</td>
</tr>
<tr>
<td>Probability of mutation</td>
<td>0.4</td>
</tr>
<tr>
<td>Probability of either single or double-point crossover being chosen at crossover</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 7.1 – Settings for the genetic algorithm used on the Kriging model and for the single-objective stand-alone GA.

As mentioned in Section 6.1.2, the exponent $p$ in the Equation 6.1 was set at 2 for all Kriging models. $p$ controls the smoothness of the correlation between sample points. While it would have been possible to also optimise the values taken for $p$, and it may have resulted in more accurate Kriging models, it would also have added more computational cost to the Kriging process.

The genetic algorithm searching for the optimum values of $\theta$ had a population size of 100 and ran for 20 generations in between each sample point (see Figure 6.2) for the experiments in Chapter 8, and thereafter was changed to a population size of 40 running for 20 generations for all subsequent experiments (the reasoning behind this change is described in Section 7.4.4). For all of the Kriging optimisations, the genetic algorithm used to search the Kriging model for promising update points used a population size of 100 running for 20 generations between sample points. The same genetic algorithm was used without a surrogate model on a single-objective optimisation problem in Chapter 8, and in this case various different population sizes were tested. These are described in more detail in Chapter 8.

7.4.3 Coding of discrete variables in Kriging

Kriging, as covered in Section 6.2.1, assumes that each variable is continuous. Specifically, the values that each variable can take are scaled to one, and the genetic
algorithm is able to choose any number between zero and one. In the design problems used in this thesis many variables have only a handful of discrete options. In order that the values selected by optimisation on the Kriging model can be run on the main model(s), it was necessary to turn the continuous variables suggested by the Kriging model into discrete choices. Rules similar to the following (illustrated for a 3-choice variable) were used:

If $X_1 \leq 0.33$ choose option 1

or if $0.33 < X_1 \leq 0.66$ choose option 2

else if $X_1 > 0.66$ choose option 3

Throughout the experiments presented in this thesis, whenever the variables being represented in Kriging were discrete, coding rules similar to those shown above were used.

7.4.4 Reducing the time overhead associated with Kriging

Perhaps ironically for a method that aims to reduce the time taken to optimise a design problem, the testing of Kriging is much more time consuming than testing a standard genetic-algorithm-based optimisation method. With a stand-alone GA, the performance can be tested on a very fast problem, and since the computing overhead of the genetic algorithm operations is low, many repetitions of this test can be performed to enable confidence in the results. Conversely, with Kriging, even if the test problem is extremely quick to interrogate (as it was in Chapter 8, since results were being called from a pre-calculated library), there is a substantial overhead associated with building and interrogating the Kriging model, and this makes doing multiple runs very time consuming.

Because of the desire to run enough different tests to be confident in the results, attempts were made to reduce the time-cost of Kriging without affecting the efficacy of the methodology too adversely.
Since the most time-consuming part of the Kriging process is optimising the vector of $\theta$ values, this was the part that was focussed on. Using the same design problem as described in Chapter 8, Kriging models were built and tuned with different population sizes, numbers of generations and probabilities for the various genetic operators tested. These models were then tested for their prediction of CO$_2$ or cost at 1000 randomly selected points in the design space, and the total mean difference between the predictions made by the Kriging model and the real output from the main models was calculated. It was found that the population size could be reduced substantially, from 100 to 40, without affecting the predictive abilities of the Kriging model too adversely (there was a 5% increase in the mean difference between the Kriging models and the main models associated with this change in population size). Since this change allowed the Kriging optimisations to be run substantially faster, it was adopted for all experiments except the first set (those described in Chapter 8).

### 7.5 Optimisation using jEPlus+EA

Most of the testing of Kriging compared it against the performance of jEPlus+EA, which uses a genetic algorithm based on NSGA-II, but modified to include integer encoding, a Pareto-archive and ensuring that each design in a population is unique (Zhang, 2012). jEPlus+EA works directly on the main models for each objective; whenever a design is suggested that has not already been tested, it is simulated in EnergyPlus and put through the CO$_2$ and cost models. At the tournament stage, designs are chosen according to their dominance rank. If both designs in a tournament have the same dominance rank then the winner is chosen based on the relative crowding of each design – designs from less crowded regions of the Pareto front being selected preferentially.
7.5.1 Settings used for optimisation using jEPlus+EA

The population size used for optimisations using jEPlus+EA was 20 for the experiments described in Chapter 8, and 10 for all other experiments. This change was due to good performance being shown with a small population size using both jEPlus+EA and a similar algorithm described by Hamdy et al. (2012) in between when the first tests were done and subsequent tests.

Apart from population size, the settings used for optimisation using jEPlus+EA were left at the default values in the package. These are shown in Table 7.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value chosen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability of mutation</td>
<td>0.2</td>
</tr>
<tr>
<td>Probability of crossover</td>
<td>1.0</td>
</tr>
<tr>
<td>Duplicate individuals allowed in population?</td>
<td>No</td>
</tr>
<tr>
<td>Population initialisation method</td>
<td>Random</td>
</tr>
<tr>
<td>Population ranking method</td>
<td>No ranking</td>
</tr>
<tr>
<td>Selection operator</td>
<td>Pick 1 from 2 tournament</td>
</tr>
</tbody>
</table>

Table 7.2 – The settings used to control the behaviour of the genetic algorithm in jEPlus+EA were left at default values.

7.6 Determination of optima through brute-force analysis

For some of the experiments (those described in Chapters 8, 9 and the highly-discrete experiment in Chapter 10), the performance of all possible designs was calculated before the optimisations were run. The building models were relatively simple and quick to evaluate on EnergyPlus, and the use of the 256-core super computer at DeMontfort University’s Institute of Energy and Sustainable Development meant evaluation of all possible designs was feasible within reasonable time-scales (less than a week). This knowledge of all possible designs had several advantages; it allowed the optimisation algorithms to run more quickly, since they could call results from
a library rather than calling EnergyPlus simulations, and it allowed comparison between the designs suggested by the optimisation methods and the true optima. Of course, in a real-world situation, if computing resources allowed the computation of all possible designs as they did in this instance, it would not be necessary to use optimisation algorithms to find optimal designs. For the purposes of this thesis they enabled more tests to be run on the optimisation algorithms, and more meaningful conclusions to be drawn about the performance of those algorithms. For the rest of the design-problems used in Chapter 10, the design spaces were too large to allow evaluation of all possible designs, even with the use of the super-computer. In these cases the algorithms ran as they would in a real-world design situation – calling an EnergyPlus simulation, CO$_2$ model and cost-model run for every design evaluated by the algorithm.

Throughout this thesis, where results are called from memory in lieu of main-model evaluations, when in a real design situation a main-model evaluation would be required, this will also be referred to as "main-model evaluations".

### 7.7 Comparing the performance of different multi-objective optimisation methods

In a single-objective optimisation it is straightforward to establish the relative performance of the two algorithms by monitoring the progress of each towards a single known optimum. This is the method used to compare the performance of a genetic algorithm with and without a Kriging model in Chapter 8. In a multi-objective design problem, establishing the relative performance is more complicated because there is a set of optimal designs, rather than a single design. In addition to the proximity of estimated Pareto designs to the designs in the true Pareto set, the number and diversity of designs in the estimated Pareto set is also important. For
this reason it may not always be possible to declare that one estimated Pareto set is better than another (Zitzler et al., 2004); it may be better on some measures of performance and less good on others.

Several different metrics have been suggested for comparing the performance of different algorithms. The metrics used to evaluate the different methods used in this thesis are as follows:

- Average quality of the Pareto estimates made. This was assessed subjectively by a visual assessment of all Pareto estimates made by each method, and objectively in one of two ways:

  1. The mean Euclidian distance between each design on the estimated Pareto front and the closest design on the true Pareto front (Figure 7.2). This gives a measure of how close to the true Pareto set those designs in the estimated Pareto set are;

  2. When the true Pareto set was not known, or in instances where there were large discontinuities in the true Pareto front, the area dominated by the estimated Pareto front was used instead.\(^5\)

- Total number of unique Pareto estimates made. More solutions, if they are of equivalent quality, is advantageous in that it gives a designer more choice.

- The diversity of designs on the estimated Pareto front. This is assessed visually. It is desirable that an algorithm produces a good diversity of estimates across the Pareto front.

Of all these metrics, the quality of the Pareto estimates was taken as being the most important throughout this thesis.

\(^5\)This metric is often referred to as hypervolume, but in this thesis the maximum number of objectives is two, meaning the Pareto front dominates a 2-dimensional area, rather than a volume or hypervolume.
In such a situation, where there are several different metrics to assess the performance of different methods, and one method may be better on some metrics and worse on others, distortion of the results through confirmation bias would be all too easy. In addition to describing the performance against all three metrics for all experiments, methods employed to minimise the risk of confirmation bias are discussed before general conclusions are made in Chapter 11.7. The methods for assessing the quality of Pareto estimates are expanded upon in the following sections.

### 7.7.1 Calculation of mean Euclidian distance

The calculation of the Euclidian distance between an estimated Pareto-optimal design and the nearest design on the true Pareto front is a simple trigonometric calculation, illustrated in Figure 7.2. The nearest Pareto point is found by calculating the distance between the estimated design in question and every point on the Pareto front, and then choosing the shortest distance.

![Figure 7.2](image-url)  

**Figure 7.2** – The Euclidian distance between an estimated Pareto-point and the nearest point in the true Pareto set is calculated from the distance in the X and Y dimensions using simple trigonometry.
The mean Euclidian distance for the estimated Pareto front is calculated according to the following formula:

\[
\text{mean Euclidian distance} = \left( \frac{\sum_{i=1}^{z} |PE_i - PT_i|}{z} \right)
\]

Where

\( z = \text{total number of Pareto estimates} \)

\( PE_i = \text{the } i\text{th Pareto estimate} \)

\( PT_i = \text{the corresponding nearest true Pareto point} \)

If the true Pareto set is known then calculation of mean Euclidian distance works well as a measure of how effective a method is at finding estimates close to the true Pareto front except in cases where there are large discontinuities in the true Pareto front, where it can be unreliable. The reason for this is shown with a hypothetical example, in Figure 7.3. For this reason, and because the true Pareto front was not always known, the Euclidian distance was only used in the experiments in Chapter 8.

**Figure 7.3** – In this hypothetical example, all estimates of the Pareto front from method A are dominated by estimates from method B therefore estimates made by Method B are without exception superior. However, if the performance were to be compared based on mean Euclidian distance, Method A would appear superior since Pareto estimates from Method B in the yellow-shaded areas are a long way from true Pareto points owing to discontinuities in the Pareto front, and this skews the result.
7.7.2 Calculating the area dominated by estimates of the Pareto front

The method for calculating the area dominated by the Pareto set (or estimated Pareto set) is relatively straightforward and is explained in Figure 7.4. However, for a given comparison between two optimisation methods, it is important to make sure that all the area calculations use the same maximum values for each objective. If this is not done then the area dominated by an inferior method could exceed that of a superior method simply by having a larger maximum in one or other objective. The maximums used were the maximums from all optimisation runs of both methods on the design-problem in question.

![Diagram](image)

**Figure 7.4** – The distance between the maximum values in each objective and Pareto point 1 can be used in order to calculate the area of the blue rectangle. The distance between Pareto points 1 and 2 in terms of Objective 2, and the distance between Pareto point 2 and the maximum value in Objective 1 can be used to calculate the area of the purple rectangle. The area for each rectangle is calculated in this way, and summed in order to find the total area bounded by estimates of the Pareto set.

Figure 7.5 shows how the calculation of dominated area can illustrate the superiority of one set of Pareto points over another, even in the presence of significant discontinuities in the true Pareto front.
Figure 7.5 – So long as the same maximum values in the two objectives are used, the area dominated by the true Pareto designs (that of all the coloured sections) should always equal or exceed the area dominated by Pareto estimates. In this case method B dominates a larger area (the two darker sections) than method A (only the darkest section), because the estimates are of a superior quality. The points are the same as in Figure 7.3, illustrating that dominated area is a more reliable method of assessing which Pareto set is better. Unlike the Euclidian-distance calculation, this method can also be used when the true Pareto set is not known.

However, there are potential weaknesses in this approach too, since an advantage of one method over another at one or other end of the Pareto front can disproportionately influence the area dominated by estimates made by that method (Figure 7.6).

### 7.7.3 Visual inspection of the quality of Pareto estimates

Because both of the objective assessments of the quality of Pareto estimates suffer from drawbacks, they are combined with a visual assessment of the quality of Pareto estimates based on all Pareto estimates made by all runs of each method. This method, too, has its weaknesses, principally that it is subjective and because the presence of a design in a cloud of Pareto estimates from many runs of an optimisation does not tell us how many times that design has been found – it could have been found on just one run, or on every single run. Because of these weaknesses,
Figure 7.6 – Improvements in Pareto-estimates at either end of the Pareto front contribute disproportionately to the dominated area. The two Pareto estimates marked as stars improve on the existing Pareto front by the same amount (the same value in Objective 1, and a reduction of 1 unit in Objective 2) yet the lower Pareto estimate contributes much more area to the total dominated area calculation, since it is at one extreme of the Pareto front. The same effect exists for estimates at the other end of the Pareto front.

Visual assessment is always used in conjunction with one of the objective measures introduced above.

7.7.4 Calculation of the statistical significance of differences in algorithm performance

Genetic algorithms are stochastic in nature, i.e. a different result will be obtained for each optimisation run. This is because the results of the key operations of population initiation, tournament selection, crossover and mutation are subject to probability. Because of this, testing each method just once tells us very little about the relative performance of each; the difference in performance shown could be anomalous. Throughout this thesis the different methods are tested multiple times on each design problem, to allow conclusions to be drawn about the relative performance of different methods with statistical confidence. However, the most common method for calculating statistical significance, the student’s T test, is not suitable for the experiments used in this thesis, since it assumes that the results are normally
distributed around a mean. For means that are approaching an asymptote, such as convergence on a true Pareto set, this is not the case. Instead of the student’s T test, the analysis in this thesis uses the more robust Mann-Whitney significance test, which does not require the results to be normally distributed.
Chapter 8

Comparing the performance of Kriging optimisation and stand-alone genetic algorithms on simple design problems

This chapter describes the method and results of two experiments. The first aimed to establish the difference in performance between a genetic algorithm searching the design space using the main model and the same algorithm searching with the assistance of a Kriging surrogate model, on a single-objective optimisation problem. The design problem was chosen to be single-objective since the particular implementation of the Kriging optimisation method uses a single objective (expected improvement) even for multi-objective design problems, meaning that for multi-objective optimisations it is not possible to use the same genetic algorithm both with and without a surrogate model.

In the second experiment, the single-objective optimisation problem was turned into a multi-objective problem through the inclusion of a cost objective, and the performance of Kriging tested against an established multi-objective genetic algorithm.
In both the single and multi-objective cases, Kriging is shown to enable good estimates of optimal solutions to be made with fewer main-model interrogations than the stand-alone algorithms.

### 8.1 Overview of methods

An overview of the methods and purpose of the experiments performed in this chapter is provided below (Table 8.1).

<table>
<thead>
<tr>
<th>Experiment type</th>
<th>Methods</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-objective optimisation</td>
<td>Optimisations both with and without a Kriging model on the same optimisation problem, using the genetic algorithm supplied with the Kriging optimisation package. Four different population sizes were used and each algorithm configuration was tested ten times.</td>
<td>Comparison of Kriging and stand-alone genetic algorithms on a single-objective optimisation</td>
</tr>
<tr>
<td>Multi-objective optimisation</td>
<td>Using the same optimisation problem as the single-objective optimisation, but with the addition of a cost objective. Optimisations were performed using Kriging optimisation and jEPlus+EA. Ten runs of each algorithm were performed.</td>
<td>Comparison of Kriging and stand-alone genetic algorithms on a multi-objective design problem</td>
</tr>
</tbody>
</table>

Table 8.1 – A summary of the experimental methods used in Chapter 8.

### 8.2 Design problem

The design problem chosen for this first test was deliberately simplistic. This enabled the analysis of all possible designs so that the progress of the optimisation could be compared against a known global-optimum design. It also meant that more tests
could be run in a given period of time, to give a better picture of the performance of the methods under various different settings.

The model was a terraced house with exposed north and south aspects, comprising two floors, each 100m$^2$, with a lounge, kitchen, dining room, toilet and hall on the first floor and three bedrooms, a hall and a bathroom on the second floor. A more detailed description of the building model is provided in Appendix A.1.

Ten design variables were chosen, with three choices possible for each variable (and therefore a total design space of 59,049 possible designs, see Equation 1.1). These are detailed in Table 8.2.

Simulations were performed using EnergyPlus for a location of London Gatwick. The single objective chosen was the minimisation of CO$_2$ emissions from the energy used to condition the building (heating, lighting and cooling). A second objective of construction cost was used in the multi-objective optimisation tests.

It was decided to include cooling energy in the calculation of CO$_2$ emissions as a penalty function for overheating in order to guide the optimisation towards designs that perform well in the summer. This was not intended to accurately represent the specification of a low-carbon building design; such a design would be unlikely to employ mechanical cooling in a northern European climate. Guiding the optimisation towards designs that perform well in summer without this penalty function could have been achieved using overheating as a constraint, and this has been the approach taken by several other low-carbon building design optimisation studies (see for example Verbeeck and Hens (2007)). However, with a desire to keep the optimisation problem relatively simple, and a necessity to take summer performance into account one way or the other, it was decided to include cooling energy rather than to develop constraint functions. How to deal with overheating risk in domestic properties that would not typically employ active cooling mechanisms may become increasingly important as winter performance standards become more and more stringent,
<table>
<thead>
<tr>
<th>Variable</th>
<th>Possible choices</th>
</tr>
</thead>
</table>
| Thermal mass layer                                 | None  
0.05m dense concrete  
0.1m dense concrete |
| Southern wall window size                          | 20/50/80% of wall area glazed                        |
| Northern wall window size                          | 20/50/80% of wall area glazed                        |
| Southern wall window type                          | Double-glazed, air-filled  
Double-glazed, argon-filled  
Triple-glazed, air filled |
| Northern wall window type                          | Double-glazed, air-filled  
Double-glazed, argon-filled  
Triple-glazed, air filled |
| External door type                                 | Solid wood  
Insulated wood  
Glazed (glazing type to match northern wall window type) |
| Southern window shading (solid pergola) size       | Extends 1m from building  
Extends 1.5m from building  
Extends 2m from building |
| N, E, W external wall insulation thickness         | 0.1m  
0.2m  
0.3m |
| Southern external wall insulation thickness         | 0.1m  
0.2m  
0.3m |
| Roof and floor insulation thickness                | 0.1m  
0.2m  
0.3m |

Table 8.2 – Design variables used in preliminary research, with the possible values each one was able to take
especially if peak summer temperatures rise as predicted by climate change models (IPCC, 2013). Several studies have observed that, as building-energy performance improves, there develops a trade-off between winter and summer performance and it may be that current building-design practices and standards, for cool climates, do not take sufficient consideration of this trade-off (see for example Coley and Schukat (2002)). The formulae used to calculate annual CO$_2$ emissions from each source are detailed in Appendix B.1 and those used to calculate construction costs are detailed in Appendix B.3.

Schedules for heating, lighting, cooling, occupancy and internal gains were left as the DesignBuilder default values for each type of zone. Lighting efficiency was left at the default value of 5W/m2-100lux.

8.3 Single-objective optimisation

The genetic algorithm that is used to construct and search the Kriging model, in the method described and used in this thesis, uses the single objective of expected improvement. Since the design problem used in this first test is a single-objective one, this means a direct comparison can be made between the performance of the algorithm with and without the use of a surrogate model.

8.3.1 Determination of true-optimum design

Because the model was relatively quick to run (around 50 seconds per EnergyPlus simulation) determination of the true-optimal design was possible.

The design with the lowest annual CO$_2$ emissions had the following variable specification

- Thermal mass of 0.1m (the thickest permitted)
- North window size of 20% of external wall area (the smallest permitted)
- North window type of triple glazed (the most thermally resistive available)
- South window size of 20% of external wall area (the smallest permitted)
- South window type of double glazed with argon fill
- Door-type of glazed
- Shading size of 1m (the smallest available)
- N, E and W external wall insulation of 0.3m thickness (the thickest available)
- Southern external wall insulation of 0.3m thickness (the thickest available)
- Roof and floor insulation of 0.3m (the thickest available)

This design had predicted annual CO$_2$ emissions of 1,843kg.

### 8.3.2 Kriging model construction and interrogation

Multiple runs of both the Kriging optimisation and the optimisation using the stand-alone genetic algorithm were done. The Kriging model was built after first sampling the design space 5, 10, 20 or 50 times. After a Kriging model had been built to fit this first set of data (using a genetic algorithm to find the values of $\theta$ that best fit the data) it was searched using the genetic algorithm for designs that maximised the expected improvement in the objective function. The genetic algorithm was set up to run with a population size of 100 and for 20 generations. At the end of these runs, the design with the greatest expected improvement (based on the Kriging model) was tested on the CO$_2$ model (which estimated CO$_2$ emissions based on energy demands calculated in EnergyPlus). The result from the EnergyPlus evaluation was added to the previously sampled points and the Kriging model re-built then searched again for another 50 generations. This meant that 2,000 evaluations were made on
the Kriging model for every EnergyPlus evaluation. This loop was set to run 150 times and was analysed afterwards to see whether, and when, the global optimum had been found.

**8.3.3 Genetic algorithm settings**

Some authors have found that the performance of genetic algorithms is relatively robust to different parameter settings except for population size (Wright and Alajmi, 2005). Because of this, and because the number of tests that could be performed was limited by time, it was decided to alter only population size in the stand-alone GA, and to leave other parameters unchanged for this test. Three different population sizes were tested on the stand-alone GA; 5, 10, 20 and 50 individuals. The starting population in each case was the same as the initial sample used in the Kriging optimisation. A total simulation budget of 500 evaluations was set, and the number of generations adjusted to meet this. All other parameters were left as the default in the genetic algorithm, and were the same when used on both the Kriging optimisation method and the stand-alone GA. These settings are listed in Table 7.1.

The genetic algorithm used in the Kriging optimisation was kept the same in all experiments (the population size was not varied), with a population size of 100 and running for 20 generations in between update points. These settings are described in more detail in Section 7.4.2.

**8.3.4 Experimental protocol**

Each experimental set-up was tested ten times. Because the performance of optimisation algorithms is stochastic, it is necessary to perform tests repeatedly in order to have confidence in the results obtained.
8.3.5 Results

The results are shown in Table 8.3. As can be seen, for all population sizes, the use of a Kriging model reduces the number of main-model evaluations required to find the true optimum. These results are tested for significance using a Mann-Whitney U test and found to be highly significant.

<table>
<thead>
<tr>
<th>Initial sample and population size</th>
<th>Mean number of main-model evaluations required to find optimum. Mean (± standard deviation)</th>
<th>P value from two-sided Mann-Whitney U test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Without Kriging model</td>
<td>With Kriging model</td>
</tr>
<tr>
<td>5</td>
<td>1154 (± 887)</td>
<td>84 (± 38)</td>
</tr>
<tr>
<td>10</td>
<td>1325 (± 1612)</td>
<td>68 (± 36)</td>
</tr>
<tr>
<td>20</td>
<td>584 (± 348)</td>
<td>88 (± 59)</td>
</tr>
<tr>
<td>50</td>
<td>625 (± 409)</td>
<td>100 (± 45)</td>
</tr>
</tbody>
</table>

Table 8.3 – Results of the ten single-objective optimisation runs with and without using a Kriging model for four different population sizes. The improvement offered by Kriging is statistically significant in all cases to a very high degree of confidence (less than a 0.1% probability that such differences in performance could be due to chance).

The improvement between the best-performing Kriging optimisation (that with an initial sample size of 10) and the best-performing stand-alone GA (that with a population size of 20) is just over an 8-fold reduction in the number of main-model evaluations required to find the true-optimum design.

How would the two methods compare if the number of main-model evaluations was severely restricted by the computing budget available? Figure 8.1 shows the average progress for the best-performing Kriging method and the best-performing stand-alone GA. The error bars represent one standard deviation above and below the mean.

---

1As can be seen, the standard deviation for the Kriging optimisation goes below the known optimum on the last two error bars. This is clearly nonsense and indicates that the data cannot be normally distributed, confirming the decision to use a Mann-Whitney U test instead of a T test in order to test statistical significance.
Figure 8.1 – The best-performing Kriging optimisation (that with an initial sample size of 10) finds the known optimum quicker than the genetic algorithm operating without a Kriging model (population size of 20). The error bars represent one standard deviation.

8.3.6 Discussion

On first examination, the advantage offered by Kriging on this optimisation problem is dramatic - more than an 8-fold improvement in the rate of finding the known optimum. However, the improvement in real-world single-objective optimisation problems may differ considerably from the improvement seen in this experiment for several reasons:

- The genetic algorithm is set-up to work well at searching a Kriging model, and may not work well at searching main models directly

The genetic algorithm used for both the Kriging optimisation and the stand-alone GA optimisation was designed to work with continuous variables, since these are what is used in the Kriging model (the GA was written for use on the Kriging model). Each variable is scaled to between zero and one, and the corresponding value for the variable in the building model chosen according to the logic shown in Formula 7.4.3.
This means that the genotype of two individuals in the population may be different when viewed by the genetic algorithm but still result in the same building design since the values for each variable are not sufficiently different to result in different choices for any of the variables. This is a problem that applies to both the Kriging and the stand-alone GA methods, but it may be more of a problem on the stand-alone GA since the use of expected improvement as the objective in the Kriging optimisation will tend to guide the search to new areas of the design space, reducing the number of duplicate designs chosen because of this coding approximation.

- Duplicate designs have been counted as main-model evaluations

With the genetic algorithm set up as it is, many individuals in one population will be exactly the same as individuals in the preceding population. In the experiment described, such duplicates were simulated again. In a real-world situation with a time-consuming main model this duplication would be easy to avoid – designs could be checked to see if they were duplicates, and if so would not trigger another simulation run, instead the result would simply be called from memory. In this experiment there were likely to be duplicate simulations done on both the Kriging and the stand-alone GA, but the problem was likely much worse on the stand-alone GA, since the use of expected improvement as an objective function in the Kriging optimisation encourages a balance between exploration and exploitation of the design space (Forrester et al., 2008).

- The design-problem may be too simple

The design problem chosen for this first test may have been too easy, as a single-objective optimisation, to allow a good test between different optimisation methodologies. Looking at the variables in Table 8.2, one might reasonably predict before even starting the optimisation, if costs are not restricted, that a domestic building in the northern hemisphere would benefit from as much insulation as possible, from
small, well-insulated windows on the northern aspect and from as much thermal mass as possible. That leaves only southern window size, type and shading as variables a designer might have doubts about with regard to the optimal assignation. Looking at the design problem retrospectively reveals it to be perhaps even more simplistic than this. Of the ten variables, only in the case of the southern window type was the optimum assignation for that variable not either the maximum or minimum available. Variables whose optimum assignation is either a maximum or minimum (monotonic variables) may be particularly advantageous to a Kriging modelling approach since the shape of the model in respect to that variable will be relatively simple rather than containing local optima.

- The genetic algorithm may not be set up optimally

Much work in optimisation research has been devoted to improving the performance of optimisation algorithms (see for example work by Wright and Alajmi (2005) and Deb et al. (2002)). The algorithm used in this experiment was very simplistic and didn’t have any of the more advanced characteristics used by, for example, NSGA-II. It may be that this does not matter so much for searching the Kriging model – searching this is so fast that many thousands of evaluations can be made, so an inefficient algorithm is acceptable. Indeed, in the case of surrogate model optimisation, because so many evaluations of the surrogate were made, it seems likely that the performance of the optimisation was limited much more by the ability of the surrogate model to capture the nature of the true function sufficiently accurately than by the efficiency of the algorithm searching the surrogate. Conversely, for the optimisation without the surrogate model, the efficiency of the search algorithm was the only thing that could affect the method’s efficiency on a given problem.

2In the case of the door type this is less clear, but the way the coding for the optimisation was written, the glazed-door option was indeed at one end of the three choices.

3Although NSGA-II is a multi-objective algorithm, so direct comparisons are difficult to make. Indeed most of the work on improving algorithms in recent years has been on multi-objective algorithms.
A fairer test of Kriging optimisation against a stand-alone genetic algorithm should address all three of these problems – use a more complicated design problem, deal with duplicate designs efficiently and use a genetic algorithm recognised for its reliable high performance. These shortcomings will be addressed in the following Section and subsequent chapters.

8.3.7 Conclusion

On a simple, single-objective design problem, the genetic algorithm provided with the Matlab Kriging package accompanying the book by Forrester et al. (2008) was used to search either the main CO$_2$ model directly for promising designs (the stand-alone GA method) or a Kriging model of the main CO$_2$ model for promising update points, which were then tested on the main CO$_2$ model and added into the Kriging model (the Kriging optimisation approach). For this design problem, the Kriging optimisation method was shown to be capable of finding a known optimum design with significantly fewer main-model evaluations than the stand-alone GA method.

However, several significant criticisms can be made of the design problem and setup of the experiment, such that drawing solid conclusions about the performance of Kriging optimisation methods compared to stand-alone GA methods is problematic. Subsequent experimental designs attempt to address these criticisms.

8.4 Multi-objective optimisation

Even with a relatively simple design-problem such as that posed in the previous section, the addition of a financial cost objective (for construction) made the optimisation much more complicated. The addition of a cost objective introduced trade-offs to be made between different variables based on how cost-effectively they contribute to an improvement in the CO$_2$ objective. Instead of searching for a single
optimum design, the search became for the Pareto-set of designs that describe the optimal trade-off between construction cost and annual CO$_2$ emissions.

While the use of a single-objective optimisation as a test for different methods allowed a direct comparison between a stand-alone GA and the Kriging optimisation method, since the same genetic algorithm could be used on both, it also introduced some problems, as described in Section 8.3.6. By transforming the design problem into a multi-objective optimisation, the optimisation was not only made more challenging but more advanced optimisation algorithms could be used. The stand-alone algorithm chosen for comparison with the Kriging methodology was jEPlus+EA, a modified implementation of the well-established NSGA-II algorithm developed by Deb et al. (2002). The modifications to the standard NSGA-II are to keep all members of the Pareto front in the population from one generation to the next (in the original code there is a chance that Pareto-individuals can be lost, in jEPlus+EA they remain within the population, meaning the population size grows over the generations), to use integer encoding (thus avoiding the problem of many genotypes being able to code for one phenotype, as described in Section 8.3.6) and to ensure that each new member added to the population has not previously been evaluated. Those individuals which stay in the population from one generation to the next (due to their membership of the Pareto set) are not re-evaluated at each generation, instead their performance on each objective is stored in memory and called from there, in order to avoid unnecessary main-model evaluations. This algorithm has previously shown good performance on building-design optimisation problems (Zhang, 2012) and has similar modifications to those made and tested more extensively against the standard NSGA-II algorithm (showing an improvement) by Hamdy et al. (2012).

The initial sample size used to build the Kriging model was 60. The results were compared for both optimisation methods after a total of 200 main-model evaluations. This figure was selected somewhat arbitrarily as a value that was small enough
that the Kriging models would not be too time consuming to run (many runs were required to establish typical performance).

8.4.1 Financial cost objective

The cost model was relatively simplistic, including only the material costs of different choices, based on estimates from industry. These costs are shown in Appendix B.3.

8.4.2 Exhaustive search of design space

Because the CO$_2$ emissions for every design were already known (from the single-objective optimisation), the calculation of the financial cost for each objective enabled the performance of every design to be established against both objectives, and from this the true Pareto set of optimal designs to be established. This knowledge of the true Pareto set allowed a more precise evaluation of the performance of the different algorithms.

