Space Exploration and Region Elimination Global Optimization Algorithms for Multidisciplinary Design Optimization

by

Adel Ayad HassounaYounis
M.Sc., Al-Fateh University, 2001
B.Sc., Al-Fateh University, 1996

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Mechanical Engineering

© Adel Younis, 2010
University of Victoria

All rights reserved. This dissertation may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.
Space Exploration and Region Elimination Global Optimization Algorithms for Multidisciplinary Design Optimization

by

Adel Ayad Hassouna Younis
M.Sc., Al-Fateh University, 2001
B.Sc., Al-Fateh University, 1996

Supervisory Committee

Dr. Zuomin Dong, (Department of Mechanical Engineering)
Supervisor

Dr. Afzal Suleman, (Department of Mechanical Engineering)
Departmental Member

Dr. Bradley Buckham, (Department of Mechanical Engineering)
Departmental Member

Dr. Wu–Sheng Lu, (Department of Electrical and Computer Engineering)
Outside Member
Abstract

In modern day engineering, the designer has become more and more dependent on computer simulation. Oftentimes, computational cost and convergence accuracy accompany these simulations to reach global solutions for engineering design problems causes traditional optimization techniques to perform poorly. To overcome these issues nontraditional optimization algorithms based region elimination and space exploration are introduced. Approximation models, which are also known as metamodels or surrogate models, are used to explore and give more information about the design space that needs to be explored. Usually the approximation models are constructed in the promising regions where global solutions are expected to exist. The approximation models imitate the original expensive function, black-box function, and contribute towards getting comparably acceptable solutions with fewer resources and at low computation cost.

The primary contributions of this dissertation are associated with the development of new methods for exploring the design space for large scale computer simulations. Primarily, the proposed design space exploration procedure uses a hierarchical
partitioning method to help mitigate the curse of dimensionality often associated with the analysis of large scale systems.

The research presented in this dissertation focuses on introducing new optimization algorithms based on metamodeling techniques that alleviate the burden of the computation cost associated with complex engineering design problems. Three new global optimization algorithms were introduced in this dissertation, Approximated Unimodal Region Elimination (AUMRE), Space Exploration and Unimodal Region Elimination (SEUMRE), and Mixed Surrogate Space Exploration (MSSE) for computation intensive and black-box engineering design optimization problems. In these algorithms, the design space was divided into many subspaces and the search was focused on the most promising regions to reach global solutions with the resources available and with less computation cost.

Metamodeling techniques such as Response Surface Method (RSM), Radial Basis Function (RBF), and Kriging (KRG) are introduced and used in this work. RSM has been used because of its advantages such as being easy to construct, understand and implement. Also due to its smoothing capability, it allows quick convergence of noisy functions in the optimization. RBF has the advantage of smoothing data and interpolating them. KRG metamodels can provide accurate predictions of highly nonlinear or irregular behaviours. These features in metamodeling techniques have contributed largely towards obtaining comparably accurate global solutions besides reducing the computation cost and resources.

Many multi-objective optimization algorithms, specifically those used for engineering problems and applications involve expensive fitness evaluations. In this dissertation, a
new multi-objective global optimization algorithm for black-box functions is also introduced and tested on benchmark test problems and real life engineering applications.

Finally, the new proposed global optimization algorithms were tested using benchmark global optimization test problems to reveal their pros and cons. A comparison with other well known and recently introduced global optimization algorithms were carried out to highlight the proposed methods’ advantages and strength points. In addition, a number of practical examples of global optimization in industrial designs were used and optimized to further test these new algorithms. These practical examples include the design optimization of automotive Magnetorheological Brake Design and the design optimization of two-mode hybrid powertrains for new hybrid vehicles. It is shown that the proposed optimization algorithms based on metamodeling techniques comparably provide global solutions with the added benefits of fewer function calls and the ability to efficiently visualize the design space.
Table of Contents

Supervisory Committee ................................................................. ii
Abstract ................................................................................................ iii
Table of Contents ............................................................................. vi
List of Tables .................................................................................... xi
List of Figures .................................................................................. xiii
Acknowledgments ........................................................................... xv
Dedication ......................................................................................... xvi
Acronyms ........................................................................................ xvii

Chapter 1. Introduction .................................................................. 1
  1.1 Overview .................................................................................... 1
  1.2 Optimization Problems .............................................................. 2
  1.3 Local and Global Optimization .................................................. 3
  1.4 The Nature of The Design Problems Considered in this Research .... 4
  1.5 Research Objectives .................................................................. 4
  1.6 General Background .................................................................. 8
    1.6.1 Design Optimization .............................................................. 8
    1.6.2 Traditional Global Optimization Methods ............................... 15
    1.6.3 Established Global Optimization Methods .............................. 18
  1.7 Dissertation Overview ............................................................... 24