8.4.3 Adjustments to try and improve the performance of Kriging

8.4.3.1 Duplicate design handling in Kriging

Because the Kriging model sees each variable as continuous (each variable taking a value between zero and one) whereas for this design problem there were in fact only three choices for each variable, when the optimisation suggested an update point as being from a region of the Kriging model that predicted a good expected improvement, it was, in some cases, a design that has already been run on the main model.

For example, taking a simplified version of the design problem in which only the first three variables are included, all three variables $[X_1...X_3]$ being equal to 0.01
would result in exactly the same design as all three variables being equal to 0.32; a design with no thermal mass layer and 20% glazing on both northern and southern aspects.

The standard Kriging methodology was adjusted so that each design suggested by the Kriging optimisation is first tested to see if it was, in fact, the same as a previously tested design. Two different methods for handling these duplicate designs were tested:

- Duplicate designs were discarded and the Kriging model searched again for another update point

or

- The value for the objective functions (financial cost and CO₂) was called from memory and included as an update point in the Kriging model

The results of these different proposed methods are included in Table 8.4 as “Duplicates excluded” and “Duplicates included”, respectively.

8.4.3.2 Adjusting the encoding length to reduce the incidence of duplicates

In the standard Kriging code accompanying the book by Forrester et al. (2008), the encoding length for each variable is 20. This means that for every value of X (every variable) there are over a million possible values. When there are in fact only a few discrete choices this is wasteful, and may increase the instances of duplicate designs being chosen as update points during the optimisation on the Kriging model. Reducing the encoding length to three (giving eight possible choices for each variable) and two (giving four possible choices for each variable) was tested as a way of improving the performance of the Kriging optimisation on discrete design problems.
8.4.3.3 Results – different Kriging configurations

The results of the Kriging optimisations using the various different set-ups are shown in Table 8.4.

<table>
<thead>
<tr>
<th>Kriging set-up</th>
<th>Mean Euclidian distance from nearest true Pareto design after 200 main-model evaluations (±mean standard deviation in Euclidian distance)</th>
<th>Mean number of estimated Pareto solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicates included, 20-bit encoding</td>
<td>9.5(±13.5)</td>
<td>29.5</td>
</tr>
<tr>
<td>Duplicates excluded, 20-bit encoding</td>
<td>9.3(±15.6)</td>
<td>30</td>
</tr>
<tr>
<td>Duplicates included, 3-bit encoding</td>
<td>6.8(±11.9)</td>
<td>34.7</td>
</tr>
<tr>
<td>Duplicates included, 2-bit encoding</td>
<td>10.4(±17.0)</td>
<td>25.4</td>
</tr>
</tbody>
</table>

Table 8.4 – Optimisation performance results for the different set-ups of Kriging used

The best performing set-up for the Kriging optimisation was with duplicates included, and with an encoding length of three (Table 8.4).

<table>
<thead>
<tr>
<th>Encoding length</th>
<th>Mean number of duplicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>54</td>
</tr>
<tr>
<td>20</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 8.5 – Shortening the encoding length to 2 did result in a statistically-significant (at the 99% confidence interval) reduction in duplicate sample points selected. However, there was no statistically-significant difference between the number of duplicates selected with an encoding length of 3 compared to that of 20.

Shortening the encoding length to three did not reduce the number of duplicate designs chosen by the Kriging method, but shortening it again to two did reduce it slightly, compared to an encoding length of 20.
8.4.3.4 Discussion

The results shown in Table 8.4 indicate no significant difference between the performance of the optimisation when duplicates were included in the Kriging model and when they were excluded. Initial analysis (as reported in Tresidder et al. (2012)) indicated that there was a larger (but still not statistically significant) advantage gained by including the duplicates in the model, and it was on the basis of this analysis that it was decided to include duplicates for subsequent experiments in this thesis. This was a mistake due to a failure to remove duplicate designs from the estimated-Pareto set. Since it is the most promising locations that will tend to produce duplicate designs (because the model predicts a high expected improvement near samples that have previously performed very well), not removing duplicates has the tendency to skew the average downwards, since duplicate points in the Pareto set tend to be very close to the true Pareto front.

There are, however, good theoretical grounds for wanting to include duplicate points in the Kriging model – if duplicate points are not included, then the Kriging model may get stuck in a loop where it repeatedly suggests designs that, according to the Kriging model, are slightly different to any previously-tested designs, but in actual fact are duplicates.

Shortening the encoding length from 20 to 2 reduced the number of duplicate points selected by the Kriging method. This reduction was statistically significant\(^4\). However, it also coincided with the worst-performing of any of the Kriging configurations tested in Table 8.4. This could be due to a mismatch between the number of values encoded by each variable with a bit length of 2 (4 choices) and the real number of possible choices for each variable (3 choices). It may be that, because the “spare” value present with an encoding length of 2 will fall on just one choice of the variable, increasing the likelihood that this choice will be selected, this has a distorting effect

\(^4\)P value of 0.0062 using a T test.
on the optimisation process. Methods to alleviate the problem of this “spare” value were not pursued. Shortening the encoding was not pursued subsequently in this thesis.

Based on the performance in Table 8.4, it was decided to compare the best performing Kriging configuration in more detail against the performance of jEPlus+EA.

8.4.4 Results – comparing the performance of Kriging with jEPlus+EA

8.4.4.1 Quality of Pareto estimates

By plotting the mean Euclidian distance between members of the estimated Pareto population and the nearest members of the real Pareto population after different numbers of main-model evaluations, the methodologies of Kriging optimisation and jEPlus+EA perform can be compared to one another (Figure 8.2).

![Figure 8.2](image)

**Figure 8.2** – The Kriging optimisation makes better estimates of the Pareto set than does the jEPlus+EA if the number of main-model evaluations is restricted to 200. The performance of jEPlus+EA catches up with that of Kriging around 400 main-model evaluations.

100
Figure 8.2 shows that, below 200 main-model evaluations, Kriging had an advantage over jEPlus+EA; the mean Euclidian distance was lower, and the standard deviation in Euclidian distances was also lower (i.e. the performance is more consistent). It took until around 400 main-model evaluations before the performance of jEPlus+EA equaled that of Kriging with 200 main-model evaluations. The improvement shown by Kriging when the number of main-model evaluations was limited to 200 was significant to the 99% confidence level.

8.4.4.2 Spread of designs on the estimated Pareto front

The spread of Pareto estimates made after 200 iterations (Kriging) and 400 iterations (jEPlus+EA) was similar (Figure 8.3). This number of iterations was chosen because this was the equivalent point in mean Euclidian distance shown in Section 8.4.4.1.

Figure 8.3 – Pareto estimates made by all runs of both methods, with the Kriging optimisation running for 200 main-model evaluations and the jEPlus+EA optimisation running for 400.

The use of standard deviation is, for the same reasons as described for the single-objective case, slightly misleading, since the data is unlikely to be normally distributed. However, it serves to give us a way of comparing the relative precision of the two methods.

Using a Mann-Whitney test
8.4.4.3 Number of unique Pareto estimates made

Kriging showed an advantage at 200 main-model evaluations in terms of the total number of unique Pareto estimates made (an average of 35 for Kriging compared to 26 for jEPlus+EA), but after 400 main-model evaluations using jEPlus+EA (the equivalent point in quality of estimates) the number of Pareto estimates made was similar to Kriging after 200 main-model evaluations (35 for Kriging compared to 31 for jEPlus+EA).

8.4.5 Discussion – multi-objective optimisation

The improvement offered by Kriging in this multi-objective optimisation test, in terms of the number of main-model evaluations required to make comparable estimates of optimal designs (based on the quality, spread and number of Pareto estimates), was of the order of a halving of the required main-model evaluations. If the discrete nature of the variables in this building design optimisation caused problems for the Kriging method, as it was suggested they might do in Section 6.2.1, they were not severe enough to render its performance lower than that of jEPlus+EA.

In the interests of saving time and to enable multiple optimisations to be run, main-model evaluations were replaced in both optimisation methods with calls to a previously calculated library of all possible results. As such the time taken for the optimisations is not an accurate reflection of how long they would take in a real-world setting, in which a brute-force calculation of all possible designs would not be feasible. However, good estimates of how long the optimisations would have taken if they had been run on the main models, including simulation calls to EnergyPlus, can be made relatively easily. Given that the EnergyPlus model was relatively simple and quick to run (approximately 50 seconds⁷), and because the extra time

⁷This is the time required to run the EnergyPlus simulations. The time required to run the CO₂ model from the demand data generated by EnergyPlus, and to run the cost models, was negligible.
required to run the Kriging model was considerable (approximately 15 hours for these experiments), the jEPlus+EA optimisation would actually be considerably quicker than the Kriging optimisation; approximately 5.5 hours (400 simulations each taking ~50 seconds) compared to nearly 18 hours for Kriging (200 simulations each taking ~50 seconds, plus a 15 hour overhead). In a hypothetical optimisation with a more time-consuming EnergyPlus model but showing a similar reduction in main-model evaluations through the use of Kriging, the two methods would be expected to take a similar amount of time if the EnergyPlus models required 4.5 minutes to run (the Kriging overhead of 15 hours divided by the number of additional simulations required by jEPlus+EA), and for Kriging to show a true time-advantage for models that are more time consuming still. These figures differ slightly from those reported in Tresidder et al. (2012) for the reasons discussed in Section 8.4.3.4.

Looking at the shape of the Kriging progression in Figure 8.2, it seems likely that, had it been allowed to continue past 200 main-model evaluations, then the estimated Pareto set of designs would have continued to improve. The decision to limit the Kriging optimisation to 200 main-model evaluations was somewhat arbitrary; although it would have been possible to run the Kriging model until up to approximately 500 sample points had been added to the model, it would have been substantially more time consuming (since with each iteration the Kriging model becomes more and more time consuming). 200 main-model evaluations was chosen as a compromise between maximising the potential of Kriging on a given optimisation problem and having a method that was fast enough that multiple tests could be run in the time available. Given that when the Kriging model suggested a duplicate design as an update point this was included in the Kriging model (but not counted towards the main-model evaluations), the Kriging models actually had many more than 200 update points, up to 303 in the optimisation in which the most duplicate points were selected. Whether the same relationship between the performance of Kriging and jEPlus+EA (that of Kriging requiring roughly half the main-model evaluations
required by jEPlus+EA) would have held true if the Kriging model had been allowed to run for considerably longer is not clear.

As described in Section 8.3.6, the building model used in this optimisation was relatively simple, and it may be that, even with the addition of financial cost as an objective, the design space was not complex enough to allow a thorough testing of the two different optimisation methods.

8.4.6 Conclusion – multi-objective optimisation

Kriging optimisation methods have been shown to be capable of handling variables with discrete choices. The performance of Kriging optimisation was compared to that of jEPlus+EA, a building optimisation program which uses a multi-objective algorithm that has been previously demonstrated to have good performance characteristics. If the computing budget was restricted to just 200 main-model evaluations then the Kriging optimisation method outperformed jEPlus+EA – the solutions in the estimated Pareto front were typically closer to the real Pareto front than those solutions found by jEPlus+EA, more Pareto estimates were made and there was no significant difference in the spread or number of solutions found between the two methods. In order that the estimated Pareto set found by jEPlus+EA was approximately equivalent in quality to that found by Kriging after 200 main-model evaluations, around twice as many main-model evaluations were required.

However, with the building model being relatively fast to interrogate (50 seconds) the reduction in main-model evaluations enabled by Kriging did not compensate for the additional time required to run the Kriging model. In a hypothetical situation with a design problem that was similarly complex to search but a building model that was more time consuming to run Kriging could be expected to show a true advantage over jEPlus+EA for building models requiring more than 4.5 minutes to run.
The optimisation problem chosen may have been too simplistic to offer a good test of the relative speeds of both methods, and from visual inspection of the progression curves it appears the Kriging optimisation would have continued to progress if it had been left to do so. These two shortcomings are addressed in subsequent experiments.

8.5 Summary

Kriging optimisation was tested against stand-alone genetic algorithms on both single- and multi-objective design problems with discrete parameters. In the case of the single-objective design problem, the algorithm used in both the Kriging optimisation and the stand-alone optimisation was the same. In the case of the multi-objective optimisation, Kriging was tested against an established algorithm which incorporated some of the latest developments in multi-objective optimisation and had already shown good performance on other low-carbon building design optimisation problems. Kriging was able to effectively handle the discrete variables used in this test, and was able to make good estimates of optimal designs with fewer main-model evaluations than the stand-alone genetic algorithm. Several conclusions were drawn early in the research that were based on an incomplete analysis of the data; these are corrected in this chapter and are minor enough not to significantly impact the overall conclusions or direction of the research. The experimental set-up for both experiments could be improved and this is addressed in subsequent chapters.
Chapter 9

Comparing the performance of Kriging optimisation and jEPlus+EA on a more complex design problem

The experiments in the previous chapter showed an advantage for Kriging optimisation in terms of a reduction in main-model evaluations required to make good estimates of Pareto-optimal designs. This chapter describes the method and results for a similar experiment on a more complex multi-objective design problem. The results show that Kriging offers a much smaller advantage on this design problem than in the previous chapter.

9.1 Overview of methods

An overview of the methods and purpose of the experiments performed in this chapter is provided below (Table 9.1).
Experiment type | Methods | Purpose
---|---|---
Multi-objective optimisation | Optimisation of a 21-year CO$_2$ and cost multi-objective optimisation using both Kriging and jEPlus+EA methods. Variables were a mix of discrete choices (such as window type) and discrete values picked from continuous ranges (such as % glazing). | Comparison of Kriging and stand-alone genetic algorithms on a a more complicated multi-objective design problem containing a mix of variable types.

**Table 9.1** – A summary of the experimental methods used in Chapter 9.

### 9.2 Building optimisation problem

The design problem used as the test-bed for the optimisation in Chapter 8 was suspected to be too simple to offer a thorough test of the optimisation methods, so this second test was designed to be more difficult for optimisation methods. Instead of having many variables for which the optimum assignation was made complicated only by the inclusion of a second objective of financial cost, the variables were chosen to be more challenging. In addition, instead of annual CO$_2$ emissions from the heating, lighting and cooling of the building, the CO$_2$ emissions objective was changed to one of total emissions over a 21 year period, including both in-use emissions (from heating, lighting and cooling the building) and embodied emissions (emissions caused by the production and construction of the building). The building model was a terraced domestic property with south and north facing exposed facades with the appropriate weather file chosen for the isle of Jersey.

#### 9.2.1 Design variables

The design variables were as follows (Table 9.2).
Variable | Options available
--- | ---
Building shape | choice of narrow (12m deep by 5.33m wide), square (8m by 8m) or wide (5.33m deep by 12m wide) floor plans
Southern aspect percentage glazed | 20%, 50% or 70%
Northern aspect percentage glazed | 20%, 50% or 70%
Southern window type | double glazed air filled, double glazed argon filled or triple glazed argon filled
Northern window type | double glazed air filled, double glazed argon filled or triple glazed argon filled
Construction type | heavyweight, lightweight or pre-fabricated straw panels
Insulation type (for heavyweight and lightweight construction types) | expanded polystyrene (EPS) or mineral wool
Insulation thickness (for heavyweight and lightweight construction types) | 5, 15 or 25cm
Air tightness | scheduled at 0.6 or 1 air changes per hour
Southern window shade depth | 0.5m, 1m or 1.5m
Lighting | LED or CFL throughout
Heating system | 90% efficient gas boiler, air-source heat pump or ground-source heat pump
Solar hot water | absent or sized to meet 2/3 of hot-water demand (hot-water demand was constant for all designs)
Photovoltaic panels | 0m², 10m², 20m² or 30m²

**Table 9.2** – Variable options available in the design problem.

This gave nearly a million possible unique designs. It was hoped that a larger design space, with this mix of variables, along with the more complex CO₂ objective of total emissions over a 21-year period, would give both a more realistic and a more challenging test to the optimisation algorithms.
9.3 Establishment of true Pareto set

Although the design space for this optimisation problem was considerably larger than that of the optimisation in Chapter 8\(^1\), the effect of some design variables was calculated outside of EnergyPlus using simple arithmetic formulae. This was the case for the effect of different heating systems, and the presence or absence of solar hot water heating and the size of photovoltaic panels. All of the possible combinations of those three variables were calculated using the demand data from just under 38,000 EnergyPlus simulations. The calculations for embodied CO\(_2\) and cost also involved simple arithmetic formulae. Although the building model was more time consuming to run than that used in Chapter 8, 38,000 simulations was still feasible using a large, multi-processor computer. Knowledge of all-possible designs meant that the optimisations could be run more quickly as they called results from memory rather than requiring an EnergyPlus simulation, and allowed more precise evaluations of the performance of different optimisation methods.

9.4 Optimisation using Kriging surrogate modelling

The Kriging surrogate optimisation method was used to search for the best Pareto-front, as in Chapter 8. The initial sample size was 60, and the optimisations were run for a total of 310 main-model evaluations.

The first requirement was to reduce the number of variables to levels that could reasonably be handled using Kriging multi-objective optimisation. Although 20 is recommended as a guide for the maximum number of variables to use in single-objective optimisation problems, a reduction from the 14 listed in Section 9.2.1 was judged prudent for multi-objective optimisation. In order to reduce the total number of design variables, the variables of construction type, insulation type and

\(^1\)Over 15 times as many possible designs
insulation thickness were combined into one variable with 13 possible choices (all possible combinations of construction type, insulation type and insulation thickness). This left a total of 11 design variables.

This amalgamation of variables into one variable could, in theory, be problematic for Kriging optimisation, and so some care was taken in how the choices for these variables were represented numerically. The first six options on the chromosome were all lightweight construction, the next six were heavyweight, and the final option was straw-bale panels. Within the lightweight and heavyweight subdivisions there was first EPS and then mineral wool insulation at each of the three thicknesses. It was intended that within the three subdivisions (lightweight, heavyweight and straw-bale) the key properties (U value, embodied carbon and price) would change in a relatively smooth manner. However, there are a number of different ways in which this chromosome could have been constructed, and the method chosen was necessarily somewhat arbitrary.\(^2\)

Since, in the previous experiment, the progress of the optimisations indicated that they would have continued to progress if left to run for longer, for this experiment it was decided to run the optimisation for a total of 310 main-model evaluations. This is still quite considerably below the maximum of 500 sample points\(^3\) in the Kriging model recommended in Forrester et al. (2008), due to the continuing desire to minimise the time required for each optimisation in order to permit multiple runs. All settings were left unchanged from those used in Chapter 8, except the population size used to search for the optimal vector of \(\theta\)s, which was decreased to 40, following the tests described in Section 7.4.4, and the encoding length of the genetic algorithm, which was increased back to the default value of 20, since some of the variables in the design problem had more than 8 possible values, and could

\(^2\)Since it would not have been possible, given the time requirements of testing Kriging optimisation methods, to try out every possible way in which this chromosome could be constructed.

\(^3\)The number of sample points in the Kriging model was actually slightly higher than 310, since some of the points included were duplicate designs, due to the discrete nature of the design variables.
not therefore be effectively coded for with an encoding length of 3. Since algorithms using very short encoding lengths will work only on variables with few choices, and in light of the lack of statistically-significant improvement in optimisation outcomes through using shorter encoding lengths, it was decided to leave the encoding length at the default of 20 to maintain the applicability of the code.

In order to establish the average performance of the method, ten independent runs were made of the Kriging optimisation.

### 9.5 Optimisation using jEPlus+EA

As in Chapter 8, the same design problem was optimised using jEPlus+EA. The settings were left as used in Chapter 8, except that the population size was reduced to 10, based on promising results on a very similar algorithm (Hamdy et al., 2012) and tests done by the developers of jEPlus+EA in the time between the preliminary experiments and this one, which both indicated that smaller population sizes could provide good estimations of Pareto designs in situations where the total number of main-model evaluations was limited. The algorithm was run for 200 generations, meaning that a total of 2,000 main-model evaluations were performed for each run of the algorithm. As with the Kriging optimisation, ten independent runs were made in order to establish the average performance of the method.

### 9.6 Results

#### 9.6.1 Quality of Pareto estimates assessed using dominated area

The purpose of using a surrogate model of any type when a more robust model exists is to reduce the number of evaluations of that main model, and thus the time
taken to achieve good estimates of optimum designs. The experiments conducted in Chapter 8 showed Kriging could make comparable-quality estimates of Pareto-optimal designs with approximately half the number of main-model evaluations required by jEPlus+EA, up to the limit of 200 main-model evaluations included in the Kriging model. The experiment performed in this chapter showed a similar, but less significant, performance advantage for Kriging over jEPlus+EA, in cases where the sampling budget was below 300 main-model evaluations (Figure 9.1 and Table 9.3).

Figure 9.1 – For the more complex design-problem posed in this section, Kriging shows an advantage for the first few-hundred evaluations, but by 300 evaluations the performance of jEPlus+EA has all but caught up. jEPlus+EA subsequently improves significantly on the performance achieved by Kriging after 310 main-model evaluations. After 2,000 evaluations (not shown in Figure) the average area dominated by the Pareto set estimated by jEPlus+EA is nearly 99% of the true Pareto set. The error bars represent one standard deviation to either side of the mean.
Table 9.3 – Progression of Kriging and jEPlus optimisations along with statistical significance of the differences in performance. This table shows the same results as in Figure 9.1, along with the P value, calculated using a Mann-Whitney test, for the difference in the area dominated by the two methods.

* This value is the likelihood of the observed difference in dominated area coming about by chance. Values less than 0.05 indicate statistical significance at a 95% confidence level.

<table>
<thead>
<tr>
<th>Number of main model evaluations</th>
<th>% of true Pareto area dominated by estimates made by Kriging</th>
<th>% of true Pareto area dominated by estimates made by jEPlus+EA</th>
<th>P value*</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>80.5</td>
<td>82.3</td>
<td>0.3075</td>
</tr>
<tr>
<td>110</td>
<td>90.4</td>
<td>87.3</td>
<td>0.0113</td>
</tr>
<tr>
<td>160</td>
<td>92.7</td>
<td>90.1</td>
<td>0.0022</td>
</tr>
<tr>
<td>210</td>
<td>93.5</td>
<td>91.6</td>
<td>0.0022</td>
</tr>
<tr>
<td>260</td>
<td>94.2</td>
<td>93.0</td>
<td>0.0036</td>
</tr>
<tr>
<td>310</td>
<td>94.5</td>
<td>94.2</td>
<td>0.1041</td>
</tr>
</tbody>
</table>

Figure 9.1 and Table 9.3 show that, early on in the optimisation process, Kriging made better estimates of the Pareto front than jEPlus+EA. This advantage was greatest at around 150 main-model evaluations; it typically took 240 main-model evaluations for jEPlus+EA to reach the same quality of Pareto estimates as found by Kriging after 150 main-model evaluations. However, after this, the advantage shown by Kriging diminished, and at 310 main-model evaluations, the stopping point for the Kriging optimisation, there was no significant difference in the area dominated by Pareto sets from the two methods (Table 9.3). Furthermore, at 310 evaluations the performance of jEPlus+EA was more consistent than that of Kriging (the standard deviation in dominated area is lower). Shortly after 310 main-model evaluations jEPlus+EA started to find better Pareto sets than Kriging had found in 310 main-model evaluations (the Kriging optimisations were not run past 310 main-model evaluations).
9.6.2 Visual assessment of the estimated Pareto front

The spread of all Pareto estimates made by both methods after 310 evaluations appears similar (Figure 9.2). If there is a difference then it appears to be that Kriging made more estimates in the lower right (more expensive and lower CO$_2$ emissions) of each cluster of Pareto designs than jEPlus+EA did.

![Figure 9.2](image)

**Figure 9.2** – The spread of all Pareto estimates after 310 iterations appears similar for both Kriging and jEPlus+EA optimisations, with perhaps a slight advantage to Kriging in finding designs in the lower right of each cluster of Pareto designs. The black line represents the true Pareto front.

It is difficult to decide, visually, if there is a difference in the quality of the estimates (in terms of how close they are to the true Pareto front) between the two methods. There seem to be some regions of the solution space where Kriging made superior estimates, and others where jEPlus+EA made superior estimates. This fits with the analysis of dominated area in Section 9.6.1 that the difference in performance at 310 main-model evaluations is not large.
9.6.3 Number of unique Pareto estimates

There was no significant difference in number of unique Pareto estimates made by both methods after 310 main-model evaluations, with Kriging making, on average, slightly more (43 compared to 39).

9.7 Discussion

Kriging offered a much smaller advantage on the design-problem in this chapter, in terms of a reduction in main-model evaluations required to make comparable-quality estimates of Pareto-optimal designs, than it did in Chapter 8. This could be for one of or a combination of several reasons.

It could simply be that the design space was more complex than the design space explored in Chapter 8.

It could be that the Kriging optimisation was stopped too early to show its true performance advantage. As described in Section 9.4, the stopping criteria for the Kriging optimisation was set below the maximum number of main-model evaluations recommended in the literature in order to allow time for multiple runs of the experiments. Perhaps running it for longer would have changed the relationship between the performance of Kriging and jEPlus+EA. However, examining the shape of the two curves in Figure 9.1, that of Kriging appears to be flattening off, while that of jEPlus+EA is continuing to rise. It does not seem likely, from Figure 9.1, that continuing to run the Kriging optimisations for longer would have changed the conclusions drawn regarding the magnitude of advantage offered by Kriging over jEPlus+EA.

It could be due to the variables being more discrete than those used in Chapter 8. Although the variables in the design problem posed in Chapter 8 were, superficially, discrete (there were only three choices for each one), the underlying function of all
but three variables (window type on two aspects and door type) was continuous (it would have been equally easy to represent the variables with a hundred or a thousand choices for each). In contrast, more of the variables in the design problem posed in this chapter (lighting system, heating system, window type, construction type, building shape) would have been difficult to represent as continuous variables. It could be that Kriging did less well on the design problem posed in this chapter because it struggled to handle these discrete-by-nature variables, or the greater proportion of them compared to the design problem in Chapter 8. A more detailed examination of the performance of Kriging and stand-alone optimisation methods on design problems with continuous variables viewed at different resolutions, and others with variables that are discrete by nature, is the subject of Chapter 10.

Given the time-cost associated with building and interrogating Kriging models, in order for it to be worthwhile using the Kriging approach on a design problem with similar characteristics to this one but longer simulation times, the simulation budget would have to be very low and the model simulation times very high.

9.8 Conclusion

Purely in terms of the number of main-model evaluations required to make comparable-quality estimates of the Pareto front, Kriging outperformed jEPlus+EA on the design-problem posed in this chapter. However, the advantage was much smaller than the advantage shown in Chapter 8, the advantage diminished as the number of main-model evaluations increased, and from the shape of the curves in Figure 9.1, it appears as though a Kriging optimisation run for longer would have likely ceased to show an advantage over jEPlus+EA. The size of the advantage offered by Kriging is small enough, when present, that in order for it to be worthwhile using a Kriging

---

4Henceforth these types of variables will be referred to as “variables that are discrete by nature”.
model on a design-problem with characteristics similar to this one, the simulation budget would have to be extremely small and model simulation times very long.

The reasons for the drop in performance shown by Kriging on this design problem, compared to the performance shown in Chapter 8, are not clear. The theory that it is due to the increased proportion of variables that are discrete by nature is tested in the following chapters.
Chapter 10

Comparing the performance of Kriging and jEPlus+EA on design problems with different variable types

This chapter systematically compares the performance of Kriging and jEPlus+EA on a range of design problems with different characteristics. Kriging is shown to offer an advantage in the design problems with continuous variable choices but conversely performs worse than jEPlus+EA on design problems with discrete variables.

10.1 Overview of methods

An overview of the methods and purpose of the experiments performed in this chapter is provided below (Table 10.1).
<table>
<thead>
<tr>
<th>Experiment type</th>
<th>Methods</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-objective optimisation of design problem with continuous variables at different resolutions</td>
<td>Optimisations using both Kriging and jEPlus+EA on a design problem with all variables being of the type that can be represented as a continuous variable, but with four different levels of resolution chosen from highly discrete to highly continuous</td>
<td>To compare the performance of Kriging and jEPlus+EA on design problems with continuous variables of different resolutions.</td>
</tr>
<tr>
<td>Multi-objective optimisation of design problem with discrete-by-nature variables</td>
<td>Optimisations using both Kriging and jEPlus+EA on a design problem in which all variables were discrete-by-nature (i.e. it would not be possible to represent them as a continuous variable).</td>
<td>To compare the performance of Kriging and jEPlus+EA on design problems with discrete variables.</td>
</tr>
</tbody>
</table>

Table 10.1 – A summary of the experimental methods used in Chapter 10.

### 10.2 Introduction

One of the principal assumptions of Kriging modelling is that the functions it is used to represent are smooth and continuous, although it is more flexible in this respect than the other surrogate modelling techniques discussed in Chapter 5, due to the greater number of model parameters that can be tweaked to make the model better fit the data (Forrester et al., 2008). This assumption that the functions are smooth and continuous may cause problems in terms of the fidelity of the Kriging model if it is used to represent design problems that contain discrete variables, as low-carbon building design problems often do (Wang et al., 2005b). The correlation between two sample points is dependent only on the distance between those points in each dimension (variable) and the values chosen for $\theta$. If sufficient samples of the design space are made, this method may adequately represent discrete variables and discontinuities, but there is a risk that on large design spaces with significant discontinuities these will be missed by the model.
The previous Chapters have shown that Kriging is able to work on optimisation problems with discrete variables, and that in some cases the use of a Kriging surrogate model enables good estimates of optimal designs to be made with fewer main-model evaluations than are required using an algorithm searching the main model directly. However, a significant advantage has not been shown in all experiments. Indeed, in the test described in Chapter 9, jEPlus+EA was able to nearly match the performance of Kriging after 310 main-model evaluations, and exceed it with slightly more evaluations. The reasons for this variability in the advantage shown by Kriging are not clear, but in the tests done so far the advantage has been smallest in those experiments in which some variables are discrete by nature, rather than merely discretisations of otherwise continuous functions. This fits with theory as described in Section 6.2.1 since the Kriging model assumes that the variables are continuous and explicitly smooths the relationship between sample points. Thus it seems reasonable to hypothesise that Kriging is able to work on variables whose underlying function is continuous, no matter how “discretised” the variables are made in the design problem, but that it may struggle with variables for which the underlying function is discrete by nature. Furthermore, since Kriging views each variable as continuous, the closer to continuous a given variable is, the greater the performance advantage offered by Kriging is likely to be.

The tests performed in this chapter aimed to take a more systematic approach to the testing of these two hypotheses. The first set of experiments tested Kriging against jEPlus+EA on a building-design problem in which the underlying nature of the variables was continuous, but which were tested at varying degrees of discreteness; either highly discrete (3 or 4 choices per variable), moderately discrete (5 or 6 choices per variable), moderately continuous (12 choices per variable) or highly continuous (100 choices for each variable). This can be viewed as the resolution chosen. The second set of experiments tested Kriging against jEPlus+EA on a building-design problem with variables that were discrete by nature (i.e. very difficult to represent
as continuous). Each experimental set-up was run 12 times for each method, in order that statistically robust observations could be made.

The design problem was based largely on the requirements of an optimisation competition run by DesignBuilder (DesignBuilder, 2012). This included a requirement that the designs should not exceed a maximum of 200 area-weighted discomfort hours per year.

10.3 Kriging configuration

The settings for the Kriging algorithm were as used in the previous chapter, except for the initial sample size, which was changed to 70, and the total number of main-model evaluations, which was 260.

10.4 Constraint handling

In this instance it was decided to handle the constraints using a penalty function whereby 30% was added to both the CO$_2$ and cost objectives if a design exceeded 200 area-weighted discomfort hours over the course of a year. There are more complex, and efficient, methods of handling constraints in which the likelihood of a design being selected to reproduce is in part dependent on whether it meets the constraints or the extent to which the constraint is violated (see for example Deb et al. (2002)). However, such methods had not been implemented in jEPlus+EA, and the methods for handling constraints are different between Kriging and optimisations without a surrogate model. Because of this it was decided to include a penalty function within the models for CO$_2$ and cost so that both the methods could be assessed on the same terms and differences in performance could be attributed to algorithm performance rather than constraint-handling differences.
10.5 Formulation of the optimisation problem

10.5.1 Building model

The building model for both sets of tests was the same, an office building with a total floor area of 3,000m$^2$, and with areas for different activities as defined in the competition specification. It was hoped that this building, and the design variables chosen, would provide a more difficult optimisation problem for the two optimisation methods. The building model is shown in Appendix A.3. The weather file chosen was for London Gatwick.

10.5.2 Tests on a design problem with continuous variables of varying resolution

For the first set of tests, 11 variables were chosen to be continuous variables (from which discrete resolutions could be taken) with the aim of producing a realistic optimisation problem. The addition of a constraint in the form of the maximum number of permitted discomfort hours made the optimisation problem more complicated still.

The 11 variables chosen for the first set of tests were as follows:

- Percent of external wall area glazed on each of four aspects (four variables)
- Building orientation relative to base case
- Insulation thickness
- Mass thickness
- Heating setpoint
- Cooling setpoint
- Natural ventilation setpoint
- Southern overhang depth

With the inclusion of temperature setpoints and building orientation, it was hoped that the design space would be more complicated than that studied in previous experiments. The windows were set to be double glazed, air filled with a low-emissivity (low-E) coating, and the blinds were set to be internal with 25mm (high reflectivity) slats.

### 10.5.2.1 highly-discrete design problem

The following values were chosen as design options for the experiment on the continuous-variable design problem with a resolution of three choices per variable, except for orientation for which there were four variables to allow for the four cardinal orientations (Table 10.2).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Options available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of each aspect glazed</td>
<td>10%, 40% or 90%</td>
</tr>
<tr>
<td>Building orientation</td>
<td>Due north, east south or west, with respect to an axis in the base-case design</td>
</tr>
<tr>
<td>Insulation depth</td>
<td>1mm, 25cm or 50cm</td>
</tr>
<tr>
<td>Mass depth</td>
<td>1mm, 40cm or 60cm</td>
</tr>
<tr>
<td>Heating setpoint</td>
<td>18°C, 20°C or 22°C</td>
</tr>
<tr>
<td>Cooling setpoint</td>
<td>24°C, 26°C or 28°C</td>
</tr>
<tr>
<td>Natural ventilation setpoint</td>
<td>20°C, 22°C or 24°C</td>
</tr>
<tr>
<td>Southern overhang depth</td>
<td>50cm, 75cm or 1m</td>
</tr>
</tbody>
</table>

**Table 10.2** – Variable choices available for the highly-discrete design problem. Four choices were included for the building orientation to allow for each cardinal aspect to be included. Insulation and mass layers of 1mm were included in the place of no insulation or mass layer, to simplify coding in EnergyPlus.

This gave a design space of 236,196 possible designs. With access to a multi-core super-computer, and a EnergyPlus runtime of around one minute, it was possible
to evaluate every possible design, as had been done in previous experiments. This enabled the estimates of Pareto-optimal designs found by the two methods to be compared to the true Pareto front.

10.5.2.2 Moderately-discrete design problem

The same design variables were used as in the highly-discrete design problem, but with more choices possible for each variable (Table 10.3).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Options available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of each aspect glazed</td>
<td>10%, 26%, 42%, 58% or 90%</td>
</tr>
<tr>
<td>Building orientation</td>
<td>0, 45, 90, 135, 180 or 270 degrees, relative to due north</td>
</tr>
<tr>
<td>Insulation depth</td>
<td>1mm, 10cm, 20cm, 30cm, 40cm or 50cm</td>
</tr>
<tr>
<td>Mass depth</td>
<td>1mm, 10cm, 20cm, 30cm, 40cm or 50cm</td>
</tr>
<tr>
<td>Heating setpoint</td>
<td>18°C, 19°C, 20°C, 21°C, 21.5°C or 22°C</td>
</tr>
<tr>
<td>Cooling setpoint</td>
<td>24°C, 24.5°C, 25°C, 26°C, 27°C or 28°C</td>
</tr>
<tr>
<td>Natural ventilation setpoint</td>
<td>20°C, 21°C, 21.5°C, 22°C, 23°C or 24°C</td>
</tr>
<tr>
<td>Southern overhang depth</td>
<td>50cm, 50cm, 70cm, 80cm, 90cm, 1m</td>
</tr>
</tbody>
</table>

Table 10.3 – Variable choices available for the moderately-discrete design problem. Note that there were only five choices available for glazing percentage.

10.5.2.3 Moderately-continuous design problem

The same design variables were used as for the highly-discrete and moderately-discrete design problems, but with 12 evenly-spaced choices between the minimum and maximum used in the highly and moderately-discrete design problems.

10.5.2.4 Highly-continuous design problem

Again, the same set of design variables were used, but in this case each one had 100 evenly-spaced choices between the minimum and maximum used for the previous design problems.
10.5.3 Tests on a design problem with variables that are discrete by nature

For the design problem with variables that are discrete by nature, nine design variables were chosen, these were window type on each aspect, blind type on each aspect and lighting type. The choices available for each variable are shown in Table 10.4.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Options available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window type</td>
<td>Double glazing; air filled, uncoated</td>
</tr>
<tr>
<td></td>
<td>Double glazing; air filled, selective coating, clear</td>
</tr>
<tr>
<td></td>
<td>Double glazing; air filled, selective coating, tinted</td>
</tr>
<tr>
<td></td>
<td>Double glazing; air filled, low E coating</td>
</tr>
<tr>
<td></td>
<td>Double glazing; argon filled, low E coating</td>
</tr>
<tr>
<td></td>
<td>Triple glazing; air filled, uncoated</td>
</tr>
<tr>
<td></td>
<td>Triple glazing; air filled, low E coating</td>
</tr>
<tr>
<td></td>
<td>Triple glazing; argon filled, low E coating</td>
</tr>
<tr>
<td>Blind type</td>
<td>Blind with low reflectivity slats</td>
</tr>
<tr>
<td></td>
<td>Blind with medium reflectivity slats</td>
</tr>
<tr>
<td></td>
<td>Blind with high reflectivity slats</td>
</tr>
<tr>
<td></td>
<td>Shade roll – light translucent</td>
</tr>
<tr>
<td></td>
<td>Shade roll – medium translucent</td>
</tr>
<tr>
<td>Lighting type</td>
<td>T5 with no control</td>
</tr>
<tr>
<td></td>
<td>T5 lighting with linear control</td>
</tr>
<tr>
<td></td>
<td>LED lighting with no control</td>
</tr>
<tr>
<td></td>
<td>LED lighting with linear control</td>
</tr>
</tbody>
</table>

Table 10.4 – Design variables used in the optimisation with variables that were discrete by nature. A different glazing and blind type could be chosen for each aspect, and window blinds could be placed internally or externally.

This gave a design space of nearly 164 million possible designs.

The variables chosen for the experiments in this section were such that representing them as a continuous vector would have been very difficult, as opposed to the variables used in the experiments in Section 10.5.2. For example, for insulation thickness, a variable used in Section 10.5.2, the choices available to a designer may be discrete (only certain thicknesses are available) but the underlying variable is
continuous (any thickness is possible). In contrast it is difficult to imagine how
different types of blinds and their position could be represented as a continuous
variable.

The design variables used in the tests described in Section 10.5.2 were fixed at the
levels shown in Table 10.5.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Fixed at</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glazing ratio</td>
<td></td>
</tr>
<tr>
<td>North</td>
<td>40%</td>
</tr>
<tr>
<td>East</td>
<td>40%</td>
</tr>
<tr>
<td>South</td>
<td>70%</td>
</tr>
<tr>
<td>West</td>
<td>40%</td>
</tr>
<tr>
<td>Orientation</td>
<td>0</td>
</tr>
<tr>
<td>Insulation depth</td>
<td>30cm</td>
</tr>
<tr>
<td>Mass depth</td>
<td>30cm</td>
</tr>
<tr>
<td>Temperature setpoints</td>
<td></td>
</tr>
<tr>
<td>Heating</td>
<td>21.7</td>
</tr>
<tr>
<td>Cooling</td>
<td>26.9</td>
</tr>
<tr>
<td>Natural ventilation</td>
<td>23.7</td>
</tr>
<tr>
<td>Southern overhang depth</td>
<td>1m</td>
</tr>
</tbody>
</table>

Table 10.5 – Variables used as design-variables in the tests on continuous variables
at different resolutions were fixed for the tests on the design problem with variables
that were discrete by nature.

10.6 Results

Throughout the discussion of the results that follows the notation below will be
used:

- $\text{Kriging}_x$ will be taken to mean Krigeing performance after $x$ main-model eval-
uations
- $\text{jEPlus+EA}_y$ will be taken to mean $\text{jEPlus+EA}$ performance after $y$ main-
model evaluations
10.6.1 Highly-discrete optimisation

10.6.1.1 Performance at the stopping point of the Kriging optimisation

Visual assessment of the Pareto front and the spread of designs

The Kriging optimisation was run for 260 main-model evaluations. All estimates of the Pareto front made by both methods after 260 evaluations are shown in Figure 10.1.

The estimates of the Pareto front made by Kriging appear, on the basis of a visual assessment, to be superior to those made by jEPlus+EA after an equivalent number of main-model evaluations. The spread of the estimates made seems approximately equal for the two different methods.

Number of unique Pareto estimates

The Kriging method made, on average, 16 unique Pareto estimates after 260 main-model evaluations. jEPlus+EA made, on average, a similar number (15) for the same number of main-model evaluations.

Quantifying the quality of Pareto estimates using dominated area

Pareto estimates made by the Kriging method after 260 main-model evaluations dominate a larger area than those made by jEPlus+EA after the same number of samples. Kriging estimates dominate 98.39% of the true Pareto dominated area, whereas estimates made by jEPlus+EA dominate 96.56%. This difference in performance is significant at the 95% confidence interval using a Mann-Whitney test.

Summary – performance after 260 main-model evaluations

On the highly-discrete design problem, Kriging performed at least as well as jEPlus+EA on the metrics of number of unique Pareto estimates and spread of Pareto estimates. Assessment of the quality of Pareto estimates (visually and by
Figure 10.1 – All Pareto estimates made by jEPlus+EA and Kriging after 260 main-model evaluations on the highly-discrete design problem. Examination of the estimated Pareto-front for estimates from all 12 optimisation runs on the highly-discrete design problem after 260 main-model evaluations for the whole Pareto front (top) and zooming in on the area with the most designs (bottom) shows that designs found using the Kriging optimisation method are consistently closer to the true Pareto front than those found by jEPlus+EA. Although in some optimisation runs jEPlus+EA produced Pareto fronts comparable to those found by the Kriging method, there were also many points that were less good.
dominated area) indicated that the quality of the Pareto estimates made by Kriging were superior to those made by jEPlus+EA after an equivalent number of main-model evaluations.

10.6.1.2 Finding the equivalence-point for the performance of Kriging and jEPlus+EA

The analysis above indicates that Kriging performs better than jEPlus+EA for a fixed number of main-model evaluations. This Section examines at what point the performance of jEPlus+EA catches up with that of Kriging; the equivalence point. The progression of area dominated by Pareto estimates made by the two methods is shown in Figure 10.2. Taking 260 main-model evaluations as a likely stopping-point for a Kriging optimisation (given the increasing time-cost of running the Kriging model as the number of sample points increases, as discussed in Section 6.1.6), it takes, on average, 600 main-model evaluations to find Pareto estimates of a comparable quality using jEPlus+EA. Therefore, in this instance, Kriging offers a factor 2.3 reduction in the number of main-model evaluations required if looking at dominated area as the most important performance metric. This is comparable to the performance advantage offered on the multi-objective optimisation described in Chapter 8, which also had continuous variables discretised to a resolution of three choices per variable.
Figure 10.2 – For the design problem with 3 design choices for each variable (4 for orientation) Kriging shows an advantage over jEPlus+EA if simulations are very time consuming and the total simulation budget is below 260. The mean performance of jEPlus+EA improves on that of Kriging$^{260}$ after 600 main-model evaluations. This represents approximately a two-fold reduction in required main-model evaluations. Error bars represent one standard deviation above and below the mean.

Performance at the equivalence point on number and spread of Pareto estimates

According to the analysis of area dominated by the Pareto estimates, the quality of jEPlus+EA estimates equals that of Kriging$^{260}$ after around 600 main-model evaluations. However, examining all Pareto estimates made by the two methods at this equivalence point reveals that Kriging is superior at finding good designs in some regions of the solution space, and jEPlus+EA is superior in others (Figure 10.3). The spread of Pareto estimates made by both methods across the Pareto front appears similar$^1$.

$^1$There are designs found by jEPlus+EA in the top left of the solution space that are not amongst the Pareto estimates made by Kriging, and vice versa for the bottom right of the solution space. However, since neither of these regions is on the true Pareto front these can be disregarded. The spread across the true Pareto front appears similar for the two methods.
Figure 10.3 – Examining all Pareto estimates made on the highly-discrete design problem by Kriging\textsubscript{260} and jEPlus+EA\textsubscript{600} (at which point the area dominated by estimates from each method is equivalent) over the whole Pareto front (top Figure) and zooming in on the area with the most designs (bottom Figure) reveals that in some regions of the solution space Kriging tends to make better estimates (top left of the cluster of designs) and in others jEPlus+EA tends to make better estimates (bottom right of the cluster). Neither method shows a clear advantage in terms of the spread of Pareto designs.

The number of Pareto estimates made by jEPlus+EA\textsubscript{600} is superior to that of Kriging\textsubscript{260}, with an average of 25 Pareto estimates compared to 16 for Kriging.
Therefore, at the equivalence point suggested by the dominated area metric, the performance of Kriging and jEPlus appears equal in terms of the spread of Pareto estimates, jEPlus+EA performs better in terms of the number of Pareto estimates and each method shows superior performance to the other in certain regions of the solution space.

10.6.2 Moderately-discrete optimisation

10.6.2.1 Performance at the stopping point of the Kriging optimisation

At the stopping point for the Kriging optimisation of 260 main-model evaluations, the quality of the estimates by both methods in the most crowded region of the Pareto front appears roughly equivalent (Figure 10.4).

![Figure 10.4](image)

**Figure 10.4** – All Pareto estimates made by jEPlus+EA and Kriging after 260 main-model evaluations on the moderately-discrete design problem. In the most crowded region of estimates the quality of Kriging and jEPlus+EA estimates appear roughly equivalent.

However, the presence of outlying Pareto-estimates at either extreme of the Pareto front, found by jEPlus+EA, indicate that this method was less reliable at making good estimates, since good estimates dominate these estimates and would render
them not on the estimated Pareto front. This is borne out by the analysis of mean dominated area after 260 main-model evaluations, on which Kriging performs better, dominating on average an area 1.6% greater than that dominated by jEPlus+EA. This difference in mean dominated area is significant with 99.9% confidence. The spread of the designs from both methods appears similar, and the mean number of Pareto estimates made is also similar, at 17 for Kriging and 18 for jEPlus+EA.

10.6.2.2 Finding the equivalence-point for the performance of Kriging and jEPlus+EA

In terms of the area dominated by Pareto estimates, the performance of jEPlus+EA didn’t quite match Kriging\textsubscript{260} at its stopping point of 600 main-model evaluations (Figure 10.5). It was, however, very close, with the performance of Kriging\textsubscript{260} being easily within one standard-deviation of the mean performance of jEPlus+EA\textsubscript{600} (Figure 10.5).

![Figure 10.5](image)

**Figure 10.5** – For the design problem with six design choices for each variable, Kriging shows a slightly larger improvement than shown in the highly-discrete problem. After 600 main-model evaluations, the performance of jEPlus is very close to equalling that achieved by Kriging after 260 main-model evaluations. Since the true Pareto set is not known in this instance, the methods are compared on the area dominated by the estimated Pareto front, rather than what % of the true Pareto dominated area has been captured by the method.
10.6.2.3 Relative performance of the two methods at their respective stopping points

A visual assessment of the Pareto estimates made by all runs of both methods suggests that jEPlus+EA_{600} is able, in general, to make better Pareto estimates than Kriging_{260} (Figure 10.6).

![Figure 10.6](image)

**Figure 10.6** – At the respective stopping points of each method (260 main-model evaluations for Kriging and 600 for jEPlus+EA) jEPlus+EA appears to make superior Pareto estimates, in that the line of black crosses lies predominantly to the left of the line of red circles.

However, analysis of the mean dominated area shows that Kriging_{260} still dominates a greater area, on average, than does jEPlus+EA_{600} (Figure 10.5). More in-depth analysis reveals that in three of the optimisation runs, jEPlus+EA has completely missed the bottom of the sweep of Pareto designs (hence the lone Pareto-estimates above and to the right of this region). This will have a disproportionate impact on the dominated area compared to other regions of the solution space since missing just a few points here results in missing a relatively large "rectangle" of the design space (see Figure 7.6). Thus, although it is true to say that the area dominated by Kriging_{260} estimates of the Pareto front is, on average, still a little greater than
that dominated by jEPlus+EA_{600} estimates, in many cases jEPlus+EA_{600} actually makes better Pareto estimates.

The spread of the designs seems similar between the two methods, and jEPlus+EA_{600} makes, on average, nearly twice as many Pareto estimates than does Kriging_{260} (32 Pareto-estimates compared to 17).

10.6.3 Moderately-continuous optimisation

10.6.3.1 Performance at the stopping point of the Kriging optimisation

All Pareto estimates made by both methods after 260 main-model evaluations are shown in Figure 10.7, below.

![Figure 10.7](image)

Figure 10.7 – All Pareto estimates made by jEPlus+EA and Kriging after 260 main-model evaluations on the moderately-continuous design problem. Neither method is clearly superior to the other; Kriging makes better estimates in the top left of the solution space, whereas jEPlus+EA makes better estimates further down the steep section of the Pareto front.

Neither method shows a clear advantage in terms of quality of estimates, under a visual assessment. Kriging does better in one region of the solution space whereas
jEPlus+EA does better in another. However, Kriging shows an improvement over jEPlus+EA in terms of area dominated by Pareto estimates, dominating on average 2.4% more area than do estimates made by jEPlus+EA. This difference in mean dominated area is statistically significant, with 99.99% confidence. The spread of designs across the Pareto front looks better for Kriging, with jEPlus not making any Pareto estimates at all between 6.3 and $7 \times 10^4$ kgCO$_2$.

In terms of the number of Pareto estimates made by each method, both Kriging and jEPlus+EA perform similarly, with Kriging making on average 17 Pareto estimates and jEPlus+EA making 15.

**10.6.3.2 Relative performance of the two methods over the course of the optimisation**

The advantage shown by Kriging, in terms of the area dominated by Pareto estimates, after 260 evaluations, existed throughout the optimisation process, and jEPlus+EA did not catch up with Kriging after 600 evaluations. Indeed, it was further from doing so than in the moderately-discrete optimisation problem (Figure 10.8).

**10.6.3.3 Relative performance of the two methods at their respective stopping points**

The estimates of the Pareto front made by jEPlus+EA between 260 and 600 main-model evaluations improved the performance across the front (Figure 10.9), and the advantage in one region observed in Figure 10.7 was more pronounced. However, there was still an advantage shown by Kriging in the low cost/high CO$_2$ region of the solution space, and this advantage means that the area dominated by estimates made by Kriging$^{260}$ was superior to that of those made by jEPlus+EA$^{600}$ (Figure 10.8). Pareto estimates made by Kriging$^{260}$ dominated, on average, 0.95% more area.
For the design-problem with moderately-continuous variables, the performance improvement in dominated area offered by Kriging is increased once again compared to that seen in the highly-discrete and moderately-discrete problems. After 600 main-model evaluations jEPlus+EA has failed to match the performance achieved by Kriging after 260 evaluations, and it is further from doing so than in the moderately-discrete design problem. Error bars represent one standard deviation above and below the mean. This improvement is statistically significant with 99% confidence.

It is difficult to determine, visually, if one method showed an advantage over the other in terms of the spread of Pareto estimates across the solution space; there are regions where jEPlus+EA failed to make estimates, and other regions where Kriging failed to make estimates. The mean number of Pareto estimates made by jEPlus+EA was more than double that of Kriging \( 38 \) compared to \( 17 \).

### 10.6.4 Highly-continuous optimisation

#### 10.6.4.1 Performance at the stopping point of the Kriging optimisation

After 260 main-model evaluations, the Pareto estimates made by Kriging were clearly superior to those made by jEPlus+EA (Figure 10.10). In terms of the area
Figure 10.9 – All Pareto estimates made by jEPlus+EA_{600} and Kriging_{260} on the moderately-continuous design problem. The advantage shown by Kriging in the top left of the Pareto front still exists, although some estimates have now been made by jEPlus+EA in this region. In the section in which jEPlus+EA was observed as outperforming Kriging in Figure 10.7, that advantage has increased further.

dominated by Pareto estimates, this advantage means that estimates made by Kriging dominated, on average, an area 3.7% larger than that dominated by estimates made by jEPlus+EA, and this difference is significant at the 99.99% confidence level.

Figure 10.10 – All Pareto estimates made by jEPlus+EA and Kriging after 260 main-model evaluations on the highly-continuous design problem. Visually, estimates made by Kriging appear superior over most of the solution space.
The spread of Pareto estimates also appears better for Kriging, although the number of Pareto estimates made by both methods was similar (an average of 13). Of all the experiments in this Section, this is the clearest advantage that has been shown for Kriging for a comparable number of main-model evaluations.

### 10.6.4.2 Relative performance in dominated area of the two methods over the course of the optimisation

The advantage shown by Kriging for the same number of main-model evaluations held throughout the optimisation, and even after 600 evaluations jEPlus+EA was still significantly behind Kriging in terms of dominated area (Figure 10.11). The difference in area dominated by estimates from Kriging260 and jEPlus+EA600, at 1.58%, was greater than for all the other tests in this Section, and is significant with 99.9% confidence.

![Figure 10.11](image)

**Figure 10.11** – The performance improvement shown by Kriging is greatest in the case of the highly-continuous design problem. Error bars represent one standard deviation above and below the mean.
10.6.4.3 Relative performance of the two methods at their respective stopping points

Although the analysis by dominated area made in Section 10.6.4.2 indicates that Kriging was able to make superior Pareto estimates to jEPlus+EA, Figure 10.12 tells a more complicated story. Similar to what was seen in Figures 10.7 and 10.9, there are regions of the solution space where Kriging made better estimates, and regions where jEPlus+EA made better estimates.

The spread of estimates seems to be better for Kriging$_{260}$ than for jEPlus+EA$_{600}$, but the number of Pareto estimates made by jEPlus+EA$_{600}$ is superior (24 compared to 13).

10.6.5 Comparing the advantage offered by Kriging on design problems with discretised continuous variables at different resolutions

At a stopping point of 260 main-model evaluations, at all resolutions, Kriging was at least equal on all metrics, and better on at least one. Thus, on this optimisation problem, Kriging was able to make superior Pareto-estimates than jEPlus+EA for the same number of main-model evaluations.

The clearest advantage shown by Kriging, for the same number of main-model evaluations, was in the highly-continuous experiment. After 260 main-model evaluations the estimates made by Kriging were clearly visually superior to those made by jEPlus+EA, both in terms of the dominated area, and the spread of Pareto estimates. The mean number of Pareto estimates from both methods was equivalent.

Deciding how many main-model evaluations were required by jEPlus+EA to match the performance of Kriging after 260 evaluations was less straightforward since the
Figure 10.12 – All Pareto estimates made by jEPlus+EA_{600} and Kriging_{260} on the highly-continuous design problem, over the whole solution space (top) and focussing on the region containing the most Pareto estimates (bottom). The advantage shown by Kriging in the low cost/high CO_{2} region of the solution space, and by jEPlus+EA in the more expensive/lower CO_{2} region that were seen in the moderately-continuous experiment are also present here. It is hard to determine, visually, which method performs best.
performance by the stopping point of the jEPlus+EA optimisations was superior on some metrics and inferior on others.

Although it is difficult to state definitively at what point the performance of jEPlus+EA equaled that of Kriging, taking just the metric of dominated area, there is a clear trend that as the experiment became more continuous, so the advantage offered by Kriging increased (Table 10.6).

<table>
<thead>
<tr>
<th>Number of choices available for each variable</th>
<th>% improvement in dominated area by estimated set of Kriging\textsubscript{260} compared to jEPlus+EA\textsubscript{260}</th>
<th>% improvement in dominated area by estimated set of Kriging\textsubscript{260} compared to jEPlus+EA\textsubscript{600}</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1.9%*</td>
<td>-0.13%†</td>
</tr>
<tr>
<td>6</td>
<td>1.6%***</td>
<td>0.34%</td>
</tr>
<tr>
<td>12</td>
<td>2.4%****</td>
<td>0.95%**</td>
</tr>
<tr>
<td>100</td>
<td>3.7%****</td>
<td>1.58%***</td>
</tr>
</tbody>
</table>

Table 10.6 – As the design problem becomes more continuous (the number of choices per variable increases) so the advantage offered by Kriging increases.  
† jEPlus+EA\textsubscript{600} showed slightly better performance than Kriging\textsubscript{260}.  
\* = significant at the 95\% confidence interval  
** = significant at the 99\% confidence interval  
*** = significant at the 99.9\% confidence interval  
**** = significant at the 99.99\% confidence interval  
If there is no asterisk this indicates that the improvement offered by Kriging is not statistically significant at the 95\% confidence interval.

It seems fair to say that, although it is difficult to define exactly when the performance of one optimisation method equals another, the advantage shown by Kriging over jEPlus+EA increases, on this particular design problem, as the variables become more continuous, and that the best performance for Kriging, relative to jEPlus+EA, was on the highly-continuous optimisation problem.
10.6.6 Optimisation with variables that are discrete by nature

While in all of the continuous optimisation problems (with different resolutions in terms of the number of choices for each variable) Kriging was as good in all metrics, and able to show an advantage in at least one, for the equivalent number of main-model evaluations, the reverse was true of the performance in the optimisation problem with variables that are discrete by nature. Pareto estimates appear consistently worse when assessed visually (Figure 10.13) and this is confirmed by analysis of the area dominated by Pareto estimates made by the different methods, which reveals that, after 260 main-model evaluations, estimates made by Kriging dominate 3.1% less area than those made by jEPlus+EA (although this difference is not statistically significant at the 95% confidence level).

Figure 10.13 – All Pareto estimates made by jEPlus+EA and Kriging after 260 main-model evaluations on the design problem with variables that were discrete by nature. The estimates made by jEPlus+EA appear superior in the upper two clusters of designs, and they look similar in the lower cluster.

This superiority of jEPlus+EA in terms of dominated area is present throughout the optimisation, and by the time jEPlus+EA had reached 600 main-model evaluations (its stopping point), estimates made with it dominated 13.75% more area than
those made by Kriging\textsubscript{260} (Figure 10.14). This improvement in performance from jEPlus+EA\textsubscript{600} over Kriging\textsubscript{260} is significant at the 99.9% confidence interval.

\begin{figure}[h]
\begin{center}
\includegraphics[width=\textwidth]{area_dominated_vs_evaluations.png}
\end{center}
\caption{For the design problem with variables that are discrete by nature there was no advantage to using the Kriging optimisation method compared to jEPlus+EA.}
\end{figure}

The spread of Pareto estimates for the two methods appears similar (Figure 10.13) after 260 evaluations, and at this point the mean number of Pareto estimates was the same, at 11.

That the performance of jEPlus+EA was at least equal on all metrics after 260 evaluations, better on the metric of mean dominated area (although not statistically significantly so), and then proceeded to improve further on the performance of Kriging\textsubscript{260}, means that it can be definitively concluded that jEPlus+EA is superior on this design problem.

### 10.7 Discussion

For the first test-case, in which the underlying variables were continuous and were discretised at various resolutions (3 or 4 choices for each variable, 5 or 6 choices, 12 choices and 100 choices), the advantage showed by Kriging improved as the number
of choices increased. This fits with the hypothesis, stated in Section 10.2, that Kriging should show a greater advantage on design problems in which the variables are continuous. For the highly-discrete problem jEPlus+EA_{600} matched (in fact, slightly improved upon) the average performance of Kriging_{260} in terms of area dominated by Pareto estimates. This advantage shown by Kriging increased such that for the moderately-discrete problem after 600 main-model evaluations the area dominated by Pareto estimates made by jEPlus+EA had not quite matched those made by Kriging_{260}, for the moderately-continuous problem the performance gap was greater still, and for the highly-continuous design problem the improvement shown by Kriging was the greatest yet.

In the design-problem in which the variables were chosen for their underlying discrete nature, Kriging performed worse, in terms of the quality of Pareto estimates (assessed both visually and by mean area dominated by Pareto estimates), than jEPlus+EA after the same number of main-model evaluations. The two methods were equivalent in terms of the spread of Pareto estimates and the total number of estimates made.

This superiority for jEPlus+EA on the design problem with variables that are discrete by nature fits with the hypothesis that the advantage offered by Kriging should be greater in instances in which the variables are continuous. While the performance of jEPlus+EA on the discrete-by-nature test was equal or superior to that of Kriging on all the metrics described in Section 7.7, there was also more variability in the quality of the estimates. Nevertheless, even with the increased uncertainty in the results, after 600 main-model evaluations the average performance of jEPlus+EA exceeded that of Kriging after 260 main-model evaluations to such an extent that even the lower bound of error (one standard deviation below the mean) for the jEPlus+EA_{600} performance exceeded the upper bound of Kriging_{260} performance.
10.7.1 Analysis of the fidelity of the Kriging models on different optimisation problems

This improvement in performance as variables become more continuous may be explained by Kriging being better able to represent design-spaces that are continuous than those that are discrete. If this is the case, it should be possible to see it by analysing the fidelity of the Kriging models. For each of the five design problems in this chapter, all of the designs tested on the main-models were compiled (thousands in each case). For each of the 12 Kriging models built for each design problem a prediction was produced for both construction cost and CO$_2$ emissions for all of these designs, except those that had been used to build that particular Kriging model. These predictions were then compared to the “real” values from the main models, to give a measure of how well the Kriging models represented the main models. These correlations are shown in Figure 10.15.

The worst correlation in Figure 10.15 is for the design problem with variables that are discrete by nature (the two pink figures). For the Kriging model of construction cost (left-hand figure) there appears to be little or no correlation between the Kriging predictions and the main-model values. The Kriging model of CO$_2$ emissions shows slightly more correlation but is still generally poor. This fits with the hypothesis that Kriging is struggling with variables that are discrete by nature and this is what is causing the reduced performance on this optimisation problem. For the design problems with discretised continuous variables at different resolutions the Kriging models of construction cost appear to improve as the design problems become more continuous, but there is not a trend in the Kriging models of CO$_2$ emissions discernible visually. Analysing the data in more detail (Table 10.7) shows that there is in fact an improvement in the fidelity of the Kriging models for both cost and CO$_2$ as the variables become more continuous. This corroborates the theory that these optimisations perform better because Kriging is better able to represent design
Figure 10.15 – Kriging predictions versus main-model predictions for the cost and CO₂ models for all 5 experiments in this chapter. The correlation between Kriging predictions and main-model values for construction costs are shown in the left-hand figures, and for CO₂ emissions in the right-hand figures for the highly-discrete (red), moderately-discrete (green), moderately-continuous (blue), highly-continuous (black) and discrete-by-nature (pink) tests. The straight lines represent a theoretical perfect model, in which the Kriging predictions exactly match the main-model values, the closer the points are to this line the better the Kriging model is at representing the main-model.
problems with continuous variables. These statistical analyses were not performed for the design problem with variables that were discrete by nature, since the solution space covered substantially different cost and CO$_2$ values, rendering the comparison of absolute values in the difference between Kriging predictions and main-model values obsolete. The lack of fidelity between the Kriging models and the main models is clear enough visually, in the case of the discrete-by-nature design problem.

<table>
<thead>
<tr>
<th>Design problem</th>
<th>Mean absolute difference between Kriging and main-model predictions for cost</th>
<th>P value of statistical significance of difference between consecutive cost models</th>
<th>Mean absolute difference between Kriging and main-model predictions for CO$_2$</th>
<th>P value of statistical significance of difference between consecutive CO$_2$ models</th>
</tr>
</thead>
<tbody>
<tr>
<td>highly-discrete</td>
<td>$1.6003 \times 10^5$</td>
<td>$1.0630 \times 10^4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>moderately-discrete</td>
<td>$1.2113 \times 10^5$</td>
<td>$1.0817 \times 10^{-162}$</td>
<td>$7.9890 \times 10^3$</td>
<td>0</td>
</tr>
<tr>
<td>moderately-continuous</td>
<td>$1.1879 \times 10^4$</td>
<td>0.0244</td>
<td>$7.5443 \times 10^4$</td>
<td>$7.0728 \times 10^{-13}$</td>
</tr>
<tr>
<td>highly-continuous</td>
<td>$9.8694 \times 10^4$</td>
<td>$1.0732 \times 10^{-124}$</td>
<td>$4.6544 \times 10^4$</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 10.7 – Mean values for the difference between Kriging predictions and main-model values (the mean fidelity) for the design problems with discretised continuous variables at different resolutions. The statistics shown here are for the same data as presented in Figure 10.15. The P values are the probabilities that the differences in fidelity between models are due to chance (estimated using a T test) for consecutive models (i.e. the P values in the moderately-discrete row are for the difference in mean fidelities between the highly-discrete and the moderately-discrete design problems). The two P values of zero indicate that the P value was lower than the smallest number available in Matlab ($2.22507 \times 10^{-308}$). P values lower than 0.05 indicate statistical significance at the 95% confidence level.

10.7.2 Worst and best-case scenarios for Kriging performance

The advantage shown by Kriging in the highly-continuous design-problem was the greatest for all the experiments performed in the course of this thesis, with the mean
performance of jEPlus+EA after 600 main-model evaluations still being significantly behind that of Kriging after 260. At the other end of the scale of improvements offered by Kriging, no advantage was shown at all on the design problem with variables that were discrete by nature; Kriging performed worse, in terms of the average area of the design space dominated by Pareto estimates, than jEPlus+EA taking the same number of evaluations. This provides a good estimate of a worst-case scenario for Kriging performance; it is the only instance in all the tests described in this thesis where Kriging performs worse than jEPlus+EA for the same number of main-model evaluations. Since the highly-continuous design-problem showed the greatest advantage for Kriging of all the multi-objective optimisations performed in this thesis, and in order to give more information about a best-case scenario, it was decided to extend this experiment in two ways; first of all extending the total number of Kriging sample points to the maximum of 500 recommended in Forrester et al. (2008) and then running jEPlus+EA long enough that it matches this performance for Kriging.

With the design problem with variables that were discrete by nature providing a worst-case scenario for the performance of Kriging on building-optimisation problems, and the extended highly-continuous design problem providing a best-case scenario, informed discussion can now take place as to whether the use of Kriging models in the optimisation process is likely to be advantageous. This discussion will consider the typical characteristics of building optimisation problems and the disadvantages noted in Section 6.2.

10.8 Conclusions

In the tests with underlying design variables that were continuous, but viewed at different resolutions (the number of discrete choices available per variable), Kriging showed an advantage in all cases compared to jEPlus+EA, in terms of the number
of main-model evaluations required to make comparable-quality estimates of the Pareto front. This advantage increased as the number of choices available for each variable increased (as they became more continuous) and was greatest on the design problem with a hundred choices available for each variable.

Conversely, on the design problem with variables that were discrete by nature, Kriging optimisation showed no advantage at all, and in fact tended to make worse estimates of the Pareto front than jEPlus+EA, given the same number of main-model evaluations. The highly-continuous and the discrete-by-nature design problems thus provide the two extremes of performance seen so far of Kriging relative to jEPlus+EA on building-design optimisations.
Chapter 11

Defining the range of Kriging performance on low-carbon building design problems

This chapter extends the experiment on which Kriging performed the best from Chapter 10 in order to give a more detailed picture of performance differences between Kriging and jEPlus+EA. The level of performance advantage offered by Kriging is found to be unlikely to overcome the drawbacks accompanying its use.

11.1 Overview of methods

An overview of the methods and purpose of the experiments performed in this chapter is provided below (Table 11.1).
## 11.2 Introduction

Chapter 10 provided us with a range of estimates for the performance advantage that might be expected from Kriging optimisation, over optimisation using jEPlus+EA. These are included, along with estimates from Chapters 8 and 9, in Table 11.2.

The worst-case scenario for Kriging performance seen in all the tests performed was for the design problem with variables that were discrete by nature, and this is taken as representing the lower end of the range of Kriging performance. This chapter extends the highly-continuous optimisation to the upper limit of the number of sample points recommended for inclusion in a Kriging model of 500 (Forrester et al., 2008), and also runs more generations of jEPlus+EA in order to obtain an estimate of the number of main-model evaluations required for jEPlus+EA to match the performance of an extended Kriging optimisation in terms of mean area dominated by Pareto estimates. These extensions are intended to give a more complete picture of the performance advantage offered by Kriging over jEPlus+EA in a best-case scenario.

<table>
<thead>
<tr>
<th>Experiment type</th>
<th>Methods</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-objective optimisation of a design problem with highly continuous variable choices</td>
<td>An extension of the highly continuous optimisation performed in the Chapter 10 to the recommended maximum number of main-model samples in Kriging. Extension of the same experiment using jEPlus+EA until solutions were approximately equivalent to those found by the Kriging optimisation.</td>
<td>To establish a best-case scenario for the advantage offered by Kriging optimisation on low-carbon building design problems.</td>
</tr>
</tbody>
</table>

Table 11.1 – A summary of the experimental methods used in Chapter 11.
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Experiment</th>
<th>Approximate reduction in main-model samples required for Kriging optimisation compared to stand-alone methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Single-objective optimisation</td>
<td>8-fold reduction*</td>
</tr>
<tr>
<td>8</td>
<td>Multi-objective optimisation</td>
<td>2-fold reduction</td>
</tr>
<tr>
<td>9</td>
<td>Multi-objective optimisation</td>
<td>No significant improvement shown at stopping point of Kriging optimisation</td>
</tr>
<tr>
<td>10</td>
<td>highly-discrete design problem</td>
<td>2.3-fold reduction</td>
</tr>
<tr>
<td></td>
<td>moderately-discrete design problem</td>
<td>Greater reduction than in highly-discrete design problem</td>
</tr>
<tr>
<td></td>
<td>moderately-continuous design problem</td>
<td>Greater reduction than in moderately-discrete design problem</td>
</tr>
<tr>
<td></td>
<td>highly-continuous design problem</td>
<td>Greater reduction than in moderately-continuous design problem</td>
</tr>
<tr>
<td></td>
<td>Design problem with variables that were discrete by nature</td>
<td>Kriging performed worse, for the equivalent number of main-model samples</td>
</tr>
</tbody>
</table>

Table 11.2 – The range of performance of Kriging compared to stand-alone methods on all experiments.

*Although this was the biggest improvement seen in any of the experiments, because it was a single-objective problem it was decided to focus on the other design problems when choosing an experiment to examine in more detail in this chapter.
11.3 Method

The optimisations described in Chapter 10 for the highly-continuous optimisation were simply re-started from their previous stopping points, and continued until, in the case of the Kriging optimisation, 500 main-model evaluations had been made. The jEPlus+EA optimisations were restarted and run for a total of approximately 1,200 generations. This was estimated beforehand to be long enough to match Kriging performance in terms of mean dominated area, and was indeed the case. Had it not been enough, the optimisations would have been restarted and run for even more generations.

11.4 Results

The area of the design space dominated by estimates of the Pareto front continued to increase from the previous stopping point for both the Kriging and jEPlus+EA optimisations. That is to say that the estimates of Pareto-optimal designs moved towards the true Pareto front$^1$; the quality of the Pareto-estimates improved. The progression of the two Pareto-fronts is shown in Figure 11.1.

The advantage shown by Kriging, in terms of the reduction in main-model evaluations, actually increased as the optimisation progressed to its new stopping-point of 500 main-model evaluations. The reduction in main-model evaluations offered by Kriging at different stages of the optimisation process is shown in Figure 11.2. 

$^1$Although the true Pareto front was not known in this experiment, since the design space was too large to allow brute-force analysis.
Figure 11.1 – The mean performance (from 12 runs) of extended runs of Kriging and jEPlus on the highly-continuous optimisation problem. The Kriging optimisation was extended to 500 main-model evaluations, and jEPlus+EA to the number required to match the performance of Kriging.

Figure 11.2 – For the highly-continuous optimisation problem, Kriging enables a reduction in main-model evaluations required to make comparable-quality estimates of the Pareto front. This advantage increases from approximately a 3-fold increase in main-model evaluations required by jEPlus+EA to match the Kriging performance, to approximately a 7-fold increase required to match Kriging performance.
11.5 Discussion

Figure 11.2 gives us a best-case improvement estimate of a 7-fold reduction in main-model evaluations for Kriging compared to jEPlus+EA. However, in addition to having chosen the largest advantage shown in Figure 11.2, this assessment of the performance is generous to Kriging in other ways. First of all, although it takes 3,500 main-model evaluations before the mean performance of jEPlus exceeds that of Kriging in terms of dominated area, it is very close for a long while before that (the difference in the area dominated by the mean Kriging Pareto front and the mean jEPlus+EA Pareto front, for example, is less than 1%). Secondly, the number of estimates of Pareto designs made by the two methods is very different; Kriging optimisations made on average 17 different estimates of Pareto-optimal designs, whereas jEPlus+EA made on average 130 different estimates. More Pareto-estimates, if other performance characteristics are similar, are advantageous since they give a designer more choices and more information about the optimal trade-off between competing objectives. Finally, and perhaps most importantly, although the analysis of the area-dominated by estimates of the Pareto-front made by the two different methods shows the performance of jEPlus+EA not "catching up" with that of Kriging until approximately 3500 main-model evaluations, looking at Pareto estimates made by all 12 optimisation runs from each method earlier in the progression of the jEPlus+EA optimisation tells a more nuanced story (Figure 11.3).

\[2\text{ Which is acceptable, since in this instance a best-case scenario is being established, rather than a robust assessment of average performance.}\]
Figure 11.3 – Examination of estimates of Pareto-optimal designs from Kriging\_500 and jEPlus+EA\_2000 for all 12 optimisation runs indicates that, while the mean dominated area may be higher for Kriging\_500 (as shown in Figure 11.1), jEPlus+EA\_2000 seems to be making superior estimates in the most active region of the Pareto front. The advantage shown by Kriging in terms of dominated area is likely due to superior estimations in one other part of the Pareto-front; the top left end of the dense mass of points.

A cursory examination of Figure 11.3 suggests a clear advantage for jEPlus\_2000 over Kriging\_500, and yet Figure 11.1 indicates that Kriging\_500 has an advantage until jEPlus+EA has made on average 3,500 main-model evaluations, in terms of the area dominated by estimates of the Pareto front. It appears to be that, because Kriging has been more successful at exploring one area of the design space (the section up and left of the black-and-red curved mass of designs in Figure 11.3), this skews the results in terms of dominated area away from what one would expect from a visual examination of Figure 11.3. The seemingly small improvements in the Pareto front made by Kriging in this region of the design space exert more influence over the total dominated area than might be expected because there is a large gap between these points and the maximum CO\textsubscript{2} emissions for all Pareto estimates, which results in a relatively large addition to the dominated area (see Section 7.7.2 for an explanation of how the area dominated by Pareto-optimal designs is calculated and can be skewed by designs at either end of the Pareto front).
In short, in order to give a best-case scenario for the performance advantage offered by Kriging, an equivalence point, in terms of the number of main-model evaluations required to make Pareto estimates of equivalent quality, at which jEPlus+EA is better on all metrics, is chosen.

Given the reasons discussed above for considering that the performance improvement in this case may be considerably smaller than a factor-7 reduction in main-model evaluations, and that the highly-continuous design problem was already chosen as representing an upper value for the advantage offered by Kriging, a factor-7 reduction is perhaps best viewed as an extreme bound on the performance advantage offered by Kriging on building-design optimisations. Based on the results seen in this thesis, instances in which the Kriging method used here offers an even bigger advantage than this can be expected to be rare.

11.6 Conclusions

Kriging was able, in six out of seven of the experiments performed, to make estimates of the Pareto front that were of comparable quality to jEPlus+EA, but with fewer main-model evaluations. However, this advantage was not universal. On one design-problem Kriging showed no reduction in main-model evaluations and instead performed worse, for the same number of model evaluations, than jEPlus+EA. Taking the best and worst performances (relative to jEPlus+EA) of Kriging gives us a worst-case scenario of inferior performance to jEPlus+EA, and a best-case scenario of a factor-7 reduction in main-model evaluations required compared to jEPlus+EA. These two bounds on the expected performance of Kriging is now used to discuss the advantages and disadvantages of applying Kriging to real-world low-carbon design-optimisation problems.
11.7 Kriging and low-carbon building design – a price worth paying?

The previous chapters have explored the level of advantage offered by Kriging on typical building-design optimisation problems, establishing an upper and lower bound to the performance increases likely compared to a high-performance optimisation algorithm working without a surrogate model (jEPlus+EA). The lower bound of Kriging performance was seen on a design problem with variables that were discrete by nature, in which Kriging performed worse, for the same number of main-model evaluations, than jEPlus+EA. The upper bound of Kriging performance, on a highly-continuous design problem, was a factor-7 reduction in the number of main-model evaluations required to make Pareto estimates of comparable quality. However, this factor-7 reduction is likely an exaggeration of the advantage offered by Kriging even in this instance, as explained in Section 11.5.

In the case of design problems with variables that are discrete by nature, the decision of whether or not to use Kriging, based on the results of this thesis, is reasonably simple: it is unlikely to offer a significant advantage, and it may actually be disadvantageous, so it is probably better to use a high-performance optimisation algorithm without a surrogate model.

When deciding whether or not a Kriging surrogate model should be used on design-problems for which the variables are more continuous, even if a best-case scenario of a seven-fold reduction in main-model evaluations is assumed, there are several factors that complicate the decision:

- Kriging has an associated time overhead
- How long does one main-model evaluation take?
- Do the computing resources allow parallel running of main-model evaluations?
The time overhead associated with Kriging can be very significant; over 20 hours for the longest optimisations in this thesis (see Section 6.1.6 for further discussion on this). This means that, even in instances in which Kriging allows a reduction in main-model evaluations, it may still be quicker to optimise using jEPlus+EA without a surrogate model. Whether or not this is the case depends on how long it takes to run main-model evaluations. For optimisations in which the main model is relatively quick to run it is unlikely to be worth using a surrogate model, whereas for optimisations in which one model run takes a very long time the use of a surrogate model may reduce the total time required for the optimisation considerably.

In the absence of parallel-computing resources, Kriging can be expected to offer an advantage in situations in which the variables are continuous (or the underlying nature of which is continuous) and evaluations of the building-energy model are very time consuming.

If the computing resources available to the designer allow the running of building simulations in parallel, the advantage offered by Kriging is reduced still further. While it would certainly be possible to incorporate the running of main-model evaluations in parallel within a Kriging optimisation, it is a matter for further research to establish the best balance between accelerating the optimisation process by running more main-model evaluations in parallel, and improving the accuracy of the Kriging models by running a search for optimum values of $\theta$ after every main-model evaluation. In contrast, the choice of how many main-model evaluations to run in parallel when using a genetic algorithm without a kriging model is more straightforward; if possible all individuals in a population should be run in parallel. For the experiments used in this thesis, with a population size of ten, a ten-fold reduction in time required for the optimisation using jEPlus+EA should be possible through parallelisation with only modest computing resources. More advanced resources, and increasing the population size used in jEPlus+EA, could improve this further. Given this relatively easy-to-achieve improvement through already-established tech-
ntology\textsuperscript{3}, the seven-fold reduction in main-model evaluations offered by Kriging, since it is very much a best-case scenario, and there is a significant time overhead associated with it, seems like a less attractive option for accelerating optimisation.

In Section 7.7, a note of caution was sounded regarding confirmation bias in situations where there were multiple performance metrics on which different algorithms could be assessed. The approach adopted in this chapter is to take a very generous view of the performance of Kriging in a best-case scenario, and to state;

Even if the best-case scenario of a 7-fold reduction in main-model evaluations required were the norm, given the drawbacks of Kriging, it is unlikely to be worth using this implementation of Kriging compared to jEPlus+EA if modest parallel-computing resources are available.

Deliberately biasing the assessment of Kriging’s performance in Kriging’s favour and still not finding an advantage, compared to jEPlus+EA, when parallel-computing resources are available, was an attempt to make conclusions more robust to the risk of confirmation bias.

Another note of caution should be sounded regarding the fidelity of the Kriging models used in this thesis. Since so many interrogations of the Kriging model were made in searching for good designs (2,000 interrogations for each update point), it seems likely that any reduction in performance compared to stand-alone optimisation methods is due to a lack of fidelity between the Kriging models and the main models of objectives. This situation may have been worsened by the choice, as described in Section 7.4.4, to reduce the population size of the GA searching for optimal values of $\theta$, from 100 to 40. Although this reduction had a relatively small effect on the fidelity of the Kriging model in the one case on which it was tested, the effect may have been different on subsequent design-problems. If this is the case then the results presented in this thesis may underestimate the typical performance of Kriging. However, even

\textsuperscript{3}jEPlus+EA is able to operate in parallel already
a population size of 40 is relatively large, and it seems unlikely that increasing the population to 100 would increase the performance advantage significantly above the best-case scenario of a 7-fold reduction in main-model evaluations. Since even that level of reduction is unlikely to make up for the drawbacks of Kriging, if parallel-computing resources are available, the conclusion stated above should hold.

The fidelity of the Kriging models could also have been improved by not only optimising for values of $\theta$ when fitting the model to the data, but also optimising the exponent $p$, rather than leaving it fixed at two. This, however, would have incurred a greater time penalty through the use of Kriging, and also seems unlikely to result in increasing the advantage above the best-case scenario of a 7-fold reduction in main-model evaluations.

Future work on the parallelisation of Kriging optimisations, of both the processes associated with the Kriging model and the running of main-model evaluations, may mean that similar improvements can be achieved for Kriging over non-parallelised optimisations as are currently available for stand-alone methods. If that does transpire, it may be worth revisiting the conclusions of this thesis and re-evaluating the potential advantages offered by Kriging in a building-design optimisation situation.

Currently, the extent to which a single building energy simulation can be run across multiple computing cores is limited (Watson et al., 2013), and so it makes more sense to use parallel computing to simulate more than one model at a time (rather than to accelerate the running of each model). If this situation changes, and building-energy simulations become parallelisable to the extent where all available parallel-computing capacity is in use for running a single simulation, this might create a situation in which Kriging once again provides a useful advantage over a stand-alone genetic algorithm on some building-design problems.
11.8 Kriging: an important role in emerging applications of optimisation in low-carbon building design?

The analysis presented in this chapter concludes that in most instances, given modest parallel-computing resources, it is not worth using the Kriging surrogate modelling techniques investigated in this thesis on typical low-carbon building-design optimisation problems. However, there may be other instances within building-design optimisation when the use of Kriging, or other types of surrogate modelling techniques, could still be highly advantageous.

The method used by Hopfe et al. (2012) to perform a robust-optimisation used a Kriging model to generate 201 permutations around each design, with the variables altered according to pre-assigned probability distributions (rather than just taking a single value, as the variables do for each design in the methods used in this thesis), and the performance chosen to represent that design being the one with the worst performance, thus ensuring that the resulting Pareto-front is robust to variables not performing as well as detailed in the design specification. It may well be the case that, even with parallel-computing resources, the type of robust optimisation performed in the Hopfe et al. (2012) study would not be feasible without the use of a fast-to-interrogate Kriging model. This sort of robust optimisation has great potential to tackle the uncertainties inherent in designing buildings with a long lifespan in a changing climate and with difficult-to-predict occupant behaviour.

Kriging, and other forms of surrogate modelling, may also be of use in constructing models for which there are not currently models available, and for which the construction of a physics-based model is difficult or impractical. For example, in addition to being interested in minimising the cost and environmental impact of a particular building project, designs should fulfil other objectives. Some of these
objectives, such as comfort, or light quality, already have physics-based models and can be relatively easily incorporated into an optimisation approach, but other objectives, for example aesthetic appeal, while in many cases also being extremely important, do not. Building a model for aesthetic appeal, given data on people’s preferences for different types of buildings, should be possible using a Kriging approach since Kriging does not require the understanding of the processes underlying a model (Forrester et al., 2008). Previous attempts to include aesthetic objectives in the optimisation process have been made (see, for example Coley and Schukat (2002)) but have filtered optimisation results for aesthetics at the end of the process, rather than integrating an aesthetic objective as part of a Pareto-based optimisation process. A Kriging approach could be used to create models for numerous other objectives for which they do not currently exist.

Syberfeldt (2009) used a surrogate model in order to screen potential new entrants to a population before they were evaluated on the main models (similar to how cost was used to screen individuals before full evaluation in chapter 14), for optimisation of automobile design problems. This method of using a surrogate as a ‘first check’ on the likely quality of designs, before spending valuable computing resources on a full simulation, may be worth further exploration in a low-carbon building design context.

The Kriging-optimisation method implemented in this thesis is time consuming for several reasons; it requires an EnergyPlus simulation after each optimisation run on the Kriging model and the process of re-calibrating the Kriging model to incorporate the update points is time consuming. Both of these could be avoided by simply building the Kriging model once, from a single (presumably large) sample of the main models, which could be run in parallel. Optimisations could then be run on this Kriging model very quickly, and estimates of Pareto-optimal designs taken directly from the Kriging model, rather than their performance being verified on the main model as is done in the method implemented in this thesis. This of course risks
inaccuracy, since the Kriging model is unlikely to accurately reflect the relationship between variables and objectives in all regions of the design space, but it may be acceptable for some design problems. This is the approach taken in two studies using surrogate models similar to those used in this thesis.

A study by Kang et al. (2013) used a Gaussian-process model with no on-line learning in order to optimise the chilled water temperatures in an HVAC system and make recommendations that can be used in the real-time operation of the building. The provision of such real-time information through optimisation would not be possible if it were run on the main models, since one run of the building-energy model took nearly two hours, but becomes feasible through the use of a surrogate model. The surrogate model was built from 70 samples of the main model, and optimisation using the surrogate took approximately 10 minutes. For the variables under investigation, the Gaussian model was shown to be sufficiently accurate.

This approach of using surrogate models without on-line learning can also be applied in design-based situations similar to those studied in this thesis. Another study by the same team (Kim et al., 2013) used a sample set of 500 main-model evaluations to build a Gaussian-process model of the relationship between window performance parameters\(^4\) on four aspects and the conflicting objectives of minimising energy use and thermal discomfort. This approach of removing the on-line learning that accounts for so much of the time-cost of Kriging optimisation has the potential to give Kriging a clear time advantage over optimisation on main models, but it remains to be seen whether the accuracy of surrogate models is sufficient that the recommendations made by such optimisations are as reliable as those made by optimisations on main models.

\(^4\)U value and solar heat gain coefficient
Part V

Other investigations performed during the thesis
Part IV investigated the main aim of this thesis stated in Chapter 3 – to establish whether the use of Kriging-surrogate optimisation methods can accelerate the optimisation of low-carbon building design problems, compared to existing non-surrogate optimisation methods. This Part explores the subsidiary questions of what degree of improvement can be expected through the use of optimisation methods compared to traditional design methods (Chapter 12), and how low-carbon building design optimisations might be further accelerated through lessons learnt during the course of the main research (Chapters 13 and 14).
Chapter 12

Man versus machine

This chapter describes a comparison of the performance of optimisation methods (both Kriging and jEPlus+EA) against human designers working without optimisation algorithms. The design problem used was the same as that used in Chapter 9. Human designers are shown to perform surprisingly well compared to optimisation algorithms, in some cases finding better designs than those found by Kriging, although never as good as those found by jEPlus+EA.

12.1 Overview of methods

An overview of the methods and purpose of the experiments performed in this chapter is provided below (Table 12.1).

<table>
<thead>
<tr>
<th>Experiment type</th>
<th>Methods</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-objective optimisation of a complex design problem using two different optimisation algorithms and traditional iterative design methods</td>
<td>Optimisation on the same design problem as used in Chapter 9 using Kriging and jEPlus, and comparing these results to those obtained by students optimising the same design problem but without access to optimisation algorithms.</td>
<td>To investigate likely performance improvements offered by optimisation algorithms.</td>
</tr>
</tbody>
</table>

Table 12.1 – A summary of the experimental methods used in Chapter 12.
12.2 Introduction

It is typically assumed that optimisation algorithms offer an improvement over standard design methods, but this assumption is rarely tested. In many instances, when reporting the improvement made by optimisation methods, either a randomly selected design or a base-case starting model is chosen as the comparison (see for example Coley and Schukat (2002), Wetter and Wright (2003) and Ihm and Krarti (2012)). Since designers in the real world are also able to search for the best solution for a given design problem, whether or not they have the assistance of optimisation algorithms, concluding that algorithms offer an advantage without testing against normal practice is misleading.

12.3 Optimisation without optimisation algorithms

In order to provide a reference point for the quality of designs achieved by designers operating on the same optimisation problem but without access to optimisation algorithms, the optimisation problem described in Section 9.2 was posed to students on the design module of the MSc course Advanced Energy and Environmental Studies, at the Centre for Alternative Technology, in mid-Wales. Although most of the students were not practicing professionals in this field, they had a wide range of expertise and most had previously attended a module on computer simulation of building-energy performance. Rather than asking the students to make estimates across the complete Pareto-front, they were asked to search for the designs with the lowest 21-year CO\textsubscript{2} emissions at three different maximum price points – £10k, £25k and £40k\textsuperscript{1}.

The students were given access to a DesignBuilder model of the building, and given instructions on what variables they were allowed to change and how. DesignBuilder

\textsuperscript{1}These costs appear low because they only include the additional costs generated by changing the variables in question, rather than the whole building costs.
was used to calculate the energy demand (by means of an EnergyPlus simulation) based on all the variables except heating system, solar hot water and photovoltaics. These variables were accounted for separately once the heating, lighting and cooling demand for the building had been calculated. They were also given access to cost and 21-year CO₂ models in a spreadsheet that enabled them to quickly calculate the values for the two objective functions using the energy demand (from DesignBuilder) and the chosen variables. These models were intended to ensure that the students had access to exactly the same information, and at the same speed, as the optimisation algorithms had, and thus any difference in performance between the two methods could be attributed to the use of, or absence of, optimisation algorithms.

There were ten students attending the module, and they worked collaboratively. They were given 2.5 days to complete the task, had access to 5 computers (each with all the relevant models on it), and worked approximately 10 hours per day.

### 12.3.1 Simplifying the design-problem

The students quickly established short-cuts which were very effective at simplifying the design problem they faced. The first of these was made after running a series of preliminary tests (approximately 20) in which they tested ideas of which design changes might offer the best reduction in 21-year CO₂ emissions and got a feeling for the price implications of their design choices. The results of these tests were plotted on axes of the two objectives, giving a Pareto front. From these tests they saw that in all the cases they had tested, the narrow building shape performed best, and so they hypothesised that this might always be the case. In order to be confident in this hypothesis, they put together a handful of designs in wide floor plans that they thought might be optimal, reasoning that if even these designs, which they had purposefully designed to work with the advantages of a wide floor plan (principally maximising solar gain and thermal mass to reduce winter heating load), did not
perform as well as the narrow designs then they could safely rule out all designs that were not of a narrow floor plan. This proved to be the case, and they were able to exclude two-thirds of all possible designs from their subsequent investigations (those designs with a square or wide floor plan).

Subsequently they realised that, since the effect of some of the variables was calculated merely by changing a value in a spreadsheet, rather than running a time-consuming EnergyPlus simulation (through DesignBuilder), it should be possible to calculate all possible combinations of these variables for any previously sampled building design. They built a spreadsheet that for any combination of variables 2–9 from Section 9.2.1 (having already ruled out both square and wide floor plans in variable 1) that had already been run in EnergyPlus, all possible combinations of variables 10–12 were calculated. They did the same for lighting, reasoning that, since the lighting load was constant, the effect of different efficiencies of lighting could be calculated arithmetically, and the effect that these different efficiencies of lighting had on the heating and cooling load was likely to be small enough that its exclusion would not affect the accuracy of results too much. These insights, combined with realising that the narrow floor plan always performed better, reduced the size of the design space to be explored from just under one million designs to just over 6,000.

Although 6,000 is still more simulations than could be completed with the computing resources available to the students, it is certainly a substantial reduction from the original value of nearly a million, and considerably easier to search using traditional methods such as trial-and-error and identifying the most influential variables and focussing efforts on finding the optimum value for those variables. The students proceeded from this point largely by these two methods.
12.4 Results

Since the students were asked to produce only an estimate of the best design at three different price points, whereas the optimisation algorithms searched for optimum designs across the Pareto front, it would not be fair to compare results on the basis of area dominated by the Pareto front, as has been done in previous chapters. Instead, from the ten runs of each optimisation method, the 5th ranked design at each price point was chosen to represent the average performance of that algorithm, and compared to the students’ estimates.

The students were able to make good estimates of the optimal designs at each of the three price points. Indeed, their estimates were slightly better, at all three price points, than those estimates made by the 5th ranked estimate found by the Kriging method (Figure 12.1). Since there was only one group of students, all working together, to make their estimates, statistically significant conclusions regarding the performance of human designers without optimisation methods cannot be drawn. However, since in this case the students outperformed the median performance of one optimisation method, it can be concluded that it is incorrect to assume that optimisation methods always offer an advantage over traditional design-methods.

The median estimates made by jEPlus+EA after 600 main-model evaluations were better than both those found by Kriging and by the students (Figure 12.1). They were better still when the runs of jEPlus+EA had reached their stopping criteria of 200 generations (typically between 1,300 and 1,500 unique main-model evaluations).
Figure 12.1 – Estimates from all three different optimisation methods produced similar results. The students were able to do slightly better than the Kriging method at all three price points, and jEPlus+EA was better after 600 main-model evaluations than either method, and better still after 200 generations (typically between 1,300 and 1,500 main-model evaluations). The yellow cloud of designs represents all possible solutions in this portion of the design space. In all but the £10k case, the median performance of jEPlus+EA, after 200 generations, found the true-optimum design. In the £10k case it found the second-best design which was very close in CO₂ emissions to the best design.

The largest percentage improvement between the students’ estimate and jEPlus+EA was at the price-point of £40k, where the true-optimal design (which was found by the median-performing run of jEPlus+EA), had 21-year CO₂ emissions of 4.7 tonnes, and the best design found by the students at this price had 21-year CO₂ emissions of 5.6 tonnes, an increase of approximately 20%. There was a larger absolute difference in CO₂ emissions of the students’ best estimate and the true estimate at the price point of £25k (1.5 tonnes difference) but this was smaller in percentage terms since the total 21-year CO₂ emissions were so much higher.
12.5 Discussion

As described in Section 12.3.1, the design problem chosen didn’t require a new main-model evaluation to establish the effect of every change in variables. The students were able to take advantage of this and dramatically increase the number of designs they could test in the time available to them, effectively reducing the design problem from a daunting ~million possible designs to a more manageable 6,000. Neither the Kriging nor the jEPlus+EA optimisation methods took advantage of this, and were instead searching the full design space of approximately a million possible designs.

It would have been better to do a test on a design space that required main-model evaluations for every design variable whichever optimisation method was chosen. However, while it may have given the students an advantage that they would not have had in a more robust test, it is the case that many optimisation studies in the literature apply complex algorithms and main-model evaluations to search for optimum designs in design spaces where much of the exploration does not require main-model evaluations and could instead be done by testing every possible design (see for example Evins et al. (2012a), Pountney (2012) and Hamdy et al. (2012)).

A method for integrating a brute-force approach on some variables within a search algorithm approach is introduced in Chapter 13.

While the aim when setting up the test was to give human designers, working without optimisation algorithms, exactly the same design problem as was being given to the algorithms, to allow robust comparisons in results to be drawn, this was not completely achieved. It was deemed too complex a problem to ask the students to search for all the designs on the Pareto-front, and so instead they were asked for their estimates of the best designs at three different price points. In contrast, the optimisation algorithms were simply asked to search for Pareto-optimal designs rather than focussing their efforts on specific regions of the Pareto-front. By using constraint-handling, the optimisation algorithms could have been asked to search
in specific price bounds, and this may have further improved the speed with which they found optimal designs at those points. However, a particular advantage of searching the whole Pareto-front is that it provides designers with potentially useful information about the trade-off between objectives that may lead them to choose designs at substantially different performance levels than they might otherwise have done. For example, by examining the solution-set for all possible designs, shown in yellow in Figure 12.1, it is clear there are several "flat" sections of the Pareto front. A designer designing a building to a given budget, without a good representation of the Pareto front, might inadvertently choose a design at the right-hand end of one of these flat sections, when a considerably cheaper design that performed as well would be available to them. The representation of the Pareto front between the chosen price points was much better in the results from the optimisation algorithms than it was in the results from the students.

In addition to reliably finding as-good or better designs at the three price points, jEPlus+EA required much less effort, once the optimisation was set up, than did the traditional, iterative approach taken by the students. The students worked very hard over the three-day period given to them to optimise the building designs, and found the task extremely mentally taxing. While the assertion that optimisation algorithms offer an improvement over traditional design methods in terms of the quality of the designs achieved may not always be true, it may be the case that to tackle such a design problem in a systematic way without optimisation algorithms is so daunting that it is rarely done in practice. In addition the students also had the advantage of more human brains and more computers working on the same problem. A more realistic test might have been to require the students to work alone, although this would certainly have been less popular as an experience for the students! Previous studies that have looked at the improvement achievable using optimisation algorithms compared to traditional design methods have found a much bigger advantage for optimisation methods (Naboni et al., 2013), although
those studies, too, have had their short-falls. Since this was only a single test, more research is required to establish the typical advantage offered by optimisation algorithms over traditional methods.

12.6 Conclusions

The objective of this experiment was to compare the performance of designers, in this case students, without access to optimisation algorithms, to methods using optimisation algorithms. Students were able to make good estimates of optimal designs, superior to those found by one of the optimisation algorithms (Kriging) but not as good as those found by another (jEPlus+EA). This was achieved in part through identifying that not all variables required main-model evaluations in order to establish the effect of changes to just that variable. This enabled students to explore the design space much more extensively than they would have otherwise been able to in the time given to them. It should be noted, however, that although the students were able to make good estimates of optimal designs, doing so required more computing power, more people working on the problem and higher stress-levels! While it may be the case that human designers are able to make estimates of similar quality to optimisation algorithms in some cases, this may be unlikely in a real-world setting due to practical constraints.
Chapter 13

Applying the students’ insights to optimisation algorithms

This chapter develops a methodology to accelerate the optimisation of low-carbon building design using insights from the previous chapter. The results show a significant reduction in main-model evaluations required to make good Pareto estimates and an increase in the number of estimates made.

During the experiments described in Chapter 12, the students, while searching for ways to better explore the design space without the use of search algorithms, were able to dramatically reduce the total number of designs that required an EnergyPlus simulation run in order to determine the performance, and in doing so they were able to make many more design estimations in the time available to them. Effectively they reduced the size of the design space. This chapter introduces and tests a simple method for applying a similar method to that used by the students but as part of an optimisation-based approach.
13.1 Overview of methods

An overview of the methods and purpose of the experiments performed in this chapter is provided below (Table 13.1).

<table>
<thead>
<tr>
<th>Experiment type</th>
<th>Methods</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-objective optimisation using a novel method for handling variables that don’t require a main-model evaluation</td>
<td>Comparing the novel method with the standard Kriging method on the same design problem as used in Chapter 9.</td>
<td>To establish whether the novel method for handling simple variables can accelerate the optimisation process</td>
</tr>
</tbody>
</table>

Table 13.1 – A summary of the experimental methods used in Chapter 13.

13.2 Demand-altering and non-demand-altering variables

The students realised that the effect of changing some variables (building shape, window ratios, window types, construction type, insulation type, insulation thickness, air-tightness, shade depth and lighting type, henceforth called “demand-altering variables”) could only be established by running a full building simulation, while the effect of changing other variables (heating system type, solar hot water system presence or absence, photovoltaic panel presence and size, henceforth called “non-demand-altering variables”) could be established by relatively simple, and very fast, arithmetic calculations for each combination of demand-altering variables already simulated. This is because, for demand-altering variables, the effect of changing the variable affects the building’s energy demands, and depends in part on the values chosen for one or more of the other variables. For example, the choice of window size will affect both the heat gains and heat losses from the building as well as light gains. This means that changing the window size will have an impact on the lighting, heating and cooling demands. The relative size of these effects will depend,
in part, on the insulation, construction type, lighting type, shape and air-tightness variables. On the other hand, for non-demand-altering variables the effect of changing the variable does not alter the building’s energy demands and thus does not require a full simulation in order to establish the effect of a change. For example, for a given heating demand, the efficiency with which this demand is met (the heating system variable) does not change any demand values, only CO$_2$ emissions values$^1$. Knowledge about how the energy/CO$_2$ model for the building is constructed is necessary to identify those variables that are demand-altering and those that are non-demand-altering. With this knowledge identifying those variables which do not require a full simulation in order to establish the effect of a change in that variable is straightforward.

13.3 Current approach to solving design-problems with a mix of demand-altering and non-demand-altering variables

This mix of demand-altering and non-demand-altering variables appears to be relatively common in the literature on building-design optimisation. For example, from a recent conference on building simulation and optimisation, at least three optimisation studies (out of a total of eight that employed optimisation algorithms) included variables that could be considered non-demand-altering: Pountney (2012) optimised a 14-variable design problem that included boiler efficiency, chiller efficiency and photovoltaic (PV) system size as variables. Likewise, the optimisation of a seven-variable design problem by Evins et al. (2012a) included the variables of HVAC system, PV system size and Solar Hot Water (SHW) system while the optim-

$^1$This is true when using the "Simple HVAC" template in EnergyPlus. More complex methods of modelling the heating system exist and their use may mean heating and cooling systems should be viewed as demand-altering variables.
isation of an eight-variable problem by Hamdy et al. (2012) included the variables of two different heating systems (primary and secondary), a micro wind-turbine, PV area and percentage of heating load taken by the primary heating system. It may be that in some instances these variables were indeed included as demand-altering variables, since only one of the studies used EnergyPlus as its simulation engine it is hard to draw comparisons.

In the study that used EnergyPlus, by Evins et al. (2012a), division of the variables into those requiring an EnergyPlus simulation (demand altering) and those not requiring one (non demand altering) seems reasonable: the heating and cooling loads were taken as ideal loads and the effect of different efficiencies of plant added in afterwards, and the solar hot water and PV calculations were done based on the available incident solar radiation on a surface of the EnergyPlus model, and this would not change from model to model for a given location since building geometry was fixed. For the other two studies it seems fair to say that if one were to approach the same design problems using EnergyPlus then treating them as non demand altering would be a reasonable method. Besides this sample of papers from one conference, a study reviewing the implementation of optimisation methods in building design (Attia et al., 2013) listed "Systems" and "Renewables" as (equal) first and fourth most popular variables to optimise, indicating that mixing demand-altering and non-demand-altering variables is common in building-design optimisation.

The approach typically taken to optimise design problems containing this mix of demand-altering and non-demand-altering variables is exactly the same as would be used if all the variables were demand-altering; each variable is treated as if its effect on the objective-function is equally time consuming to calculate and every different combination of variables triggers a new building simulation.
13.4 Proposed method for handling non-demand-altering variables

In some instances, recognising that certain variables are non-demand-altering and do not require a full building simulation in order to establish the effect of changes in those variables alone, means that the design space can be reduced to such an extent that a brute-force approach to optimisation becomes feasible. For example, the study by Evins et al. (2012a) had a total design space size of 4,608 designs$^2$, but to establish the effects of only those variables that are demand-altering would require just 128 EnergyPlus simulations, and all possible combinations of the non-demand-altering variables could be calculated (at minimal computational expense) outside of EnergyPlus in order to determine the performance of all 4,608 designs. A brute-force approach, where feasible, guarantees that the true Pareto-front has been found, rather than just an estimate of it. Given that a brute-force approach is possible in this case with just 128 EnergyPlus simulations, when the optimisation used 400, brute force would be a better approach, being certain of finding all Pareto-optimal solutions with fewer simulations than using the optimisation algorithm.

However, for many design problems, removing those variables that are not demand-altering still leaves a design space that is too large for a brute-force approach. For example, in the case of the Pountney (2012) study, removing those variables identified as potentially non-demand-altering (boiler efficiency, chiller efficiency and PV) still left a design space of over 220,000 possible designs (out of an original total of over 14 million), which is likely to be too large for a brute-force approach unless the simulation run-time is very short and/or the possibility of using a large quantity of parallel computing exists. Likewise, for the study used in Chapter 12, the removal

$^2$Two choices for the shade variable, multiplied by four choices of glazing area, multiplied by four choices of wall U value, multiplied by four choices of window U value, multiplied by three choices of HVAC system, multiplied by four choices of PV, multiplied by three choices of solar hot water to give a total of 4,608 possible unique designs.
of non-demand-altering variables still left a design space of 37,908 possible designs the evaluation of which required a run of EnergyPlus. Although in this case it was possible to evaluate every possible design, since a fast-to-run model was chosen and a powerful, 256-core computer, was available, in "real world" design-optimisation situations this is unlikely to be the case. Could it be possible to take advantage of the fact that many designs do not require an EnergyPlus simulation in order to establish their performance, while still using an optimisation-based approach?

The students participating in the experiment described in Chapter 12 were able to reduce the size of the design space to be searched from 909,792 to 37,908 possible designs by considering only the effects of demand-altering variables when choosing EnergyPlus simulations to perform, and adding in the effect of every possible combination of non-demand-altering variables to each design that had been simulated using EnergyPlus\(^3\). The proposed method applies the same idea to a Kriging optimisation of the design problem, using the optimisation method to choose promising combinations of demand-altering variables to simulate using EnergyPlus, and at the end of the optimisation adding in the effect of every possible combination of non-demand-altering variables.

The effect of some of the non-demand-altering variables depends on the value of another non-demand-altering variable (in this design-problem this was the case for solar hot water, the effect of which – on CO\(_2\) emissions – depends on the efficiency of the water-heating system being replaced by the solar hot water system). In these cases the calculation of non-demand-altering variables must follow a hierarchical order. The proposed method is summarised in Figure 13.1.

\(^3\)They reduced the design space even further by including lighting type as a non-demand-altering variable, surmising that while this wasn’t strictly the case, the effect of lighting type on the heating and cooling demand for the building was small enough that it could safely be disregarded.
Build EnergyPlus model of building and set-up to output demand data for annual heating, cooling and hot water loads

Run building optimisation with gas HVAC and no renewables

For each design produced by step 2, calculate CO₂ emissions and costs for the same design but with different HVAC systems

For every design produced by step 3, calculate the CO₂ and cost of the same design but with solar hot water

For every design produced in steps 3 and 4, calculate the CO₂ and cost of the same design with different sizes of PV system fitted

**Figure 13.1** – Hierarchical calculation of every possible combination of non-demand-altering variables enables the evaluation of many more designs with minimal computational cost.

*This is an example of the approximate total evaluations that might be assessed on the main models using a Kriging optimisation methodology.

The effect of the hierarchical calculation of every possible combination of non-demand-altering variables adds, in this example, an additional 23 design points for every simulation run of EnergyPlus. This has quite a dramatic effect on the appearance of the trade-off between cost and CO₂ emissions (Figure 13.2).
Figure 13.2 – Visualisation of the trade-off between cost and CO₂ emissions before and after the addition of different heating methods and renewables. Since technologies that add cost but reduce CO₂ emissions are being added, the cloud of designs extends below and to the right of the initial designs. Different symbols are used for different heating types (detailed in the legend), larger symbols indicate the use of solar hot water and the three additional colours of blue, green and red indicate the use of 10m², 20m² and 30m² of PV, respectively.

In the example given in Figures 13.1 and 13.2, the new method enables the assessment of the performance of 6,000 building designs for the price of £250⁴. Because this method is able to make many times more design estimations for the same number

⁴This does not take into account the extremely small computational cost of calculating the performance of the additional designs.
of EnergyPlus simulations than the standard Kriging optimisation method in which all variables are assumed to require a full simulation, it might be expected to be able to make better estimates of the Pareto-front with the same budget of EnergyPlus simulations. Whether or not this is the case is examined in the remainder of this chapter.

13.5 Results

Visual comparison of estimated Pareto-fronts after 250 main-model evaluations appears to confirm the hypothesis that treating the demand-altering and non-demand-altering variables separately enables better estimates of the Pareto front to be made for the same computing budget (Figure 13.3).

![Figure 13.3](image)

**Figure 13.3** – Examination of the whole Pareto-front at the end of the optimisation (250 main-model evaluations) indicates that the new method is able to more consistently make good estimates of Pareto-optimal designs - the blue crosses are clustered closer to the true Pareto front than are the red circles.

Zooming in closer to specific sections of the Pareto front shows that, not only is the new method more reliable at finding good designs, but that in many instances the new method has succeeded in finding designs that were not found by any of the
optimisation runs of the old method (Figure 13.4). The spread of Pareto estimates seems at least as good in the new method as it is in the old method.

Figure 13.4 – Zooming in on each section of the Pareto front in turn confirms that the new method is able to make better estimates of the Pareto front.
The results shown in Figures 13.3 and 13.4 are compelling, demonstrating that the new method offers an advantage over the previous method, at least for this design problem. However, they don’t show how the advantage offered by the new method changes over the course of an optimisation run, or how many unique designs are typically found by each method. Figure 13.5 shows that the advantage, in terms of quality of estimates (measured by mean area dominated by Pareto estimates) demonstrated in Figures 13.3 and 13.4 is present throughout the optimisation process, meaning that this method will show an advantage, on this design-problem, even if fewer EnergyPlus simulations are used than have been used in this case. Presumably, if the optimisation were left to run for enough main-model evaluations, the performances of the two methods would converge, since they would eventually both find the true Pareto-set.

The Pareto estimates found by the new method dominate between 4 and 5% more of the true Pareto dominated area than Pareto estimates made by the old method throughout the course of the optimisation. The improvement in the total number of designs on the estimated Pareto-front is perhaps an even more compelling advantage of the new method – there are approximately three times as many designs on the estimated Pareto-front of the new method than on that of the old method. In a real-world design situation this gives the designer more options to choose from when deciding upon the final design. This advantage is also present throughout the optimisation process.

13.6 Discussion

Although the results described in Section 13.5 look very promising – the method is able to estimate a better-quality Pareto front for the same number of EnergyPlus simulations as the standard Kriging optimisation method on this design problem, there are reasons to believe that it may not be robust on all design problems, or
Figure 13.5 – Considering both the mean area dominated and the mean number of designs on the estimated Pareto-front shows a clear advantage for the new method throughout the course of the optimisation.
at least that its performance might be improved even further. This is because, although the method described has the advantage that many more evaluations can be completed for the same computational budget, it has the disadvantage that the optimisation method is searching for a Pareto-front that is not actually the Pareto-front of most interest.

When asking the optimisation method to search for the optimum combination of demand-altering variables, with the non-demand-altering variables fixed at gas HVAC and no solar hot water or PV systems, it will search for the Pareto-front of that particular design problem, when what is actually of interest in is the Pareto front of all 12 variables, of both types. It may be the case that for this design problem the two Pareto fronts are very similar in shape, and adding in different HVAC systems and combinations of PV and solar hot water merely results in similar replications of that Pareto front being transposed in more-expensive and lower-CO\(_2\) steps along the Pareto front. If this is the case the same combinations of demand-altering variables would be on the Pareto front both with and without the additional, non-demand-altering variables added in. But it is not hard to imagine a scenario in which the optimum combination of demand-altering variables depends in part on the combination of non-demand-altering variables, and in this case the method might be less efficient at searching for Pareto-optimal designs.

Examining the results in more detail, it is clear that this is actually the case for this design problem, but it is to a small enough degree that the new method still shows an advantage over the standard one. Figure 13.6, shows that the Pareto-front found in one run for the 9 demand-altering variables does not actually include all the designs that are on the Pareto front for the full design space of 12 variables. There are seven designs that are not Pareto-optimal when considering only nine variables, but which become Pareto-optimal with the addition of the additional HVAC and renewable technologies.
While most of the designs on the estimated Pareto-front for all variables were also in the estimates of Pareto-optimal designs made for demand-altering variables alone, a handful were not, and only became Pareto-optimal after the addition of new heating systems and/or renewables.

While, for this design-problem, the mis-match between the Pareto-front that the optimisation method is searching for and the real Pareto-front of interest is small enough that the new method still shows an advantage over the standard method, this may not be the case for other design spaces. Refinement of this method would give more confidence in its robustness and potentially improve performance on design problems, such as the one tested here, where it already shows an advantage compared to existing methods.

13.7 Future work

One way of improving the method to avoid the mis-match described in Section 13.6 would be to include the non-demand-altering variables as the optimisation progresses, rather than at the end. For some simple non-demand-altering variables, such as PV, this should be possible within the Kriging optimisation method because its effect is typically assumed not to interact with any other variables, so can simply
be accounted for with a single arithmetic formula for each design. For other variables, such as the effect of changing the HVAC or solar hot water variables, integration with Kriging optimisation would be much more problematic. While the optimisation is taking place, the genetic algorithm has access only to Kriging models of CO₂ and cost. In order to integrate the calculation of these variables at each step, it would seem to be necessary to build Kriging surrogate models of each of the building energy demands of interest (in this case heating demand, cooling demand and hot water demand), meaning the optimisation would then be using four surrogate models (one for cost, and one each for heating demand, cooling demand and hot water demand, from which the annual CO₂ could be calculated). This may well be feasible, but adds an additional level of complexity, and will increase the time-cost associated with using a Kriging approach.

It would be much simpler to integrate a method in which the non-demand-altering variables were incorporated at each generation within an optimisation methodology such as jEPlus+EA (or another NSGA-II based method), in which the population on which the algorithm acts is made up of values from the original models, rather than from surrogate models. There are several ways in which this could be done, the most straightforward might be to simply have some code that checked whether a design suggested by the search algorithm required an additional simulation or not, and if it did not the value could be calculated from a library, with the effect of the non-demand-altering variables added in. Another way would be to search for a Pareto-front using just the demand-altering variables but when it came to assessing the performance of each design all possible combinations of non-demand-altering variables could be added in, and the designs chosen to progress to the next generation selected based on their performance compared to the full Pareto front.
Chapter 14

Further exploitation of fast-to-calculate shortcuts

This chapter introduces a novel method for accelerating the optimisation of constrained low-carbon building design problems. The results show a very significant improvement over conventional constraint-handling methods.

The previous chapter introduced a method for taking advantage of variables for which the calculation of their effect on the performance of a building-design is extremely quick, in order to reduce the effective size of the design space. From this point of view, it is also clear that, in addition to some variables being faster to evaluate than others, the same can be said for some objectives compared to others.

The multi-objective optimisation problem used throughout this thesis is between CO$_2$ emissions and construction cost, and yet it is the calculation of CO$_2$ emissions that takes by far the most time, since it requires an EnergyPlus simulation. By contrast, the calculation of construction cost, if it is done outside of EnergyPlus$^1$, can be done extremely quickly, since it is an arithmetic calculation. This disparity between the time required to establish the performance against one objective

---

$^1$There is an option for allowing EnergyPlus to calculate construction costs, and this was used for the optimisations described in Chapters 10 and 11. If this is used it is not possible to calculate cost without running a simulation.
and the time required to establish the performance against another is not currently
exploited in building-design optimisation. This chapter introduces a method for
focussing the search for optimal designs on regions of the Pareto front that are of
particular interest, using computationally-cheap objectives as a filter to do so, with
the objective of improving the quality of the Pareto estimates obtained. This new
method is compared to an alternative method of guiding the optimisation towards
the region of interest using standard constraint-handling methods.

14.1 Overview of methods

An overview of the methods and purpose of the experiments performed in this
chapter is provided below (Table 14.1).

<table>
<thead>
<tr>
<th>Experiment type</th>
<th>Methods</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-objective</td>
<td>Comparing the novel method for</td>
<td>To establish whether</td>
</tr>
<tr>
<td>optimisation</td>
<td>constraint handling with the</td>
<td>the novel method for</td>
</tr>
<tr>
<td>using a novel</td>
<td>standard method, both using NSGA-II as the genetic algorithm. handle constraints</td>
<td></td>
</tr>
<tr>
<td>method for</td>
<td>Using the same design problem as</td>
<td></td>
</tr>
<tr>
<td>handling</td>
<td>used in Chapter 9.</td>
<td></td>
</tr>
<tr>
<td>constraints</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 14.1 – A summary of the experimental methods used in Chapter 14.

14.2 Carbon and cost budgets

Using an optimisation tool to search for the optimal trade off between cost and CO₂
emissions is a common application in building-design optimisation studies, and if
optimisation methods are adopted by designers in the real world this is likely to be
one of their main applications, especially as mandatory CO₂ emissions standards for
new buildings become increasingly stringent. However, while the trade-off between
CO₂ and cost throughout the design space may be of interest to the academic, it
is likely that designers will have CO\textsubscript{2} and/or cost targets in mind from the start of the project. The total design space of interest to the designer might be considerably smaller than the full design space, and searching the full design space may be wasteful. If it were possible to keep the search below a certain cost or CO\textsubscript{2} value (or within an upper and lower boundary), without adversely affecting the efficacy of the search algorithm then it should be possible to improve on the quality of Pareto estimates made by the search algorithm in the area of the design space that is of interest to the designer.

### 14.3 Constraint-handling methods

One method for guiding the optimisation towards the area of interest would be to use a constraint-handling method but apply this to either the cost or CO\textsubscript{2} values, rather than to a stand-alone constraint objective\(^2\). This should have the effect of guiding the search effort towards the area of interest, but may have significant disadvantages in some cases. In the tests of the new method presented later in this chapter, performance is compared to the alternative of using the established constraint-handling mechanism proposed by Deb et al. (2002) in the paper that introduced the NSGA-II algorithm. This method handles designs at the tournament selection stage in the following way, with the objective of guiding the optimisation towards regions of the design space that do not violate constraints. For a given pair of designs, A and B, in the tournament selection:

- if design A meets the constraint criteria and design B does not then design A is chosen, regardless of its performance against other objectives compared to design B;

\(^2\)Constraint handling methods are typically applied to a third objective, such as comfort, for which the designer is only interested in meeting a minimum standard, rather than the optimum trade-off throughout the design space.
• if neither design A nor design B meet the constraint criteria then the design with the smallest overall constraint violation is chosen, regardless of its relative performance against other objectives;

• if designs A and B both meet the constraint criteria then the design to progress to the next population is chosen purely on the standard selection criteria for NSGA-II; dominance rank and crowding distance.

From a theoretical point of view, this method suffers from a particular problem. Because mutation and crossover, the operators which drive the evolution of a population, are blind to constraints, in situations where constraints are very hard to meet they are likely to lead to designs that violate those constraints (Chootinan and Chen, 2006). Because, NSGA-II has a preference for designs that meet constraint criteria over those that do not, in optimisation problems with difficult constraints fewer mutated or crossed-over designs will be selected to progress to the next population than would be the case in the absence of those constraints. In this situation the effect mutation and crossover exert on the populations is reduced compared to an unconstrained situation. This might be expected to slow convergence towards the Pareto-optimal set of designs. Constraint-handling methods have been noted to sometimes cause premature convergence (Mezura-Montes and Coello Coello, 2011), and to require additional simulations to achieve convergence (Hamdy et al., 2011) and this may be one of the causes. The process by which the standard NSGA-II constraint-handling method may lead to a reduction in the rate of evolution in a population is illustrated in Figure 14.1.
Unconstrained NSGA-II

Constrained NSGA-II

Key

1. Individual in unconstrained optimisation or that meets constraint criteria, with a Pareto-rank of 1
2. Individual that does not meet constraint criteria, with a Pareto-rank of 3 and a constraint-violation of +2
3. Individual that has undergone genetic change
4. Winner of tournament decided by crowding distance

Figure 14.1 – The behaviour of a hypothetical population of ten designs, under the influence of crossover and mutation operators, in a constrained and unconstrained optimisation. In the constrained optimisation none of the genetically-altered individuals are present in the final population, whereas for the unconstrained optimisation, subject to exactly the same operations, two of the genetically-altered individuals are present in the final population. This illustrates how the standard constraint-handling method used in NSGA-II may result in a reduction in the influence of mutation and crossover operators.
14.4 Proposed method for handling constraints in computationally-cheap objectives within NSGA-II

The method proposed in this chapter takes advantage of the computationally-cheap nature of the cost calculation in order to check each potential design for whether or not it meets the criteria, and if it does not then to start over again at the previous step of the optimisation algorithm. The aim of the method is to spend EnergyPlus simulations, for which the budget is assumed to be limited, only on designs that meet the cost criteria.

In the proposed method, at each stage in the evolution of the population of individuals in NSGA-II, when individuals are suggested as members of the population they are checked to see whether or not they meet the constraints in the computationally-cheap objective (in this case construction cost). If they do then the second objective (CO$_2$ emissions) is calculated and the individual design takes its place in the population ready for the next stage of the optimisation process. If, on assessing the computationally-cheap objective, it is found that the individual design does not meet the constraints, that design is discarded and a new individual is suggested by the algorithm. These checks take place at each step; population-zero initiation, crossover and mutation.

Under standard population-zero initiation, individual designs are created at random until the pre-set population size is filled, meaning that for a design-problem in which the constraints are difficult to meet many designs in the first population will violate constraints. The performance against design objectives is then calculated. The modification detailed in Figure 14.2 checks the performance against the computationally-cheap objective (in this case construction cost) before calculating computationally expensive objectives.
Randomly assign variables to create new individual design

Does individual meet constraint?

Add individual to P-0

Does P-0 have the required number of individuals?

No

Progress to next stage of optimisation

No

Yes

Yes

Figure 14.2 – The modification of the initiation of the first population (P-0) is relatively simple. Each individual that is created by random assignation of variables is checked against the constraint criteria; if the criteria is not met the process starts again, if it is met then the process proceeds as usual. This modified P-0 initiation aims to ensure that all individuals meet the constraint criteria while maintaining random assignation of variables.

If the design does not meet constraints in the computationally-cheap objective it is discarded and another design chosen at random. In the experiment configuration used in this chapter, if no design meeting the constraint has been found after a hundred attempts then the last-suggested design is accepted into the P-0 population. A hundred attempts was more than long enough in all cases tested in this chapter to ensure that all the designs in the initial population met the constraint, although this will of course depend on the particular constraint characteristics of the design-problem being optimised.

In order to ensure that all members of the post-crossover population meet the constraints, it is not simply enough to discard those that do not meet constraints and subject the pair to the crossover operator again – the crossover operator only results in crossed-over individuals some of the time, and the number of genes that are
subject to change varies. The proposed method, shown in Figure 14.3, ensures that the same number of genes undergo change as was the case in the original crossover (the one that resulted in an individual that did not meet the constraint criteria).

Figure 14.3 – The modifications necessary to ensure the crossover operator produces a population that meets the constraint criteria are more complex than those for the P-0 initiation. The proposed method ensures that the evolutionary pressure is the same as it would be in the absence of the constraint-handling mechanism, but that the new population meets the constraints.

The mutation operator also required careful modification to ensure that the same level of genetic change was exerted in order to create a constraint-meeting individual as had been exerted to create the constraint-violating one.
Figure 14.4 – The modifications to the mutation operator are more complex again than the crossover operator. This new method ensures that the evolutionary pressure is the same as it would be in the absence of the constraint-handling mechanism, and that all individuals in the population meet constraint criteria.

14.5 Experimental method

In order to test this proposed method of handling constraints in NSGA-II, two sets of optimisations were run on the same design problem as used in Chapters 9, 12 and 13, with upper and lower constraints set on the cost objective of £25,000 and £15,000 respectively. The optimisation algorithm used was an open-source Matlab NSGA-II program. This code already had a constraint-handling mechanism that worked as proposed by Deb et al. (2002) and described in Section 14.3, and this was chosen as the comparison against which the proposed method would be com-

---

3Available at http://www.mathworks.com/matlabcentral/fileexchange/31166-ngpm-a-nsga-ii-program-in-matlab-v1-4
pared. For the proposed method, the same code was used, but each of the operators for population-initiation, crossover and mutation were modified to implement the methods proposed in Section 14.4. The population size was left at the default of 50\(^4\) and each optimisation was run for 100 generations\(^5\). All other algorithm settings were left as defaults. Ten runs of each method were completed.

14.6 Results

The method for altering the population-initiation, crossover and mutation operators was successful at ensuring that no constraint-violating individuals existed in any of the populations of the new method.

A visual inspection of the constrained section of the Pareto-front suggests that the new method performs substantially better than the standard method on this design problem (Figure 14.5); there appear to be approximately the same number of Pareto estimates made by each method, and those made by the new method (shown in red crosses) appear, on average, to be closer to the true Pareto front than those made by the standard method (black circles). The spread of Pareto estimates from both methods seems comparable.

\(^4\)Although smaller populations have been shown to work well in modified versions of NSGA-II (Hamdy et al. (2012)), since this version was the standard implementation it was felt to be better to leave the algorithm unchanged from defaults.

\(^5\)The optimisation runs were relatively quick since dynamic simulation results were called from a pre-calculated library of all results, and since no surrogate model was used.
Figure 14.5 – A visual assessment of all estimates made after ten generations in all optimisation runs suggests that the proposed method is superior at making estimates of the Pareto front within the restricted region in this design-problem.

Figure 14.5 shows a snapshot of performance, after ten generations. The progression of both optimisation methods, in terms of the percentage of the true Pareto dominated area that their estimates dominate, is shown in Figure 14.6.

Figure 14.6 – The proposed method shows a clear advantage over the standard NSGSA-II constraint-handling method throughout the course of the optimisation.
Figure 14.6 shows that the proposed method is better at making Pareto-estimates, for all generations\(^6\) tested, in this constrained design space than the standard method of constraint handling in NSGA-II. Indeed, the new method is so much better, on this design problem, than the standard method, that the mean area dominated by the new method is always more than one standard deviation (shown by the error bars) greater than the mean area dominated by the standard method. This superiority is statistically significant at the 99% confidence interval for every generation\(^7\), and significant at a 99.9% confidence interval for generations 7–15.

The new method also finds more designs that are on the estimated Pareto-front (Figure 14.7). This advantage is greatest early on in the optimisation process, when the number of designs on the Pareto front is less than the maximum number of individuals in the population. As the optimisation progresses, the estimated Pareto-set for both methods approaches the whole population, and so the difference between the methods becomes less. In an implementation of NSGA-II using a Pareto archive (as jEPlus+EA does) this advantage might be expected to continue for longer as the population size is allowed to grow.

\(^6\)Previous chapters have used the number of main-model evaluations, rather than the number of generations, as the measure of computational effort. Since this experiment used a different optimisation method it was easier to use number of generations, and this can similarly be thought of as a proxy for computational effort.

\(^7\)Tested using a Mann-Whitney test.
14.7 Discussion

The proposed method was effective at ensuring that no constraint-violating individuals were present in any of the populations during the optimisations on the test design problem. If a design problem is much more constrained than the one used in this test then it might result in constraint-violating individuals being present in some populations. This could be remedied by increasing the number of attempts made by the algorithm to find new individuals that meet the constraint criteria (the method tested here made 100 attempts in each case, although it probably never needed this many).

The proposed method showed superior performance to the standard NSGA-II constraint-handling method on this design problem, but this does not necessarily mean it will show an advantage on all types of design problem in which a computationally-cheap constraint applies. Furthermore, the proposed method was tested against only one other constraint-handling method, the standard one implemented in NSGA-II (that described by Deb et al. (2002)). While it was shown to be superior compared to
the Deb method, other authors have also introduced constraint-handling methods that improve on the Deb method (Chootinan and Chen (2006), Mezura-Montes and Coello Coello (2011)) so it may be that if the new method were compared to more recent constraint-handling methods, the advantage would be reduced. Testing on a wider range of problems, and against a wider range of constraint-handling methods, would certainly be worthwhile.

In addition to being of use in design-problems in which there is a known budget, this method could in fact be extended to any design-problem in which there are constraints in objectives that are computationally cheap. This is, of course, not always the case; popular constraints such as comfort and daylighting (Attia et al., 2013) require a full building-simulation so cannot be handled in this way. However, other popular constraints such as window-to-wall ratio and areas of building surfaces (floor, roof) or volume could be handled in the same way as cost is handled in this example.

The proposed method could also be adapted to work in cases where there is a target performance standard which is computationally expensive to assess (such as a target for CO₂ emissions), rather than a target cost range. In such cases, cost could act as a proxy for the performance standard and ensure that computational effort is not spent on simulating designs that are very unlikely to meet the performance standard. This would involve first doing a quick, scoping optimisation to give the designers an idea of the cost range in which designs meeting the performance criteria are most likely to be found. The decision of the maximum cost to be included in the subsequent, constrained optimisation would be relatively straightforward – setting this at slightly higher than the cost of the cheapest design that met the performance criteria in the scoping optimisation would ensure that at least one design meeting the performance criteria would be found. It would be less straightforward to assign the lower bound of the constrained search, since it is difficult to predict how much a more prolonged and focussed optimisation will improve on the Pareto-front found by the scoping
optimisation. In this instance it would be better to err on the side of caution and set the lower-limit lower than the cost expected by designers to be required to meet performance standards. In this way, designs meeting the performance criteria and showing exceptional value for money are unlikely to be excluded by the constraints. This suggested method is summarised graphically in Figure 14.8.

Figure 14.8 – Generating a proxy target in the computationally-cheap-to-assess objective from a target in the expensive-to-assess objective. In this example a short optimisation run has been done, generating seven Pareto estimates. If there is a target maximum CO₂ emissions of 100 tonnes then the search space can be restricted to designs cheaper than £266,500, since 3 designs that meet the 100-tonne CO₂ target have already been found at this price, and a longer optimisation can be expected to find designs at least as good as this. Such a restriction in the design space should result in better optimisation results using the methods presented in this chapter.

Comparing Figure 14.6 with Figure 9.1 (both of these tests were done on the same design-problem) indicates that the algorithm, and its set-up, being used for the test on the constraint-handling method may not perform as well as jEPlus+EA. Figure 9.1 shows that the area dominated by Pareto estimates made by jEPlus+EA exceeds 90% after approximately 200 main-model evaluations, whereas it takes the proposed method more than 10 generations (500 main-model evaluations) to exceed 90%. Likewise, after 2,000 main-model evaluations jEPlus+EA dominates more than 99% of the true Pareto dominated area, whereas for the new method described in this section on average 97% of the true Pareto area is dominated after a hundred generations (5,000 main-model evaluations). Although this is not a very robust comparison,
since Figure 9.1 shows performance across the whole design space whereas Figure 14.6 focuses on the constrained design space, it does appear that jEPlus+EA has better performance than the original version of NSGA-II implemented here. This fits with what would be expected from improvements made by implementing a Pareto archive, found by Hamdy et al. (2012) (jEPlus+EA implements a similar Pareto archive). Implementing the new constraint-handling method in jEPlus+EA, or another, more modern, implementation of NSGA-II, in order to test whether similar performance improvements apply, would be extremely worthwhile.

The method introduced was implemented on NSGA-II, but it should also be possible to implement a similar method within a Kriging optimisation method.

There was a small time-cost associated with the proposed method, due to the loops for population initiation, crossover and mutation running multiple times. This time cost added, on average, 9 minutes to the optimisation process (the standard method took on average 15 minutes, and the new method took on average 24 minutes).

While nine minutes is very significant compared to the small overall time-cost of the test case, in a real-world design problem in which CO$_2$ performance must be calculated using a dynamic simulation, rather than called from memory, it is likely to be of little consequence.

14.8 Conclusions

The method introduced in this chapter for handling constraints in objectives that are computationally cheap to evaluate shows a considerable improvement in performance over standard constraint-handling methods on the design-problem on which it was tested. This improvement is due to the ability of the method to focus the computational effort only on searching the constrained design space. The method ensures the same level of influence from genetic change as in an unconstrained optim-
isation; all individuals in a population, including those that have undergone genetic change through crossover or mutation, will meet the constraint criteria.

The method was implemented on a freely-available NSGA-II program which lacks some recent improvements to the algorithm. Further testing, on different design problems, against other constraint-handling mechanisms, with more, and less, difficult-to-meet constraints, and with a more up-to-date algorithm, is necessary to establish the level of improvement that can be expected from this new method.
Part VI

Whole thesis conclusions and future work
Context

The design of low-carbon buildings is a complex optimisation problem, with potentially very large numbers of possible designs, epistatic variables whose effect depends on the values of other variables, and the evaluation of each design taking considerable time. Existing methods for searching for the best designs (in the case of a single objective) or Pareto optimal designs (in the case of two or more objectives), while efficient at reducing the number of design evaluations compared to a brute-force approach, can still be prohibitively time consuming. With this in mind, this thesis set out to investigate whether Kriging surrogate models offered a way of accelerating the optimisation process for low-carbon building design.

Conclusions regarding aims and objectives

The main aim of this thesis was to establish the extent to which the use of Kriging-surrogate optimisation methods can accelerate the optimisation of low-carbon building design problems, compared to existing non-surrogate optimisation methods. In respect of this aim, the experiments conducted in the course of this research allow the conclusion that;

- even the best-case performance of Kriging was not good enough to compensate for the current inability to implement main-model evaluations in parallel within Kriging. A jEPlus+EA optimisation implemented in parallel on the same optimisation problem would be expected to make Pareto-estimates of comparable quality and quantity, and to do so significantly faster.
In addition to the best-case performance for Kriging not being enough to outpace a jEPlus+EA optimisation implemented in parallel, Kriging was also observed to suffer from other disadvantages:

- The performance of Kriging was less reliable than that of the jEPlus+EA, especially on design problems with discrete variables (common in building design), on which Kriging performed significantly worse than jEPlus+EA in terms of main-model evaluations required to make good Pareto estimates.

- The time-cost of building, refining and interrogating the Kriging model is substantial and, in cases where the main model is relatively fast to interrogate, means that even if the Kriging approach allows optimisation with fewer main-model evaluations, it may still be more time consuming than jEPlus+EA, even before considering paralellisation.

- The use of the Kriging model imposes an upper limit on the budget for main-model evaluations that is not present with jEPlus+EA, this means jEPlus+EA can be run for longer for particularly difficult optimisation problems.

In addition to the main aim, this thesis had a number of additional objectives. These are re-stated and discussed in turn below:

- to establish which types of low-carbon building optimisation problems Kriging surrogate-optimisation methods are suitable for, if not all problems.

Kriging showed an advantage, in terms of a reduction in main-model evaluations required to find comparable quality designs compared to an efficient stand-alone GA, on low-carbon building optimisation problems in which all variables were continuous. In the absence of parallel computing resources, and for optimisation problems with time-consuming main model evaluations, the research presented here suggests the use of Kriging could enable comparable quality optimisations to be completed more
quickly than using jEPlus+EA. In optimisation problems with discrete variables the research presented here suggests that jEPlus+EA is superior to a Kriging method. If parallel computing resources are available, the results of this thesis suggest that jEPlus+EA is a better choice on all types of low-carbon building optimisation problems.

- To establish the particular advantages and disadvantages of Kriging optimisation methods, when used on low-carbon building optimisation problems.

The advantage of Kriging optimisation methods is a reduction in main-model evaluations required to find good designs, but this advantage is not robust – in the work conducted for this thesis it was only present in design problems with continuous variables. This lack of robustness is the main disadvantage of the Kriging approach. The other disadvantages are that the Kriging approach is difficult to accelerate through the use of parallel computing resources, and that there is a considerable time-cost associated with building, refining and interrogating the Kriging model.

- To establish what degree of improvement in designs can be expected through the use of optimisation methods in general, compared to traditional design methods.

Insufficient research was conducted to make robust conclusions regarding this objective. However, the one experiment conducted showed that human designers working without optimisation methods are not always out-performed by optimisation methods (in some cases they performed better than the Kriging method, but they never out-performed jEPlus+EA), although the optimisation methods allow a much better picture of the trade-off between competing objectives.

- To develop other methods to further accelerate or improve the optimisation of low-carbon building design, based on observations and lessons learnt throughout the research.
Two new methods were developed in the course of this thesis. The first new method categorises variables according to whether they altered the energy demand of the building or not, and thus whether or not they require evaluation by a dynamic simulation. The implementation of this method enabled comparable designs to be found with fewer dynamic simulations than existing methods which do not categorise variables in this way. Further development is required to make this method more robust. The second new method applied a similar approach to objectives, categorising them according to how quickly the performance of a model against the objective could be calculated, and using the fast-to-interrogate objectives to filter out designs that do not meet constraints in that objective. This method showed significant improvement compared to traditional constraint-handling methods in terms of a reduction in dynamic simulations required to find comparable quality designs and in terms of the total number of unique Pareto estimates made.

**Future work**

Several areas of important work could be investigated as an extension to the work undertaken in this thesis. The first, and perhaps most important of these, is to develop a robust methodology for using Kriging optimisation with parallel-computing resources.

It may be worth trying to improve the fidelity of the Kriging model on building design problems through both increasing the population size of the genetic algorithms searching for optimal values of $\theta$, and through optimising the values of the exponent $p$ in order to fit the data more faithfully. Since the problems with the Kriging method demonstrated in this thesis were to do with the fidelity with which it was able to represent the main model (especially in the case of models with many discrete variables), this change may improve the performance of Kriging models quite significantly.
The method used in this thesis builds a Kriging model for both the CO₂ emissions and the construction cost of the building designs. Currently approximately half the additional time-penalty associated with using the Kriging method is time spent building, refining and interrogating the Kriging model for cost. Since the main model for construction cost is much faster to interrogate than that for CO₂ emissions, a fraction of a second in the case of the experiments presented in this thesis, this should be an area where the Kriging method could be improved. However, if the algorithm is going to use multi-EI as its search term, replacing the Kriging model for cost with the "real" model is not entirely straightforward because a value for the error in the model is required. Since, in the case of the Kriging model, this error is the estimated error between the Kriging model and the main model, it may be a simple matter of fixing the error to zero or a very low number. Preliminary tests were done using such a method, but while they reduced the time penalty associated with Kriging, the results were inferior to those found using Kriging. Further investigation is required establish why this was the case.

A better method of using the main cost model but a surrogate model for CO₂ emissions might be to use a multi-objective algorithm to search for the optimal trade-off between cost and CO₂ emissions, with a Kriging model providing a prediction of CO₂ emissions and the cost model being interrogated directly for estimates of costs. This would also have the advantage that a more advanced algorithm, such as NSGA-II, could be used with the Kriging methodology. It may be that using this method improves the performance of Kriging optimisation, and reduces the additional computational-cost associated with it, to such an extent that it becomes more competitive compared to stand-alone algorithms.

While the main conclusions of this thesis suggest that Kriging is not a good method for accelerating the process of low-carbon building design optimisation, that is not to say that there are not very promising applications for it within low-carbon building design optimisation. Kriging could be used to create models for objectives that are
traditionally not modelled at all, such as the aesthetic qualities of a building. This
would enable multi-objective optimisations with aesthetic criteria included as an
objective or constraint. Kriging may also have important applications in optimising
buildings to be robust to uncertainties such as climate change or occupant behaviour.

The two new methods introduced in this thesis to accelerate optimisations by taking
advantage of fast-to-calculate shortcuts require further testing and refinement to
establish the typical improvement offered by them, and their robustness over a
range of optimisation problems. Both of these methods showed very promising
improvements over standard methods in the tests performed in this thesis.
Appendix A

Building models used for simulation in EnergyPlus

A.1 Building model used for experiments in Chapter 8

The building model used for experiments in Chapter 8 was a two-storey domestic property, shown in Figure A.1. The model was deliberately kept relatively simple in order to allow comprehensive evaluation of all possible designs.
Figure A.1 – The building model was a two-storey, terraced, residential property with a square floor plan and with south and north facing exterior walls. The maroon coloured blocks are adiabatic blocks, used to represent buildings to either side of the model in a terrace.

External walls, roofs and floors in the building model contained design variables whose thickness was to be optimised by the different methods under test (Figure A.2).
Figure A.2 – The construction elements containing design variables used in the building model shown in Figure A.1. Construction variables do not have a thickness marked since this varied according to the variable assignation.

Other changes were made to the model from default settings, including setting the internal flooring as 19mm thickness timber flooring and the internal partitions as lightweight gypsum plasterboard partitions. The air-change rate was set to a constant of 0.5 air changes per hour and scheduled ventilation was chosen to reduce model run times. Activity settings, lighting settings and temperature setpoints were all left as the default values for the relevant activities for different zones of the building.

A.2 Building models used for experiments in Chapters 9, 12, 13 and 14

Three different shapes of building model were used in the experiments described in Chapters 9, 12, 13 and 14, since building shape was one of the design variables to be optimised. The three different shapes were a square floor plan (Figure A.3), a narrow floor plan (Figure A.4) and a wide floor plan (Figure A.5).
Figure A.3 – The square version of the building model used in Chapters 9, 12, 13 and 14. The arrow shows north, and the maroon coloured blocks are adiabatic blocks, used to represent buildings to either side of the model in a terrace.
Figure A.4 – The narrow version of the building model used in Chapters 9, 12, 13 and 14. The maroon coloured blocks are adiabatic blocks, used to represent buildings to either side of the model in a terrace.
There were three wall constructions available as design variable – heavyweight, lightweight and straw-bale. The heavyweight construction was the same as the wall construction shown in Figure A.2, except that two options for insulation type (expanded polystyrene or mineral wool) were available and three different insulation thicknesses (5, 15 and 25cm). The lightweight and straw-bale constructions are shown in Figure A.6.
Figure A.6 – Lightweight and straw-bale external wall construction types, as used in the experiments in Chapters 9, 12, 13 and 14. The insulation layer in the lightweight wall could be either expanded polystyrene or mineral wool, and had three different choices for thickness. The straw-bale wall specification was fixed as shown.

External roof and floor constructions were not varied in this optimisation. The constructions used are shown in Figure A.7.
Figure A.7 – The external floor and roof constructions used in the building models in Chapters 9, 12, 13 and 14 were fixed as shown.

Internal flooring was set as 19mm timber flooring and internal partitions as lightweight gypsum plasterboard partitions. Activity settings, lighting settings and temperature setpoints were all left as the default values for the relevant activities for different zones of the building.

A.3 Building model used for experiments in Chapters 10 and 11

The optimisations in Chapters 10 and 11 optimised the design of an office building. The form of the building model is shown in Figure A.8.
Figure A.8 – The building model for the office optimisation used in Chapters 10 and 11 was a simple rectangular shape with two storeys. The ground and first floor layouts were identical.

The competition requirements included neighbouring buildings surrounding the building to be optimised. This requirement was not included in the optimisations, in order to reduce simulation run-times.

The wall constructions were the same as used in Chapter 8 (shown in Figure A.2), with the thermal mass (heavyweight concrete block) and expanded polystyrene insulation layer thicknesses chosen as design variables. The roof construction was also the same as used in Figure A.2, with the insulation layer thickness chosen as a design variable. The ground floor construction was as shown in Figure A.2, but with expanded polystyrene instead of urea formaldehyde foam as the insulation. The thermal
mass (cast concrete) and insulation (expanded polystyrene) layer thicknesses were chosen as design variables. Internal flooring was 19mm wooden flooring, and internal walls were 100mm dense cast concrete walls. Apart from the design variables, all settings were as proscribed in the competition brief, apart from ventilation and infiltration. Ventilation was set as natural ventilation with a constant rate of three air changes per hour, and infiltration was calculated using a DesignBuilder crack template of “good”.
Appendix B

Cost and CO₂ models used throughout the thesis

Since the objectives of this thesis were to compare the efficacy of different optimisation methods on low-carbon building design problems, rather than to use optimisation methods to answer questions about low-carbon building design, the specification of the models used to calculate performance against objectives (cost and CO₂) has not been described in detail in the main parts of the thesis. That the models are accurate is less important than if they were being used to answer specific questions about optimum designs for different scenarios, so long as they are accurate enough to maintain the characteristics of real-world, low-carbon building design problems.

The models used to calculate the two objectives (cost and CO₂) are detailed in this Appendix. Consequently, where possible, costs and embodied CO₂ data has been taken from industry sources, but in some cases estimates were made owing to lack of information. These instances are noted in the relevant sections.
B.1 Calculating annual CO\textsubscript{2} emissions from gas and electricity use

Demand for heating, cooling and lighting for different designs was calculated through EnergyPlus simulations. In order to calculate the annual emissions of CO\textsubscript{2} from the energy used to meet the demands of the building models, the following formulas were used.

\[ CO_{2}^{\text{heating}} = \left( \frac{\text{Demand}_{\text{heating}}}{\text{Efficiency}_{\text{heating}}} \right) \times CO_{2}^{\text{Factor}_{\text{gas/electricity}}} \] \hspace{1cm} (B.1)

\[ CO_{2}^{\text{cooling}} = \left( \frac{\text{Demand}_{\text{cooling}}}{\text{Efficiency}_{\text{cooling}}} \right) \times CO_{2}^{\text{Factor}_{\text{electricity}}} \] \hspace{1cm} (B.2)

\[ CO_{2}^{\text{lighting}} = \text{electricity}_{\text{lighting}} \times CO_{2}^{\text{Factor}_{\text{electricity}}} \] \hspace{1cm} (B.3)

The total annual CO\textsubscript{2} emissions from gas use, if gas was used as the heating fuel, were equal to the CO\textsubscript{2} emissions from heating as calculated in Equation B.1 (gas used for cooking was not taken into consideration). CO2 emissions from electricity, if gas was used for heating, were calculated as follows:

\[ CO_{2}^{\text{electricity}} = CO_{2}^{\text{cooling}} + CO_{2}^{\text{lighting}} \] \hspace{1cm} (B.4)

and for designs in which electricity was also used to supply heat:

\[ CO_{2}^{\text{electricity}} = CO_{2}^{\text{heating}} + CO_{2}^{\text{cooling}} + CO_{2}^{\text{lighting}} \] \hspace{1cm} (B.5)
The total CO2 emissions arising from energy use in the buildings was calculated as follows:

\[ CO_{\text{annual}} = CO_{\text{gas}} + CO_{\text{electricity}} \] (B.6)

Since different experiments were conducted at different times, and the estimated average CO2 factors issued by the relevant bodies in the UK changed over the course of the thesis, different CO2 factors for both electricity and gas were used in different experiments.

### B.1.1 CO2 factors used for the experiments described in Chapter 8

The following CO2 emissions factors for gas and electricity were taken from the DEFRA guidance for company reporting at the time the experiments were started (DEFRA, 2009).

\[ CO_2 \text{Factor}_{\text{electricity}} = 0.42 \text{kg kWh}^{-1} \] (B.7)

\[ CO_2 \text{Factor}_{\text{gas}} = 0.19 \text{kg kWh}^{-1} \] (B.8)

### B.1.2 CO2 factors used for the experiments described in Chapters 9, 12, 13 and 14

For the experiment in Chapters 9, 12, 13 and 14, in which the CO2 objective was total emissions over a 21-year period (in-use and embodied), the CO2 factor of electricity was assumed to vary as shown in Figure B.1.
Figure B.1 – The CO₂ emissions per unit of UK grid electricity were assumed to decline as shown. Adapted from predictions by DECC

The in-use emissions from electricity were calculated by multiplying the annual electricity usage by the relevant CO₂ factor for each of the 21 years, as follows:

\[
\sum_{i=1}^{21} CO₂_{electricity_i}
\]  

(B.9)

For the same experiment, the CO₂ factor of gas was assumed to stay constant. This constant was set at:

\[
CO₂ Factor_{gas} = 0.1832 kg kWh^{-1}
\]  

(B.10)

taken from the DECC estimate at the time the experiment was conducted. The 21 year emissions from gas use in the building were simply the annual emissions, using this CO₂ factor, multiplied by 21.
<table>
<thead>
<tr>
<th>HVAC type</th>
<th>Coefficient of performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gas boiler</td>
<td>0.9</td>
</tr>
<tr>
<td>Air-source heat pump (heating and cooling)</td>
<td>2.5</td>
</tr>
<tr>
<td>Ground-source heat pump (heating and cooling)</td>
<td>3.5</td>
</tr>
<tr>
<td>Air-conditioning (for use with gas boiler)</td>
<td>1.67</td>
</tr>
</tbody>
</table>

Table B.1 – Coefficients of performance for different HVAC plant used for experiments.

B.1.3 CO₂ factors used for the experiments described in Chapters 10 and 11

The experiments described in Chapters 10 and 11 were built last of all, and the CO₂ emissions recommended by DECC were higher than in the other models, they were, for electricity:

\[
CO_2 Factor_{electricity} = 0.685 kg/kWh
\]  

(B.11)

and for gas:

\[
CO_2 Factor_{gas} = 0.195 kg/kWh
\]  

(B.12)

B.1.4 HVAC efficiency

The following coefficients of performance were assumed for different HVAC plant throughout the thesis (Table B.1). These were taken from the CO₂ accounting software CarbonMixer.


\[\text{Available at www.carbonmixer.com}\]
B.2 Calculating embodied CO₂ emissions

The experiments described in Chapters 9, 12, 13 and 14 used minimisation of the total CO₂ emissions over a 21-year period as an objective. The contribution of in-use emissions to this 21-year total was calculated as described in Section B.1.2. The embodied CO₂ emissions, defined as emissions arising from the manufacturing of the materials used to construct the buildings, were calculated as shown in Table B.2 for each of the materials that could change from one design to the next (materials that would have the same quantity in all models were not counted). Two sources for the figures for embodied CO₂ emissions were used; the ICE database (Hammond and Jones, 2011) and the database supplied with the building-simulation software DesignBuilder².

²DesignBuilder Software Ltd. Available online at http://www.designbuilder.co.uk
<table>
<thead>
<tr>
<th>Construction item</th>
<th>Embodied CO₂ (kg/m²)</th>
<th>Source of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double glazing air fill</td>
<td>83</td>
<td>ICE 2.0</td>
</tr>
<tr>
<td>Double glazing argon fill</td>
<td>83</td>
<td>ICE 2.0</td>
</tr>
<tr>
<td>Triple glazing argon fill</td>
<td>118</td>
<td>Estimate based on ICE 2.0 figure for double glazing</td>
</tr>
<tr>
<td>Bricks (for all lightweight and heavyweight wall constructions)</td>
<td>39.27</td>
<td>DesignBuilder database</td>
</tr>
<tr>
<td>Dense concrete, 10cm (for all heavyweight wall constructions)</td>
<td>18.4</td>
<td>DesignBuilder database</td>
</tr>
<tr>
<td>EPS insulation one cm thickness</td>
<td>1.1</td>
<td>ICE 2.0</td>
</tr>
<tr>
<td>Mineral wool one cm thickness</td>
<td>1.2</td>
<td>ICE 2.0</td>
</tr>
<tr>
<td>Straw bale wall (complete but without plaster)</td>
<td>0.115</td>
<td>DesignBuilder database</td>
</tr>
<tr>
<td>Gypsum plaster (all lightweight and heavyweight wall constructions)</td>
<td>4.94</td>
<td>DesignBuilder database</td>
</tr>
<tr>
<td>Sand/lime plaster (for straw-bale walls)</td>
<td>0.16</td>
<td>DesignBuilder database</td>
</tr>
<tr>
<td>Photovoltaic modules</td>
<td>242</td>
<td>ICE 2.0</td>
</tr>
<tr>
<td>Solar hot water modules</td>
<td>150</td>
<td>Estimate - value chosen between triple glazing and PV</td>
</tr>
<tr>
<td>Plywood decking (for shading)</td>
<td>0.07</td>
<td>ICE 2.0</td>
</tr>
</tbody>
</table>

Table B.2 – CO₂ emissions embodied in various materials the quantities of which varied in different building designs

Embodied CO₂ emissions were calculated by multiplying the quantity of each of the materials used by their associated embodied CO₂ emissions from Table B.2, and summing these CO₂ emissions. Total CO₂ emissions over the 21 year period were calculated by adding the in-use emissions (Equations B.9 and B.10) to the embodied emissions.

B.3 Cost models used in Chapter 8

The cost model used in the first experiments was simplistic and excluded some costs because of missing data. The cost estimates were obtained from a commercial
window supplier (Glendinning, 2009) and an architects price book (Langdon, 2007). These are shown in Table B.3.

<table>
<thead>
<tr>
<th>Construction item</th>
<th>Cost £/m²</th>
<th>Source of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double glazing air fill</td>
<td>80</td>
<td>Commercial window supplier</td>
</tr>
<tr>
<td>Double glazing argon</td>
<td>86</td>
<td>Commercial window supplier</td>
</tr>
<tr>
<td></td>
<td>fill</td>
<td></td>
</tr>
<tr>
<td>Triple glazing air fill</td>
<td>185</td>
<td>Commercial window supplier</td>
</tr>
<tr>
<td>Dense concrete 10cm thickness</td>
<td>6.4</td>
<td>Architects’ price book</td>
</tr>
<tr>
<td>Bricks</td>
<td>17.08</td>
<td>Architects’ price book</td>
</tr>
<tr>
<td>Doors</td>
<td>Costs not included in model</td>
<td></td>
</tr>
<tr>
<td>Shading</td>
<td>Costs not included in model</td>
<td></td>
</tr>
</tbody>
</table>

Table B.3 – Costs for different construction elements used in the experiments described in Chapter 8.

B.4 Cost models used in Chapters 9, 12, 13 and 14

The costs used to calculate the cost objective in the experiments described in Chapters 9, 12, 13 and 14 are detailed in Table B.4. These cost estimates were obtained from the CO₂ accounting software CarbonMixer, information from architects (White, 2012) and the cost modelling software Costmodeller³.

³Available at http://www.amtech.co.uk/cost-modeller
<table>
<thead>
<tr>
<th>Construction item</th>
<th>Cost £/m²</th>
<th>Source of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double glazing air fill</td>
<td>100</td>
<td>Carbon Mixer version 2</td>
</tr>
<tr>
<td>Double glazing argon fill</td>
<td>115</td>
<td>Carbon Mixer version 2</td>
</tr>
<tr>
<td>Triple glazing argon fill</td>
<td>135</td>
<td>Carbon Mixer version 2</td>
</tr>
<tr>
<td>Bricks (for all lightweight and heavyweight wall constructions)</td>
<td>17.08</td>
<td>Architects’ price book</td>
</tr>
<tr>
<td>Dense concrete, 10cm (for all heavyweight wall constructions)</td>
<td>6.4</td>
<td>Architects’ price book</td>
</tr>
<tr>
<td>EPS insulation one cm thickness</td>
<td>1.282</td>
<td>Costmodeller</td>
</tr>
<tr>
<td>Mineral wool one cm thickness</td>
<td>0.48</td>
<td>Costmodeller</td>
</tr>
<tr>
<td>Straw bale wall (complete but without plaster)</td>
<td>140</td>
<td>Information from architect</td>
</tr>
<tr>
<td>Gypsum plaster (all lightweight and heavyweight wall constructions)</td>
<td>11.23</td>
<td>Costmodeller, board finish plaster</td>
</tr>
<tr>
<td>Sand/lime plaster (for straw-bale walls)</td>
<td></td>
<td>Included in cost of straw-bale wall</td>
</tr>
<tr>
<td>Photovoltaic modules</td>
<td>640</td>
<td>Carbon Mixer version 2</td>
</tr>
<tr>
<td>Solar hot water modules</td>
<td>600</td>
<td>Carbon Mixer version 2</td>
</tr>
<tr>
<td>Plywood decking (for shading)</td>
<td>34.18</td>
<td>Costmodeller</td>
</tr>
<tr>
<td>Airtightness measures</td>
<td>2,000</td>
<td>Estimate</td>
</tr>
<tr>
<td>Condensing boiler</td>
<td>870</td>
<td>Carbon Mixer version 2</td>
</tr>
<tr>
<td>Air-source heat pump</td>
<td>4,500</td>
<td>Carbon Mixer version 2</td>
</tr>
<tr>
<td>Ground-source heat pump</td>
<td>15,000</td>
<td>Carbon Mixer version 2</td>
</tr>
</tbody>
</table>

Table B.4 – Costs for the various materials used in the experiments in used in Chapters 9, 12, 13 and 14.
The costs used in Chapters 10 and 11 were taken from the specification for an optimisation competition run by DesignBuilder. They are detailed in Table B.5.

<table>
<thead>
<tr>
<th>Construction item</th>
<th>Cost per unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roof insulation (mineral wool batt)</td>
<td>£40/m³</td>
</tr>
<tr>
<td>Wall insulation (expanded polystyrene)</td>
<td>£100/m³</td>
</tr>
<tr>
<td>Thermal mass (dense concrete)</td>
<td>£4,000/m³</td>
</tr>
<tr>
<td>Southern window overhangs (up to 0.5m depth)</td>
<td>£540/m²</td>
</tr>
<tr>
<td>Southern window overhangs (between 0.5 and 0.75m depth)</td>
<td>£430/m²</td>
</tr>
<tr>
<td>Southern window overhangs (between 0.75 and 1m depth)</td>
<td>£410/m²</td>
</tr>
<tr>
<td>Blinds</td>
<td></td>
</tr>
<tr>
<td>Low reflectivity slats 25mm</td>
<td>£50/m² glazing</td>
</tr>
<tr>
<td>Medium reflectivity slats 25mm</td>
<td>£60/m² glazing</td>
</tr>
<tr>
<td>High reflectivity slats 25mm</td>
<td>£70/m² glazing</td>
</tr>
<tr>
<td>Shade roll</td>
<td></td>
</tr>
<tr>
<td>Light translucent</td>
<td>£40/m² glazing</td>
</tr>
<tr>
<td>Medium translucent</td>
<td>£50/m² glazing</td>
</tr>
<tr>
<td>Double glazing</td>
<td></td>
</tr>
<tr>
<td>Air filled, uncoated</td>
<td>£400/m²</td>
</tr>
<tr>
<td>Air filled, selective coating, clear</td>
<td>£420/m²</td>
</tr>
<tr>
<td>Air filled, selective, tinted</td>
<td>£430/m²</td>
</tr>
<tr>
<td>Air filled, low E coated</td>
<td>£430/m²</td>
</tr>
<tr>
<td>Argon filled, low E coated</td>
<td>£460/m²</td>
</tr>
<tr>
<td>Triple glazing</td>
<td></td>
</tr>
<tr>
<td>Air filled, uncoated</td>
<td>£500/m²</td>
</tr>
<tr>
<td>Air filled, low E coated</td>
<td>£530/m²</td>
</tr>
<tr>
<td>Argon filled, low E coated</td>
<td>£560/m²</td>
</tr>
<tr>
<td>Compact fluorescent lighting with no control</td>
<td>£55/m² floor area covered</td>
</tr>
<tr>
<td>T5 lighting</td>
<td></td>
</tr>
<tr>
<td>No control</td>
<td>£60/m² floor area covered</td>
</tr>
<tr>
<td>Linear control</td>
<td>£75/m² floor area covered</td>
</tr>
<tr>
<td>LED lighting</td>
<td></td>
</tr>
<tr>
<td>No control</td>
<td>£95/m² floor area covered</td>
</tr>
<tr>
<td>Linear control</td>
<td>£105/m² floor area covered</td>
</tr>
</tbody>
</table>

**Table B.5** – Relevant materials costs for the optimisations described in Chapters 10 and 11.
Bibliography


DCLG (2009). Improving the energy efficiency of buildings and using planning to protect the environment.


DECC (2008). Reducing the uk’s greenhouse gas emissions by 80 percent by 2050.


Forrester, A. I. J. (2013). Personal communication.


IPCC (2013). Working group 1 contribution to the ipcc fifth assessment report: Climate change 2013: The physical science basis. Technical report, IPCC.


White, C. (2012). Founder of white design, architects working in strawbale buildings. Personal communication by e-mail.


Appendix C

Publications

Peer reviewed conference papers:

The following papers were published during the course of the research described in this thesis. Tresidder was the lead author on the second and third paper listed, and a contributing author on the first one.


The original text of these papers is included in the following section.
HOW TO INTEGRATE OPTIMIZATION INTO BUILDING DESIGN PRACTICE: LESSONS LEARNT FROM A DESIGN OPTIMIZATION COMPETITION

Yi Zhang¹, Andy Tindale², Arturo Ordonez Garcia³, Ivan Korolija¹, Esmond G Tresidder¹, Marco Passarelli², Penelope Gale⁴

¹ De Montfort University, Leicester, United Kingdom
² DesignBuilder Software Ltd., Stroud, United Kingdom
³ Universitat Rovira i Virgili, Tarragona, Spain
⁴ Zero Energy Design Ltd., Manchester, United Kingdom

ABSTRACT
As part of the Advanced Design + Optimization (ADOPT) project, we organized a design optimization competition in late 2012, with an original goal of testing building design optimization tools and their applications in design process. Despite that our original goal was missed due to software delays, the competition itself was a great success, with more than 200 people taking part and 3 winners selected from 30 final submissions. These submissions provide excellent insight into the (manual) optimization approaches taken by participants with diverse background and experiences. This paper presents in detail how the competition was organized, and analyses the submissions received. In addition, the application of design competition in classrooms is discussed. We believe the lessons learned from the competition will guide future development of optimization tools and their integration into the building design practice.

INTRODUCTION
This paper presents a building design optimisation competition held between November and December 2012. The competition was part of the Advanced Design + Optimisation (ADOPT) project funded by the UK Technology Strategy Board. In the project, DesignBuilder Software Ltd., De Montfort University and Zero Energy Design Ltd. teamed up to develop innovative building performance simulation software for assisting decision making in various stages of life cycle of buildings. The main goal of the project is to develop building design tools with robust algorithmic optimisation capability suitable for selecting practical and cost-effective energy-saving measures. A range of tools have been developed during the project, such as DesignBuilder JobServer and Optimisation, jEPlus+EA⁵, surrogate modelling methods, and the jEPlus Simulation Server (JESS) platform. The original purpose of the competition was to raise awareness of building design optimisation methods, and to test the usability of the ADOPT tools. However, the outcome from the competition exceeded our previous expectations in many ways. In this paper we will discuss the experiences and lessons learned from the competition.

The application of numerical optimization in building design has gained wide interest from both academics and practitioners. Applications of optimisation methods have been reported in many academic publications, most of which were carried out by coupling optimisation tools (e.g. Matlab Global Optimisation Toolbox, GenOpt ⁶, or research packages developed in house) with building simulation tools. Such approaches are not suitable for practitioners, who are often constrained by time, programming skills, and access to suitable computing facilities. Specialized optimisation tools also exist, such as BEOpt⁷ for residential buildings and Ecotect for shading devices. Whole-building optimisation tools that fit in various stages of the design process are yet to be developed. Early examples of such developments can be seen in Autodesk Vasari, IES OPTIMISE, and DesignBuilder Optimisation. The question remains, however, whether optimization tools can bring about a step change in building-design practice, and give us better buildings as a result. The intention of the competition is to find some answers to this question.

Why did we choose the format of a design competition? Compared to alternative methods for collecting information on how optimisation methods fit into real-life design process, such as surveys, interviews and focus groups, a competition is more ‘fun’. Firstly, participants can be more motivated to do their best in the task. Secondly, apart from the design brief and the information about the availability of tools, participants are not exposed to communications that may interfere with their own decision-making process. In this way, we (the organizers) can maintain a passive role, and derive findings only from submitted reports and feedbacks.

We had two questions in mind when designing the competition:

- What are the processes used by designers when approaching a challenging multi-criteria design problem?
• How useful are optimisation tools in such processes?
Since the planned competition was coincided with teaching activities of some of co-authors, we were able to test the competition in classroom for teaching purpose, which will be discussed further in the paper.
The body of the paper contains four parts: details of the design of the competition, analysis of the submissions and results, case study on design and optimisation approach used by competition entrants, and discussions on our experience of using the competition for training purpose.

DESIGN OPTIMISATION COMPETITION
When designing the competition task, we drew lessons from three sources as references. Reinhart and colleagues (Reinhart et al., 2011) reported their experience of teaching building energy simulation as a game to architectural students. The students were given 90 minutes to complete a task of improving an office building design with a total of over 400,000 design variants. The available design options included building forms, orientations, envelope configurations (roof, wall, window constructions and glazing ratios), external shading, lighting, and control options. Cost for each design is calculated using rough estimates and must fit within a set budget. The students were not required to carry out simulations. Instead, they submitted proposed designs to a team of modellers who carried out annual simulations and provided monthly fuel breakdown and heat balance reports back to the students. The students then submitted alternative designs after reviewing the results from the previous rounds. Due to the 90 minute time constraints, the students could only test up to 10 different designs during the game.
The arrangement of the simulation game is clearly appropriate for the architectural students who have not received sufficient training in modelling. As a result, they have to rely on limited feedback from dedicated modellers. Also, due to the expensive nature of building simulation, the students can only explore a small number of design options, which makes it infeasible to adopt a “scientific” approach, such as carrying out sensitivity analysis on design parameters. This situation is rather similar to what architects face in real-life projects, i.e. their ability to explore design options is severely constrained by the limitations of communication, time and resources. For our purpose, however, the participants are expected to be sufficiently fluent with modelling, and the time/resources constraints are not as strict as those in the classrooms.
The second reference point was from the experience of one of the coauthors on organizing a competition between human design approach and algorithmic optimisation. The competition was carried out over the course of a 5 day MSc module at the Centre for Alternative Technology (CAT) in Wales. Students were given a building-design problem of a dwelling, which they optimised through trial-and-error and analysis of thermal, cost and embodied CO2 models. At the same time the same design problem was approached using a surrogate-modelling optimisation approach. The design options included shape, glazing ratio, window and construction type, insulation, airtightness, shading, lighting, heating and renewable systems. The total number of designs was over 900,000. All known designs had previously been calculated, so a comparison of the different optimisation approaches against the known Pareto front (carbon emission vs. cost) was possible.
For us, part of the purpose of this exercise is to see how optimisation algorithms perform compared to empirical design approaches. Ten students with 5 computers worked 10 hours per day for 2.5 days on the problem, and produced solutions that are comparable to what algorithms would find within 5 hours on one computer. The task was relatively simple and did not reflect the level of complexity that designers regularly face in non-domestic buildings. The above two examples both have prescribed design options and finite number of possible designs. We wanted a more open-ended problem of which neither the organizers nor the participants can know what the best solution is.
The example of open-ended problems came from the student modelling competitions of the IBPSA Building Simulation conferences. The competitions in BS’09 and BS’11 were modelling of control strategies for the hybrid ventilation system of a public building. The building was predefined; however, participants had freedom to implement any control strategies using any modelling tools, as long as the comfort criteria were met. Entries were judged by the reported modelling approach and the resultant energy performance. In BS’13, the competition task is to design an energy positive dwelling with a focus on matching renewable sources to predefined household consumption patterns. Since participants can use any simulation tools, as long as it is “validated code”, the competition is open to abuse simply because of the error-prone nature of modelling.
We debated long and hard on whether to make submitting in DesignBuilder format a requirement of this competition. Unfortunately we could not find a viable alternative option that allowed us to check the quality of the models and to judge entries based on objective measures. (Just imagine how we were going to read through 30 EnergyPlus models to check errors if we had allowed IDF as a submission format!) In the end, we decided to allow only DesignBuilder submissions while allowing sufficient freedom for innovative designs. The competition should provide a good challenge for a range of different levels of skill with building design and building energy simulation. However, the
participants needed some experience with DesignBuilder and EnergyPlus simulation to enter the competition.

All of the above-mentioned competitions focus on energy, with either cost or comfort as a constraint. Real-life problems tend to be more complicated. In this competition, we define the following design objectives alongside functional specifications:
- Minimises operational carbon emissions.
- Minimises construction cost.
- Satisfies the comfort criteria based on ASHRAE 55 standard.
- Satisfies the minimum daylighting requirement.

The site is located south of London, not far from Gatwick Airport and the layout is shown in Figure 1. The building can be positioned anywhere within the red dashed site boundary.

A summary of the required building size is as below:
- Total floor area - 3,000m²
- Total “Office” floor area (Office activity template) - At least 2,400m², of which at least 320m² are cellular offices
- Total “Utility” floor area (Utility activity template) - At least 420m²
- Remainder of the space are “Circulation” floor area (Circulation activity template)
- Minimum storey height is 3.2m

Apart from the constraints mentioned above, participants have complete freedom to create any building form and internal layout. All design options that are available to participants are summarized in Table 1. All components are costed in DesignBuilder.

The competition entries must include a DesignBuilder model and a brief report form. The entries were judged through a 2 stage process. Stage 1 identified a short list of entries based on reported carbon and cost figures. The short list of successful entries were decided using a systematic technique. Each submission will have its operational carbon emissions and construction cost data plotted on scatter diagram (see Figure 3). The short list of designs will be identified as those that meet the competition requirements and lie closest to the “Pareto Front”.

Stage 2 identified winners by further quality check of the models and subjective judgement on the following criteria:
- Daylighting and comfort performance of the building (as well as the cost and carbon performance already considered in Stage 1). The winning entries will provide a sensible balance between cost and carbon/energy performance.
- Practicalities – is the design actually buildable and would it work in practice?
- Innovation – credit will be given for creative use of new or unusual design techniques or building characteristics.
- Aesthetics – does the building design fit well within the site and look pleasing from the outside as well as fulfilling the needs of the occupants inside?
- Quality of the argument supporting the submitted design.
- Consideration and details that have gone into the submission.

The competition was launched on 24 October 2012 with a closing date for submission on 18 December, which was later extended to 28 December 2012.

### Table 1 Available design options

<table>
<thead>
<tr>
<th>Site and orientation</th>
<th>Anywhere in marked area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building geometry and layout</td>
<td>Freedom on layout provided the floor area requirements are met</td>
</tr>
<tr>
<td>Materials</td>
<td>Fixed set of 46 options</td>
</tr>
<tr>
<td>Constructions</td>
<td>Free to define custom constructions using the 46 competition materials</td>
</tr>
<tr>
<td>Window size and position</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Glazing type</td>
<td>Fixed set of 18 options, including double or triple glazing with different gases and coatings</td>
</tr>
<tr>
<td>Blinds</td>
<td>Fixed set of 5 options</td>
</tr>
<tr>
<td>Local shading</td>
<td>Fixed set of 9 options</td>
</tr>
<tr>
<td>Lighting</td>
<td>Fixed set of 5 options, including CFL, T5 and LED lighting. Some with daylight control</td>
</tr>
<tr>
<td>HVAC</td>
<td>Fixed set of 7 options, including Fan-coil, VAV, VRF, air-source heat pump, ground-source heat pump, low-temperature radiator and chilled beam systems. Some of them use only natural ventilation for cooling</td>
</tr>
<tr>
<td>Control (schedules and setpoints)</td>
<td>Unlimited</td>
</tr>
</tbody>
</table>

- 1862 -
to delays on our side, the DesignBuilder optimisation tool and the jEPlus online simulation facilities were not available to the participants until 23 November and 14 December, respectively. These delays have severely limited the usefulness of such tools to the participants. Winners of the competition were announced on 16 January, 2013.

RESULTS AND ANALYSIS

201 participants from 37 countries/regions registered for the competition, and downloaded the competition file package. By 28 December, 2012, 30 submission from 12 countries were received (see Figure 2). Spain stands out in the chart because the competition was used as an assignment for an MSc module, and the students (in groups) were required to make a submission. These account for 6 of the 10 entries from Spain. Among all entries, 14 were from academia, including students, and 16 were from practitioners.

Reported construction cost and operational carbon emissions of all entries are plotted on chart in Figure 3. In addition, a reference case is also included. The reference case is based on the example building included in the competition support pack, with minor changes so that it meets all competition requirements. Initial screening by design requirements and proximity to the Pareto front resulted in 8 submissions (see Figure 4) being shortlisted for the final round. Two of those (214 and 131) were later found to contain features that were disallowed according to the competition rules and were therefore disqualified. The red diamonds on Figure 3 are entries that were disqualified for some reason, typically:

- Discomfort or daylighting requirements not respected

Three winners were selected from the shortlist by a panel of judges from the ADOPT team which includes representatives from academia, industry consultants and software developers. All submitted material was considered in the final judging process. 1st Prize was awarded to Tran Tuan from the School of Architecture at University of Hawaii at Manoa, USA (109). Not only was Tran Tuan’s design on the “sweet spot” on the Pareto front of best designs, but also demonstrated high architectural quality and attention to detail in the design and the report. Joint 2nd prizes went Amir Rezaei-Bazkiaei from Ebert & Baumann Consulting Engineers, USA (140), and Milos Seatovic from Quiddita, Serbia, (018). Rezaei-Bazkiaei’s design is a very well thought out design using natural ventilation and based on a circular form, which is especially effective from the daylight and cost perspectives. The report explains in-depth the concepts and processes used in the design.

---

**Figure 2 Submissions locations**

- Constructions using non-competition materials
- Total building area outside the allowed limits
- Incorrect model options used.
- Site orientation changed

---

**Figure 3 Operational carbon emissions vs. construction costs**

- Winners
- Qualified
- Disqualified

---
Seatovic’s submission is a well-researched design having the lowest carbon emission in the competition, with natural ventilation with good HVAC control. The design also uses PCMs in external walls and ceilings.

Looking at reported carbon emission figures from all qualified entries (18 in total), the average value is 168.2 ton CO₂, representing a 43.5% reduction from the reference building provided to entrants. Average construction cost is GBP 3.09 million, 12.6% higher than the reference case. However, there are five entries with construction cost on par or lower than the reference case, with entry 140 showing 13.6% savings on cost. These results have shown great potential for improving building energy performance and reducing cost by careful analysis with simulation tools. Further investigations by some of the judges showed that there is still room for improvement on even the best entries. We will make the submissions available online and invite further discussions from the community.

We did a quick survey on the design solutions entrants have selected for their designs. They are summarized in the table below. Please note these include all entries.

From the submitted models and summary reports, it is clear to see that entrants have put in a substantial amount of effort in researching and improving their designs. 13 entrants reported time spent on the competition. The average is 132 person-hours. This include a large proportion of time spent on waiting for simulation results, as the competition models often take more than an hour to simulate. One entrant reported 40 hours spent on design/analysis, and 300 hours on simulation. From the authors’ own experience, the ratio can be even worse. This situation highlights the need for faster models, faster computers, automated optimisation processes, or all of these. In fact, 10 out of the 18 entrants who have left feedbacks in the report forms explicitly commented on the need for speed. The remote simulation option (via jEPlus link) was enabled in DesignBuilder at the very late stage of the competition, so few participants were able to take advantage of this facility. Nevertheless, over half (16) of the entrants reported using the DesignBuilder optimisation tool. In the next section, we will further analyse the design approach and optimisation tools used by participants.

**OPTIMISATION IN DESIGN PROCESS**

Participants were required to summarize in the report forms the design approach taken. Although the level of detail varies widely among submissions, they do provide useful insight into the entrants’ thought processes. However, a systematic review of all 30 submissions was not possible within the time constraints of this paper. Here, we report only a few selected cases to demonstrate the different approaches a (human) designer may take to tackle a challenging problem.

The first example is entry 214, submitted by an experienced energy consultant and DesignBuilder user. Below is the reported approach (with edits by the authors):
Table 2 Summary of submitted design solutions

| Form | Typically 4-5 storeys high (73%); majority have a rectangular shape (including square, ‘L’, ‘H’ and courtyard forms); only 33% have circular, pentagonal, or irregular polygonal shapes; about half of the designs have a wide aspect facing south; a few designs have jagged levels to provide self-shading. |
| Layout | There is not a clear pattern arising from the submitted designs. Some put Utility and Circulation areas as internal zones or on the North side of the building. The effect of this on energy consumption requires further investigation. |
| Window size | 26.7% of entries have high glazing level (window to wall ratio > 0.6); 33.3% have a window to wall ratio lower than or equal to 0.4; the remainder (40%) have medium glazing level. |
| Roof light | 40% of designs have skylights, sun spaces or atria |
| Construction | 20% of cases have external walls insulated to the UK best practice level or higher. Most constructions have internal exposed medium to high thermal mass. Only 4 designs have chosen external walls and roof with no dedicated thermal mass layer. Three entrants used PCMs. One of them used PCMs as the only internal thermal store. |
| Lighting | 3 entries went for LED lighting. Others chose T5, most with linear control. Only 4 entries did not have lighting control. |
| Heating | Low temperature hot water heating (43.3%), VRF (26.7%), ground source heat pump (13.3%), air source heat pump (10.0%), fan coil unit (6.7%), and 1 under-floor heating (disallowed) |
| Cooling | 17 entries (56.7%) have natural ventilation only. 12 entries have mechanical cooling, with VRF accounting for 8 of them. 1 used another system that is not allowed in the competition. |
| Control | All entrants adjusted temperature control set points in attempt to meet the comfort criteria. |

1. To find plan forms that could achieve a 2% daylight factor over 50% of the area with a window to wall ratio between 30 and 40% (figures guided by experience from UK practice). A shallow plan building is decided to meet the requirement of daylighting and effective cross ventilation.

2. To find optimum position and orientation with regard to daylight for these plan forms on the site including shading from adjacent buildings. This was done by numerous simulation on single zones on different floors to establish a minimum glazing area consistent with the daylight requirements. Glazing types and depth of building were also investigated to meet the daylight requirements.

3. The decision to adopt a naturally ventilated, heated only design approach came from experience and knowledge of UK climate.

4. The decisions on Internal layout, type of structure, lighting system and heating system was guided by experience. Low cost has been one of the main drivers in selecting these components.

5. Control strategy for heating and natural ventilation ensuring windows cannot be opened when heating is on. Both switchover dates between heating and natural ventilation, and the starting time of daily heating schedule are checked taking into account of the comfort requirements.

6. Finally, the glazing ratio, the heating set point and the natural ventilation set point were fine tuned to achieve lowest carbon emission while meeting the comfort and daylighting requirements. A short run of DesignBuilder optimisation was carried out. The optimisation result, though far from optimum, was used to guide the final decisions.

7. Further checks on the final model to see if it meets current UK building standards were performed. This step was not reported by anyone else in the competition.

It is clear that the entrant was able to make most of the design decisions based on his own experience in the field and a good understanding of the problem at hand. Simulation trials and optimisation were only needed for fine tuning of some of the parameters, such as position of the building, glazing ratio, depth of building, and temperature set points etc. This is evidently an efficient way to achieve good designs (note: 214 was disqualified for setting incorrect internal gain schedules. With that error corrected, the design shows very similar performance and cost to submission 018, one of the 2nd prize winners). However, a human designer is confined by his/her own experience and presumptions. In this example, instead of the 4 storey L-shaped building, a 8 storey rectangular block of the same depth could be put along the north edge of the site to receive more daylight and solar gain.

The next example is from entrant 209, who was studying building energy and design at postgraduate level at the time of the competition. In fact, quite a few submissions from students have shown strong academic rigor in their approaches. Unfortunately we can only present one example in this paper. According to the report, the final design was the result of investigating 83 different building models. In total, 210 hours were spent on the project. Features of the design approach are summarized below.
1. Low cost is a primary goal that was pursued throughout the design process. Some decisions were made purely on the basis of cost.

2. Simple rectangular shape was chosen, after comparing with different massing, e.g. compact, tall, or U-shaped forms.

3. Zero cost decisions (orientation, set points) were made first. Set points were adjusted individually, by analysing the sources (zones and time) of discomfort.

4. “Sensitivity analysis” was carried out on a range of design options, including shape, orientation, internal partition, external wall and roof construction and colour, window type, shading, HVAC system, lighting system, and set points, in order to establish the influence of each option on different objectives and the “relevance” metric.

5. In effect, the “sensitivity analysis” was used as a learning process, to make up the designer’s lack of experience in building energy performance to some extent. All following decisions were based on the result of the sensitivity analysis.

6. The orientation of the building was later fine tuned by rotating it at 5 degree steps. And the glazing ratio was tuned up based on the discomfort measure only.

7. The entrant is clearly uncomfortable with dealing with multiple objectives simultaneously. Various metrics were introduced along the process to assist decision making. Without a systematic plan, this method is less effective and more error-prone.

This approach represents an attempt to explore alternative design solutions systematically. The most remarkable feature is the use of a non-dimensional metric that combines all design objectives (carbon, cost, comfort and daylight). The metric is named as “relevance” and is used in the sensitivity analysis for determining the relative importance of each design parameter. The “relevance” measure effectively converts the multi-objective problem into a single objective problem. Optimisation experts will be quick to point out the issues with such approach, i.e. the arbitrary formula of “relevance” is likely to dictate final conclusions. However, humans are innately inefficient when dealing with multiobjective problems. Converting them into single objective problems is a natural process and probably the only way to tackle complex problems without the aid of numerical optimisation tools.

The entrant tried to use DesignBuilder optimisation to fine tune the glazing ratio (40%-48%), heating and cooling setback temperatures. However, due to software errors, DesignBuilder did not finish the task. Many participants left positive feedbacks about the optimisation facility, but were frustrated by the lack of maturity and simulation speed at the same time. The deficiencies of the existing tool include:

- Optimisable design variables only apply to the building level. It is not yet possible to optimise individual glazing ratios on different facades, or temperature settings in different zones, for example.
- Inadequate integration of constraints handling in the multi-objective optimisation gives rise to inefficiency.
- Parallel simulation only became available in DesignBuilder towards the closing date of the competition period. Without parallel simulations, optimisations can take a very long time, especially when using the full building model.
- The remote simulation option (through the jEPlus link with DesignBuilder) is not working reliably enough.

**EXPERIENCE IN THE CLASSROOM**

When designing the competition, the co-authors had a vision that the material could be a useful pedagogic resource for building energy training. This idea was tested in the classroom as part of an MSc course. The Rovira i Virgili University, Tarragona, Spain (URV) offers a master’s degree course on Air Conditioning Technologies and Energy Efficiency in Buildings. It includes a 10-session/30-hour module on building energy simulation, in which two of the co-authors teach. DesignBuilder and EnergyPlus are used in this module to help students develop analytical skills and understand the impact of different architectural design variables on the environmental and energetic performance of buildings. Concept of optimisation is also introduced to the students, albeit only briefly due to time constraints in class. Nevertheless, the students are exposed to the systematic approaches to optimal designs for buildings.

Since the module (November 2012) coincided with the Competition, it was decided that the final coursework for the students is participating in the Competition. Ten students were paired between themselves to form five teams. Interestingly one team broke up during the process, so finally six submissions were made. The students were motivated to do well in the Competition and achieved excellent results:

1. Two submissions were shortlisted for the final stage, including one achieving target carbon and cost measures very close to the prize winners. One project was not shortlisted, but not far off. On the other end, three submissions were among the worst designs: one with high CO₂, one with excessive cost, and one failed to meet the comfort requirement. It should be noted that all students were new to the software tools used in the Competition.
2. In this module, students focused on architectural
design aspects of buildings. Most of them have
engineering or HVAC background, and received
training on building energy systems on the same
course. Combining this and the typical practice
in Spain may help explain that 4 out of the 6
submissions used active cooling (VRF or Fan
Coil system) in their designs.
3. In teaching, systematic approaches for exploring
design options were introduced. This was
reflected in the submissions. All students
adopted a strategy to optimize building
characteristics first, and system operations next.
Most reported the use of parametric analysis and
simplified models during various stages of the
design process.

We tried to collect further feedback from the students
regarding their experience from participating in the
Competition. A brief questionnaire (in Spanish) was
given to all students. Questions include:
1. To what extent the contents taught in the module
   was useful in the Competition?
2. Was participating in the Competition relevant to
   your knowledge and skills on building design?
3. Did you enjoy doing the Competition as the final
coursework?

Students were asked to choose from (in effect) “not
at all”, “a little”, “just enough” and “very much” for
an answer. In addition, they were encouraged to give
comments on the main technical challenges they have
met, and suggestions on future improvements of the
course and the Competition.

Three votes of “a little”, six “just enough” and one
“very much” were received for Q1. Five of the
students expressed that the course was too short for
mastering a software package like DesignBuilder.
Also received were five comments on the inclusion of
more practical examples, such as working with
complex buildings, application of optimization
processes, and interpretation of simulation outputs.

Q2 and Q3 received identical responses from the
students: five answered “just enough” and the other
five, “very much”. This means that all students
considered their participation in the competition as
a positive experience. In general, students considered
that the competition was well organized. Some
commented on accepting only DesignBuilder model
submissions limits the tools can be used in the
competition. Also, the competition may contain two
categories, for entrants with different levels of skills
and experiences.

From the experience, we consider the Design
Optimisation Competition is a great pedagogic
resource. It is an excellent way to motivate students
to extend knowledge and improve skills. With the
right tools in place, it would be possible to train the
new generation of practitioners with optimisation-

based design skills and systematic approaches. These
are crucial for delivering better buildings for the
future.

CONCLUSION
The application of numerical optimization in building
design has gained ever-increasing interest from both
academics and practitioners. Commercial software
vendors have been working on tools for bringing
optimization to the industry. However, can
optimization tools really bring about a step change in
building-design practice, and give us better buildings
as a result? No clear answer has yet emerged.

We organized the Design Optimization Competition
in late 2012 as part of the ADOPT project. The
original purpose of the competition is to test
optimization tools developed in the project, and to
analyze how they fit into the design process. We
failed to achieve this goal, due to software delays.
Apart from that, the competition is an excellent
success that provided us rich source of information,
especially insights into the existing design strategies
and approaches.

The competition set out a challenging task. The
participants were provided with a specific site plan,
and were asked to come up with a design that
“optimizes” both operational carbon emissions and
construction cost, while fulfilling requirements on
thermal comfort and daylight use. The participants
have a large (practically infinite) number of design
options to explore. These options include building
form, geometry, and internal layout, orientation,
construction, openings, shading options, lighting and
HVAC, and control parameters.

The submitted designs represent a wide range of
principles and strategies. It was very interesting to
see the contrast between the approach taken by an
experience designer and that by a student who is
studying building energy performance. The student
relied more on systematic experimentation to identify
patterns and links between design variables and
performance. The designer, on the other hand, used
his wealth of experience to guide key design
decisions. However, in both approaches, we saw the
limitation of manual “optimization”, i.e. human
designers are not effective in dealing with multiple
variables or multiple objectives. This is where
optimization tools have much to offer.

The full details of the competition, including all
support documents and models, and all submissions
from the participants, will be made publicly available
on the ADOPT website\(^1\). This can be a valuable
resource for both research and education. Our
experience of using the competition in postgraduate
training has shown that students enjoyed the learning
experience from the competition. It is a good method
to impart the concept of design optimization to the
future practitioners.
ACKNOWLEDGEMENT

We would like to thank all entrants for participating in the competition, and the feedbacks they have given us. The ADOPT project is funded by the Technology Strategy Board, UK.

REFERENCE


1 ADOPT website: http://www.iesd.dmu.ac.uk/~adopt/
2 jEPlus website: http://www.iesd.dmu.ac.uk/~jeplus/
3 GenOpt website: http://gundog.lbl.gov/GO/
4 BEopt website: http://beopt.nrel.gov/
5 BS’09 Competition:
6 BS’11 Competition:
ACCELERATION OF BUILDING DESIGN OPTIMISATION THROUGH THE USE OF KRIGING SURROGATE MODELS

Es Tresidder¹, Yi Zhang¹, and Alexander I. J. Forrester²

¹Institute of Energy and Sustainable Development, De Montfort University, Leicester, United Kingdom
²School of Engineering Sciences, University of Southampton, United Kingdom

ABSTRACT
This paper describes an experiment to test the performance of Kriging surrogate modelling optimisation techniques on a building design problem with discrete design choices. Surrogate modelling optimisation offers advantages over traditional optimisation techniques on design problems with expensive (time consuming) performance evaluation models. The techniques are tested for both single and multi-objective optimisation problems with the objective of minimising both annual CO₂ emissions predicted by a dynamic simulation and construction cost. The estimated CO₂ emissions and costs of all possible designs were first established through comprehensive analysis using a multi-processor computer, enabling the performance of the optimisation to be assessed precisely against a known single optimum or Pareto front. The performance is compared against an evolutionary algorithm (EA) searching the dynamic simulation model on the same design problem. The results show that for this design problem, Kriging surrogate modelling optimisation is effective at finding estimates of optimum designs. In the case of the single-objective optimisation it is able to find the optimum in fewer simulation calls than the stand-alone EA. In the case of the multi-objective optimisation it is capable of finding a better Pareto front if the total number of simulations is restricted, although the time cost associated with Kriging does not always mean it is worth using.

INTRODUCTION
The design of low energy buildings is a non-linear optimisation problem. Many variables interact with each other, meaning the optimum assignment for each variable depends partly on the value chosen for other variables. The best combination of variables will also depend on the use intended for the building, the climatic zone in which it is located, and site specific conditions such as shading and exposure to wind. The situation is further complicated by the addition of cost minimisation as a second objective. Many different variables are important to the energy performance, but each additional variable taken into consideration makes the set of all possible designs (the design space) exponentially larger.

The time required to estimate energy performance using a dynamic simulation model may mean that the total simulation budget is severely limited. With many thousands of possible designs, and a limited simulation budget, efficient methods of searching the design space become especially valuable.

Several methods for exploring this design space have been proposed to try and find optimal building designs (Verbeeck and Hens, 2007; Coley and Schukat, 2002). One of the most promising of these, evolutionary algorithms (EAs) use Darwinian concepts of evolution by natural selection to improve a population of building designs. While this method is effective at finding optimum designs, even with complex, multi-modal and discontinuous design problems (Hasan et al., 2008; Jin, 2005), ‘convergence’ on an estimated optimum design or Pareto front can still take many hundreds or even thousands of design samples. This may be infeasible if each sample is a time-consuming dynamic simulation.

Various approaches have been taken to tackle this problem of limited time, a vast design space and time-consuming performance estimation. These have included ‘tuning’ the various parameters of the evolutionary algorithm (Wright and Alamji, 2005), reducing the complexity of the building models to allow more simulations to be run (Wang et al, 2005) and seeding the starting population with known good designs (Hamdy et al., 2011).

Another approach to this problem is to build a surrogate model after an initial sample of the dynamic simulation model (the main model), and use the EA to repeatedly search the surrogate model instead of the main model. After each search of the surrogate model the design suggested is checked on the main model. The surrogate may be much faster to interrogate than the main simulation model, meaning that more interrogations can be made in the available time. Surrogate models typically incur a time cost of their own (they are time consuming to build), but the reduction in time due to fewer main-model evaluations can more than make up for this if the main model is time consuming to interrogate (Brownlee et al, 2010). The use of surrogate models has been suggested for building design optimisation (Wetter and Wright, 2004).
This paper tests the efficacy of one sort of surrogate model, the Kriging model, for use in optimising cost-effective, low-energy building designs. Kriging optimisation has been shown to be effective in other engineering design problems (Huang et al., 2011; Forrester et al., 2007 and 2008). However, most of these examples have been on design problems with continuous variables (De Guido et al., 2011). This paper tests the effectiveness of Kriging optimisation on a single and multi-objective, highly discrete, building design problem. Its performance is compared to that of EAs searching the main model directly, with the performance measure being the total number of main-model simulations required to find the known optimum (for the single-objective optimisation), or the quality of the estimate of the Pareto front after 200 main-model simulations.

SURROGATE MODELS AND KRIGING

After first sampling the main model, the output can be approximated using a surrogate model. Instead of making calls to the building model for all results, we can fill in the gaps between the set of observed data by fitting a surrogate model, i.e. a model which can be used in lieu of direct calls to the building model by the EA. Such models are, essentially, curve fits through known data and may take many forms, including polynomial regression (Box and Draper, 1987), neural networks, radial basis functions (Broomhead and Lowe, 1988), support vector regression (Smola and Schölkopf, 2004) and the method we concentrate on here: Kriging (named after its developer, Danie Krige, see, e.g. Sachs et al., 1989). A review of surrogate models of fitness functions in evolutionary computation is made by Jin, 2005.

As an example of how powerful the Kriging method can be, Figure 1 shows the contours of the Branin function (an analytic function often used to test optimisation algorithms) and a Kriging prediction based on 20 observations. The similarity between the two contour maps is remarkable; the surrogate has captured all the key features of the true Branin function. Clearly, if the true function is time-consuming to compute, an optimisation process could save time and resources by calling on the surrogate model for data rather than the true function.

EXPERIMENT DESIGN

Building model

A building model was built using DesignBuilder\(^1\) and exported to run in EnergyPlus\(^2\). The building model was of a simple rectangular residential building, air-change rates were fixed and solar reflections not modelled in order to keep calculation times as short as possible. This allowed comprehensive analysis of all solutions so that the performance of the optimisation methods could be compared against known optima. The design problem is described in more detail in a previous paper by the authors (Tresidder et al., 2011). Each analysis of the model took around 50 seconds to run in EnergyPlus. Using such a simple model was judged to be beneficial because the aim of the paper was to compare optimisation techniques, rather than to answer specific building-design questions, and this comparison is much more accurate if the true optimum is known (Hawe and Sykulski, 2007). Ten design variables were chosen with three discrete choices available for each one.

Discrete variables

The design variables used in this study were discrete, with three choices for each one. The Kriging model, on the other hand, assumes that variables are continuous with values ranging from 0 to 1. To turn the designs suggested by the EA on the Kriging model into designs that could be tested on the main simulation model the following simple formula was used:

- If the value is below 1/3 choose option 1.
- If the value is between 1/3 and 2/3 choose option 2.
- If the value is above 2/3 choose option 3.

Objective functions

The objective function for the single-objective optimisation was the total CO\(_2\) emissions from energy used to heat, light and cool the building as described in Tresidder et al. (2011). The multi-objective optimisation also included construction-cost minimisation as an objective. The cost model was relatively simplistic and included only material costs. These costs were based on cost estimates provided by industry.

Kriging surrogate modelling

The surrogate modelling routines used here are from the freely available Matlab toolbox that accompanies Forester et al. (2008). The initial set of observed data is chosen according to a sampling routine based on evolutionary programming that searches for the Latin hypercube design which maximizes the minimum distance between sample locations, i.e. the set of experiments

---

1. http://www.designbuilder.co.uk/
that most evenly cover the design space (Morris and Mitchell, 1995).

Based on these samples a Kriging model is built, with its hyper-parameters chosen by maximising the likelihood of the data using the EA provided in the Kriging package. The Kriging model can then be used to predict the expected improvement (EI) at unsampled points in the design space. An EA is then used to search the Kriging model for designs with a high EI. After each run of the EA on the Kriging model the suggested design is tested on the main model and the information gained is incorporated in the Kriging model for subsequent optimisations. In this way, areas of the model that erroneously suggest promising designs are not allowed to distort the optimisation process.

The process of Kriging model optimisation used in this study is summarised in Figure 2 opposite.

**Single-objective optimisation algorithms and testing**

The algorithm used to search the Kriging model can instead be used to search for better designs directly on the main simulation model. In this way, it was possible to construct a like-for-like test in which, starting with the same initial sample population, the EA was used to search for optimum designs either with or without the surrogate model in place.

The single-objective optimisations with a surrogate model were run with starting sample sizes of 5, 10, 20 and 50. The algorithm was then set to run until it found the known optimum. From the same starting populations, optimisations were run without the surrogate model on population sizes that equalled the starting population size. For each sample/population size a total of 10 optimisations were run to establish the average performance of the algorithm. The performance of the two processes was compared based on the total number of main model simulations required to find the true optimum design.

**Multi-objective optimisation algorithms and testing**

The expected improvement used in Kriging single-objective optimisation can be reformulated to work with more than one objective. The EI becomes an estimate of the probability that a design suggested by the EA will improve on the current pareto front. Combining more than one objective into a single objective can be problematic, since it requires the user to pre-assign relative importance to each objective (Wang et al., 2005). However, this is not the case with a multi-objective EI, meaning we can keep a single-objective algorithm without pre-assigning weightings to each objective.

Rather than combining the two objectives (and thus having to assign weightings to each objective) to enable the same EA to be used without the surrogate model, it was decided to compare the surrogate optimisation method against a well established multi-objective EA.

The algorithm jEPlus+EA, an efficient, multi-objective EA that uses a modified version of the well known NSGAII algorithm (Deb et al. 2002) was used as the benchmark against which the performance of Kriging optimisation was compared. jEPlus+EA has previously shown very good performance (Zhang, 2012) on building optimisation problems. The

---

3 “Maximising the data” means trying to find the Kriging model that maximises the likelihood that the known data points could have come from the Kriging model.

4 Forrester et al. 2008
modifications made to the standard NSGAII algorithm were to preserve all members of the current estimate of the Pareto front from one generation to the next (meaning the population tends to grow as the generations progress), to use integer encoding and to avoid duplicate designs.

Settings for the jEPlus+EA were left at the default values present in the package, and the settings for the surrogate modelling optimisation were left as described in Tresidder et al. 2011, apart from the encoding length, which was altered as described below.

For the surrogate model optimisation the total simulation budget was set at 200 samples. Of this budget, 60 samples were used to build the initial Kriging model, with a further sample taken after each run of the EA on the Kriging model for a further 140 samples (to make a total of 200 samples). The optimisations were each run 10 times with a different random seed. This allowed an estimate of the average performance of the algorithm on this design problem.

Because Kriging assumes that variables are continuous, when it is used on a discrete design problem many points with a high expected improvement will in fact turn out to be duplicates of previously sampled designs. A database was built while the optimisation ran which stored information on all previously sampled designs. This could be checked for duplicates before calling the main model. Several ideas for how the performance of Kriging optimisation might be improved on discrete design problems were tested based on trying to reduce the number of duplicate sample points. These were:

- Discarding duplicate samples. If the EA suggests a design that has in fact already been sampled this sample point is not included in the updated Kriging model and the EA is told to start searching over again.
- Including duplicate samples. If the EA suggests a design that has already been sampled then the performance of this design is called from memory and included in the Kriging model. The total number of unique samples (and therefore simulation calls) is kept to 200.
- Reducing the encoding length. The default encoding length in the Kriging modelling package is 20bits, encoding over 10,000 possible numbers, but since there are only three different designs this means that many genetic changes will not result in phenotypic changes (leading to duplicates). Encoding lengths of 2bits and 3bits were also tested in addition to the default encoding length.

The performance of the multi-objective algorithms was compared based on several metrics. Because the true Pareto front was known, the first metric was the mean Euclidian distance between the estimated Pareto points and the nearest true Pareto point. The second metric was the standard deviation in the mean Euclidian distance for each test. This gives an idea of the variability in distance from the pareto front. For both of these metrics a lower number was considered superior. The final metric was the number of unique designs on the estimated Pareto front. More designs on the Pareto front was held to be superior.

RESULTS AND ANALYSIS

Single-objective optimisation

With the same starting population, one optimisation with a Kriging surrogate model and one without can be considered a pair. In the case of every pair tested, for all different population sizes, the Kriging surrogate optimisation always found the optimum design before the optimisation without the surrogate model. The average performance of the optimisations (from 10 runs of each) is shown in Table 1, below.

<table>
<thead>
<tr>
<th>Initial sample/ population size</th>
<th>Average number of main-model samples required to find optimum. Mean (+ SD)</th>
<th>Two-sample t-test, assuming unequal variance, P (T&lt;=t) two-tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Kriging model</td>
<td>With Kriging model</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1154 (+887)</td>
<td>84 (+38)</td>
</tr>
<tr>
<td>10</td>
<td>1325(+1012)</td>
<td>68 (+36)</td>
</tr>
<tr>
<td>20</td>
<td>584 (+348)</td>
<td>88 (+59)</td>
</tr>
<tr>
<td>50</td>
<td>625 (+409)</td>
<td>100 (+45)</td>
</tr>
</tbody>
</table>

Table 1. A summary of results for the single-objective optimisation, comparing the number of main-model samples required to find the optimum for an evolutionary algorithm with and without a Kriging surrogate model,*P<0.05; **P<0.01.*

The differences observed between the performances of the two optimisation methods were significant at the 99% confidence level for all except the initial sample size of 10, which was significant at the 95% confidence level.

Furthermore, comparing the performance of the best performing Kriging optimisation (initial sample size of 10) with the best performing non-Kriging optimisation (population size of 20) the difference between these two populations is significant at the 99.9% confidence level.

However, knowing simply that one methodology is able to find the optimum design faster than another doesn’t necessarily mean it is a lot better – they might both get very close to the optimum design very quickly, and the user may not be concerned about the
last fraction of a percent of improvement. It is perhaps better to evaluate the performance visually:

Figure 3, above, compares the performance of the best-performing Kriging optimisation (initial sample size of ten) with the best-performing stand-alone EA (population size of 20). The lines with markers are the average (mean) performance, the unmarked lines the mean + 2 standard deviations (i.e. approximately 95% of optimisations could be expected to perform better than this). It can be seen that in addition to finding the optimum design faster, the Kriging modelling optimisation is also considerably faster at getting close to the optimum. Based on these results, 95% of Kriging optimisations could be expected to outperform the mean performance of the EA operating on the main model.

**Multi-objective optimisation**

The performance of the different types of multi-objective optimisation are summarised in Table 2, opposite.

It can be seen that if the sampling budget is limited to 200 samples, the best performing of the Kriging optimisations is better, on average, on all metrics of performance than jEPlus+EA; the estimated Pareto points are closer to the true Pareto points, there is less variability in the distance of the estimated Pareto points from the true Pareto points and there are more designs on the estimated Pareto front. The improvement in performance based just on the Euclidian distance was tested using a T test and found to be significant with 99.9% confidence (P value = 0.0002).

<table>
<thead>
<tr>
<th>Optimisation method</th>
<th>Euclidian distance Mean (+mean SD in Euclidian distances)</th>
<th>Ave number of Pareto solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kriging, duplicates included, encoding 20-bit</td>
<td>7.4(+11.6)</td>
<td>29.5</td>
</tr>
<tr>
<td>Kriging, duplicates discarded, encoding 20-bit</td>
<td>9.3(+15.6)</td>
<td>30</td>
</tr>
<tr>
<td>Kriging, duplicates included, encoding 3-bit</td>
<td>4.6(+9.5)</td>
<td>38.2</td>
</tr>
<tr>
<td>Kriging, duplicates included, encoding 2-bit</td>
<td>10.2(+18)</td>
<td>26.3</td>
</tr>
<tr>
<td>jEPlus+EA</td>
<td>12.6(+11.2)</td>
<td>25.8</td>
</tr>
</tbody>
</table>

Table 2. Comparing the performance of all multi-objective optimisation algorithms tested with a sampling budget of 200 samples of the main simulation model. Algorithms that used a Kriging surrogate model are shaded in grey.

Similarly to the single-objective case, merely examining the results in a table doesn’t give us all the information we want – we cannot see how well spread out the estimated Pareto designs are, or understand how close they are to the true Pareto front compared to the rest of the design space. In order to visualise the results, the performance of the median (fifth best) performing (in terms of mean Euclidian distance), Kriging and jEPlus+EA optimisations is
plotted alongside all known designs and the known Pareto front (Figure 4).

![Figure 4](image_url)

**Figure 4.** A comparison of the estimations made by the 5th best performing Kriging (diamonds) and jEPlus+EA (crosses) optimisation runs with a sampling budget limited to 200 runs of the main simulation model. The true Pareto front is marked with a solid line and all known designs are shown as light grey circles.

Examining Figure 4, it can be seen that not only do the Kriging estimations (marked with diamonds) tend to be closer to the true Pareto front than the jEPlus+EA estimations (marked with crosses), they are also over a greater extent of the true Pareto front.

The performance of the Kriging model when duplicates were included was better than when they were excluded, and the performance was further improved by reducing the encoding length to 3 bits rather than 20 bits. However, these improvements were not solid enough to be statistically significant with 95% confidence. Shortening the encoding length further still reduced the performance on all metrics.

This calculation of which optimisation method performs best when the simulation budget is restricted is an over-simplification. Because of the time penalty associated with using the Kriging model it is useful to know how many samples are required for jEPlus+EA to match the performance of the Kriging optimisation, and then to calculate whether or not this means the Kriging model has been beneficial on this optimisation problem.

Because the performance is being measured on three different metrics, it is impossible to say exactly when the performance of one method equals or better the other (Zitzler et al., 2004). However, looking only at the Euclidian distance metric, jEPlus+EA got as close to the known Pareto front as Kriging optimisation after on average 505 main model samples.

**DISCUSSION**

In the case of the single-objective optimisation, the addition of a Kriging model enabled the optimum design to be found with dramatically fewer samples of the main simulation model. Based on the performance of the algorithms on the single-objective design problem posed in this paper, the time cost associated with Kriging becomes worth “paying” if the main-model simulation takes more than 14 seconds. Since we know that the main-model took around 50 seconds to interrogate for the building design used in this study we can conclude that, for the parameters tested, Kriging offered an advantage in single-objective optimisations.

The performance advantage offered by Kriging in a multi-objective optimisation when tested against the established algorithm jEPlus+EA was much less dramatic. Kriging did offer an advantage in the case described if the sampling budget was limited to 200 samples. However, jEPlus+EA got equally close to the Pareto front in, on average, 2.5 times the number of main-model simulation calls (Table 2). With a time penalty of approximately 15 hours for running

---

5 With the following computer set-up: Intel E5530 CPU (2.4GHz), 16GB memory, SUSE Linux 64bit (Linux version 2.6.16), Matlab version R2011b.
the Kriging model on this optimisation problem, it would actually be faster to run jEPlus+EA for 500 or more main-model samples than to run the Kriging model for 200 main-model samples. For the parameters described here, on the multi-objective optimisation problem the time “cost” of Kriging is not worth paying.

If the relationship of jEPlus+EA requiring ~500 main model interrogations to equal the performance of Kriging after 200 main-model interrogations holds for more complex building models then Kriging would start to show an advantage for optimisations on designs with main-model simulations taking more than three minutes. However, a more complex design problem may show very different behaviour under the two optimisation methods than that shown in this study. The results shown here are encouraging that an advantage might be offered by Kriging on more complex building design optimisations, but more work is required before solid conclusions can be drawn.

Running jEPlus+EA for many more generations always resulted in a better Pareto estimate on all metrics than achieved by the Kriging optimisation after 200 main-model runs. If computing power allows, and the performance of the designs is very important, running jEPlus+EA for many generations, appears to offer a better estimate of the Pareto front. The performance advantage offered by Kriging on optimisations with restricted main-model simulations was much greater for the single-objective optimisation than for the multi-objective optimisation. This suggests that jEPlus+EA performs considerably better than the standard EA provided with the Surrogat modelling optimisation package.

It may be possible to improve the performance of Kriging optimisation by changing the EA that searches the Kriging model to include some of the characteristics of the EA used in jEPlus+EA.

Potential users of the surrogate modelling techniques described in this paper should also note that this method does suffer from limitations in terms of the number of variables that can be considered. These limitations tend not to be present when using an EA searching the main model. For each additional variable that is added the Kriging model becomes exponentially more complex (and therefore more time-consuming to build). The upper limit for the number of variables that can be included is when examined in this paper, but it may be that it is not possible to use this method on many more than the 10 variables optimised in this study.

The improvement in performance made by reducing the encoding length from 20 bits to 3 bits requires more investigation, first to establish whether this result holds true generally, and if so to investigate exactly why it results in better performance.

CONCLUSIONS
Kriging surrogate modelling has been shown to work on building design problems with highly discrete design choices.

In the single-objective design problem posed in this paper use of the kriging surrogate model enabled the optimum design to be found with fewer calls to the main simulation model compared to using the same evolutionary algorithm with no surrogate model. For the single-objective optimisation the time penalty associated with using Kriging was more than compensated for by the reduced time required due to fewer main-model interrogations.

For the multi-objective design problem posed in this paper the use of a surrogate model allows a significantly better approximation of the Pareto front to be made if the number of calls to the main simulation is limited to 200. Using jEPlus+EA without a Kriging model, a Pareto front of similar quality took approximately 500 calls to the main simulation engine. However, with the building model used in this study the reduced number of calls to the main-model did not justify the increased time cost associated with using a Kriging model.

Further work is required to establish the likely reduction in main-model simulations offered by Kriging on more complex and time-consuming building models and whether this reduction justifies the increased time cost associated with building and interrogating the Kriging model.

References


Wright J and Alamji A (2005). The robustness of genetic algorithms in solving unconstrained building optimization problems. 9th international IBPSA conference, Montreal, Canada


ABSTRACT
This paper sets out to test the performance of surrogate modelling optimisation techniques on a highly-discrete building design problem. The true optimum is first established through comprehensive analysis using a multi-processor computer, this enables the performance of the optimisation to be assessed precisely. The performance is compared against a stand-alone evolutionary algorithm (EA) on the same design problem. Preliminary results show that for this design problem surrogate modelling is capable of finding the true optimum, and suggest that it may be more reliable and faster at doing so than a stand-alone EA.

INTRODUCTION
The total CO$_2$ emissions from energy use in a given building are dependent on a large quantity of variables including occupant behaviour, building location and building design. Focussing just on the building design choices still leaves an extremely large set of possible solutions. While for some of these variables the optimum may be a simple case of maximising or minimising there are many others for which choosing the optimum value will involve striking a balance between competing objectives and will depend on the values chosen for other variables. Furthermore, even for relatively simple variables the optimum assignment, when considered alongside other variables and with a limited monetary budget, is non-trivial.

For each additional variable under consideration, the number of possible designs (the design space) increases exponentially, meaning that a comprehensive analysis of every design in a dynamic simulation engine rapidly becomes prohibitively time consuming.

Methods for efficiently searching the design space for the optimum design have been proposed (Verbeeck and Hens, 2007; Coley et al., 2002), with one of the most successful and extensively studied methods being evolutionary algorithms (EAs). These use Darwinian concepts of selection, sexual reproduction, mutation and crossover to “evolve” better buildings from an initial sample population. This method has been shown to be effective at finding optimum designs (Hasan et al., 2008), but can still require many hundreds, or even thousands of samples of the dynamic simulation engine. Because this method remains very time-consuming, many studies of this type have sought to either reduce the complexity of the building models in order to minimise time taken for each sample (Wang et al., 2005), adjust the various parameters of the evolutionary algorithms in order to more efficiently search the design space (Wright and Alamji, 2005), or set the optimisation off to a good start by “seeding” the initial population with known good designs, rather than starting with a random sample (Hamdy et al., 2011).

Surrogate modelling optimisation is a method that has been used in other engineering fields for the optimisation of designs for which the evaluation of samples is time consuming (Huang et al., 2011; Forrester et al., 2007 and 2008). It involves building a probabilistic surrogate model after initial sampling of the design space, and running the evolutionary algorithm on this surrogate model, rather than on the main model. Interrogating the surrogate model is considerably faster than interrogating the dynamic simulation model. This has been noted as promising for the field of building design optimisation (Wetter and Wright, 2004). However, concerns remain regarding the ability of the surrogate model to handle highly discrete variables (Hawe and Sykulski, 2007). Since building design problems often involve choices between discrete variables, the performance of surrogate modelling optimisation techniques on these sorts of design problems needs to be established before concluding whether they offer improvements over evolutionary algorithms used on their own.

This paper tests the efficacy of surrogate modelling optimisation at finding the single-objective optimum in a design space with ten highly discrete variables. The true optimum is determined through comprehensive analysis using a multi-processor computer, and a more traditional evolutionary algorithm without a surrogate model is tested on the same problem for comparison.

METHODOLOGY

Building model
The basic building upon which the comprehensive analysis and subsequent optimisations were done was kept relatively simple in order to speed up the
comprehensive analysis process, and because the aim of the research was to test surrogate modelling as an optimisation methodology, rather than to answer particular questions about building design. The model was built in Design Builder, a user-friendly front end to the established simulation engine Energy Plus (US department of Energy, 2011). After building the model it was exported as a file to be run independently in Energy Plus. The model was simulated using typical weather data for London Gatwick, England. The model was a two-storey terraced house with a footprint of 100m² (200m² total floor area), with the first floor consisting of a lounge, kitchen, dining room, hall and toilet and the second floor having three bedrooms a hall and a bathroom. Activity schedules for lighting, heating, cooling and internal gains were left as the default for each type of zone.

**Choice of variables**

Ten variables were chosen, each of which was allocated three possible discrete values. These are shown in Table 1, below. A comprehensive analysis of all possible designs involved 59,049 simulations.

**Table 1.**

Variables to be altered in comprehensive analysis and optimisation, along with available choices for each variable.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Possible choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermal mass layer</td>
<td>None, 0.05m dense concrete, 0.1m dense concrete</td>
</tr>
<tr>
<td>Southern wall window size</td>
<td>20% of wall area glazed, 50% of wall area glazed, 80% of wall area glazed</td>
</tr>
<tr>
<td>Northern wall window size</td>
<td>20% of wall area glazed, 50% of wall area glazed, 80% of wall area glazed</td>
</tr>
<tr>
<td>Southern wall window type</td>
<td>Double-glazed, air-filled, Double-glazed, argon-filled, Triple-glazed, air filled</td>
</tr>
<tr>
<td>Northern wall window type</td>
<td>Double-glazed, air-filled, Double-glazed, argon-filled, Triple-glazed, air filled</td>
</tr>
<tr>
<td>External door type</td>
<td>Solid wood, Insulated wood, Glazed (glazing type to match northern wall window type)</td>
</tr>
<tr>
<td>Southern window shading (solid pergola) size</td>
<td>Extends 1m from building, Extends 1.5m from building, Extends 2m from building</td>
</tr>
<tr>
<td>N, E, W external wall insulation thickness</td>
<td>0.1m, 0.2m, 0.3m</td>
</tr>
<tr>
<td>Southern external wall insulation thickness</td>
<td>0.1m, 0.2m, 0.3m</td>
</tr>
<tr>
<td>Roof and floor insulation thickness</td>
<td>0.1m, 0.2m, 0.3m</td>
</tr>
</tbody>
</table>

**Choice of objective function**

The objective function chosen as a way of comparing the performance of the building designs was the total CO₂ emissions from energy used to heat, light and cool the building. Heating and cooling were provided by district gas and electrical systems, respectively, and lighting efficiency was left at the default value of 5W/m²·100lux. The CO₂ content was assumed to be 0.19kg/kWh for gas and 0.42kg/kWh for electricity. Although it is unlikely that a domestic property in London would be fitted with a cooling system this was felt to be an appropriate penalty function for overheating.

**Comprehensive analysis**

The comprehensive analysis was prepared and executed using jEPlus, a batch processor for parametric Energy Plus studies. The use of a 256-core computer enabled what would otherwise have been a prohibitively time consuming investigation. The outputs from these studies were used to calculate total operational CO₂ emissions from each of the 59,049 different designs. Knowledge of the true optimum (the design with the lowest CO₂ emissions) means the performance of the two optimisation methods could be accurately assessed, although such knowledge would of course not be available when using optimisation techniques on a real design problem.

**Optimisation algorithms**

All of the optimisation algorithms used in this study are freely available on Dr Alexander Forrester’s website (http://www.soton.ac.uk/~aijf197/academic.htm). Unless otherwise stated all settings were left as default.

In both types of optimisation, because the value for all possible designs was already known sample points were taken from the results file, rather than re-running every simulation in Energy Plus.

**Traditional evolutionary algorithm optimisation**

The evolutionary algorithm used was a genetic algorithm with the following default characteristics:
- Tournament size of 5
- 20 bits per variable
- Probability of reproduction (copying) = 0.1
- Probability of crossover = 0.5
- Probability of mutation = 0.4
- Equal chance of either single or double point crossover.

The population size and number of generations were altered as described in the results section.

**Surrogate modelling optimisation**

The surrogate modelling optimisation process starts with sampling the design space at points chosen using a Morris-Mitchell-optimal Latin hypercube. After this initial sampling has been done a Kriging
surrogate model is built that allows a prediction of the value at un-sampled points based on their proximity to previously sampled points. The same evolutionary algorithm as described above is then run on this surrogate model for a set population size and number of generations (a population size of 100, and 50 generations in this case) and the predicted best design is taken and simulated on the true model. The true value of the objective function at that point is then used to refine the surrogate model before another run of the evolutionary algorithm. This iterative process between evolving the design on the surrogate model, testing suggested optima on the main simulation model and refining the surrogate model continues for a number of iterations chosen by the operator. For a more detailed description of surrogate modelling optimisation see for example Forrester et al., 2007 and 2008.

RESULTS

Comprehensive analysis
The spread of values for the objective function was between 1843 and 2910kgCO₂ with the following values for each variable characterising the global optimum:

- Thermal mass = 0.1m dense concrete
- North window size = 20% of external wall
- South window size = 20% of external wall
- South window type = double glazed, argon filled
- North window type = triple glazed, air filled
- Door type = glazed
- Shading size = 1m
- N, E, W external wall insulation = 0.3m
- Southern external wall insulation = 0.3m
- Roof and floor insulation = 0.3m

Stand-alone evolutionary algorithm (EA)
A typical progression of the evolutionary algorithm is shown in Figure 1, below. The first point corresponds to the best-performing design in the initial population. The optimum in this case is found after 20 generations.

The performance of the evolutionary algorithm was tested across a range of different population sizes and numbers of generations. The results of these tests are shown in Table 2, below.

Table 2. Optimisation results for the stand-alone evolutionary algorithm.

<table>
<thead>
<tr>
<th>Population characteristics (size, number of generations)</th>
<th>CO₂ emissions of best design in starting population</th>
<th>Optimum design found?</th>
<th>Number of samples required to find optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 individuals 100 generations</td>
<td>2150kg</td>
<td>Not found. Second best design found.</td>
<td>Not found after 500 samples</td>
</tr>
<tr>
<td>10 individuals 50 generations</td>
<td>1910kg</td>
<td>Yes</td>
<td>130</td>
</tr>
<tr>
<td>20 individuals 25 generations</td>
<td>1960kg</td>
<td>Yes</td>
<td>400</td>
</tr>
</tbody>
</table>

Surrogate modelling optimisation
The progression of a typical surrogate modelling optimisation is illustrated in Figure 2, over. Each point in Figure 2 represents a sample of the true function. Points within the shaded area are designs sampled during the initial Morris-Mitchell-optimal Latin hypercube based sample. Subsequent points represent attempts to find the optimum design. Between each consecutive point 50 generations of evolutionary algorithm are run on the surrogate model.
The performance of the surrogate modelling optimisation was assessed for different initial sample populations. The results of these tests are summarised in Table 3, below. The total number of samples required includes the initial sample; i.e. if 70 samples were required to find the optimum and the initial sample size was 50 then 20 iterations of the optimise-test loop were run.

Table 3. Performance characteristics for runs of the surrogate modelling optimisation performed with different sizes of initial population.

<table>
<thead>
<tr>
<th>Initial sample size</th>
<th>CO₂ emissions of best design in initial sample</th>
<th>Optimum design found?</th>
<th>Total number of samples required to find optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1963kg</td>
<td>Yes</td>
<td>93</td>
</tr>
<tr>
<td>25</td>
<td>1876kg</td>
<td>Yes</td>
<td>31</td>
</tr>
<tr>
<td>50</td>
<td>1893kg</td>
<td>Yes</td>
<td>70</td>
</tr>
<tr>
<td>100</td>
<td>1915kg</td>
<td>Yes</td>
<td>113</td>
</tr>
</tbody>
</table>

DISCUSSION

Both the stand-alone evolutionary algorithm and the surrogate modelling optimisation were capable of finding optimum, or very near to optimum, designs for the tests performed. The performance of the surrogate modelling optimisation appears to be slightly better than that of the evolutionary algorithm on the basis of the total number of samples required to find the optimum. The surrogate modelling technique found the optimum design every time (the stand-alone EA failed in one test), and the number of samples required by even the worst performing surrogate modelling optimisation (113 for the sample size of 100) was still fewer than required for the best performing stand-alone EA (130 samples, for the population size of 10).

However, it is clear that how quickly either methodology finds the optimum depends heavily on the designs selected in the first phase of the process – the first population for the stand-alone EA and the initial sample for the surrogate modelling optimisation. The optimisation runs that found the optimum point fastest, for both the stand-alone EA and the surrogate modelling optimisation were those that had found very good designs, by chance, in the initial sample or starting population (see column 2 in Table 2 and Table 3). The selection of the initial population in the stand-alone EA is random, and while the selection of sample points for building the surrogate model is systematic it is made without any knowledge of the nature of the design space. Therefore the selection of both the starting population and the initial sample is a stochastic process – a particular technique or set-up may have worked well simply through chance. Although surrogate modelling performed better in the tests presented here, more trials would be required to make robust conclusions on which method had the better performance on this design problem.

Retrospectively it is clear that the design problem posed is also simpler than intended – all the variables except one (southern window type) involved simply maximising or minimising the variable, and there were probably no significant local optima. This will have made the optimum design much easier to find. Running the same tests with a more difficult design problem, either through the inclusion of a cost function or through the inclusion of more complex variables may be required to conclude more robustly that surrogate modelling is capable of handling highly discrete variables and which of the two optimisation methods is most suitable.

There may be additional improvements that can be made to the way surrogate modelling deals with discrete variables. The surrogate modelling method assumes that each variable is continuous between zero and one, and code is needed to turn these fractions into one of three discrete choices. Because changes to the population made in the “evolution” stage may be very small, the same design may be sampled many times. Up to a third of the sample designs chosen in the optimisation stage were duplicates. If a way of telling the surrogate model that only a discrete number of choices exist for a given variable this should improve performance considerably.

CONCLUSIONS

For the design problem studied in this paper both the surrogate modelling optimisation and a stand-alone evolutionary algorithm were shown to be able to find designs very close to the true optimum, and in most cases actually find it. The results presented here indicate that surrogate modelling may be able to find the optimum more reliably and with fewer sampling
points, although insufficient tests were done to make this conclusion statistically robust.

A more complex design problem, either through the addition of more variables, variables expected to give local optima, or including cost as a second objective, should allow a more robust analysis of the relative strengths and weaknesses of the two optimisation techniques.

Although surrogate modelling has been shown here to be effective at searching the design space, the assumption that each variable is continuous means that many of the points sampled during the optimisation phase will be duplicates. A method for reducing the number of duplicate samples could be expected to improve the efficiency of the surrogate modelling optimisation.

REFERENCES


Wright J and Alamji A (2005) The robustness of genetic algorithms in solving unconstrained building optimization problems. 9th international IBPSA conference, Montreal, Canada