Chapter 2. Literature Review and Related Work ............................ 29
  2.1 Introduction ................................................................................. 29
  2.2 Global Optimization Algorithms ............................................... 30
  2.3 Different Approaches and Optimization Methods for Black-Box Functions .................................................................................. 32
  2.4 Region Reduction and Elimination ............................................. 36
  2.5 Space Exploration and Region Elimination for Computationally Intensive Design Problems ................................................................. 38
  2.6 Global Optimization Using Mixed Surrogate and Space Elimination in Computation Intensive Designs ................................................................. 40
  2.7 Multi-objective Optimization Based Metamodeling Techniques for Computationally Intensive Design Problems ................................................................. 43
  2.8 Design Optimization of Real Life Applications .......................... 46
    2.8.1 Optimal Design Parameters of Automotive Magnetoelectrical Brake System .................................................................................. 46
2.8.2 Design Optimization of High Efficiency EV/PHEV/EREV Electric Mode Operations ................................................................. 47
2.8.3 Optimization Algorithms Used in HEV Design .................................................. 50
2.9 Identification and Use of Appropriate Sampling Techniques ............................. 52
2.9.1 Classical Designs ........................................................................................................ 53
2.9.2 Space Filling Designs ................................................................................................ 53
2.10 Identification and Use of Appropriate Metamodels ........................................... 56
2.11 Metamodels and Computer Simulations ............................................................... 58
2.11.1 Response Surface Method ...................................................................................... 60
2.11.2 Kriging Metamodels .............................................................................................. 61
2.11.3 Radial Basis Function ............................................................................................ 64
2.12 Statistical Validation Methods ................................................................................ 67
2.12.1 Root Mean Square Error (RMSE) ........................................................................... 67
2.12.2 Relative Maximum Absolute Error (RMAE) ......................................................... 68
2.12.3 R-Square ................................................................................................................ 68
2.13 Multiple Objectives .................................................................................................. 68
2.14 Test Benchmark Problems and Industrial Design Case Studies ....................... 70

Chapter 3. Approximated Unimodal Region Elimination for Global Design Optimization ................................................................................. 73
3.1 Introduction .................................................................................................................. 73
3.2 Generic Global Optimization Problem ................................................................. 74
3.3 Major Steps of the Proposed Algorithm ............................................................... 75
3.4 Metamodeling for the Divided Region .................................................................... 76
3.5 Region Elimination Algorithm ................................................................................ 78
3.6 Testing Using Benchmark Problems ........................................................................ 82
3.6.1 Benchmark Test 1 - Alpine Function ................................................................. 83
3.6.2 Benchmark Test 2 - Banana Function ................................................................. 85
3.6.3 Benchmark Test 3 - Beak Function ......................................................................... 86
3.6.4 Benchmark Test 4 - Goldstein and Price Function (GP) ...................................... 87
3.6.5 Benchmark Test 5 - Branin function (BR) ............................................................. 88
3.6.6 Benchmark Test 6- Schaffer's F6 Function ............................................................ 89
3.6.7 Benchmark Test 7- Griewank function (GN) ......................................................... 90
3.6.8 Benchmark Test 8- Generalized Polynomial Function (GF) .......................... 91
3.7 Summary and Discussions of Test Results .......................................................... 92
3.7.1 The Proposed Algorithm Present Limitations and Continuous Work ............ 93

Chapter 4. Metamodeling and Search Using Space Exploration and Unimodal Region Elimination .................................................. 95
4.1 Introduction ........................................................................................................... 95
4.2 The Proposed Algorithm ...................................................................................... 98
4.2.1 Steps of the Proposed Algorithm .................................................................. 98
4.3 Kriging Metamodel for Approximation .............................................................. 102
4.4 Sampling Techniques ......................................................................................... 104
4.5 Region Elimination ............................................................................................ 105
4.6 Test Problems and Results ................................................................................. 105
4.7 Summary and Discussion of Test Results ......................................................... 111
4.7.1 The Proposed Algorithm’s Limitations ...................................................... 112
4.8 Conclusions ......................................................................................................... 113

Chapter 5. Mixed Surrogate and Space Elimination in Computationally Intensive Designs ............................................................ 114
5.1 Introduction ......................................................................................................... 114
5.2 Used Surrogate Models ...................................................................................... 115
5.3 The Proposed Approach .................................................................................... 116
5.3.1 Metamodel with Mixed Surrogates ............................................................. 116
5.3.2 Steps of the Proposed Approach ................................................................. 117
5.4 Test Problems ..................................................................................................... 122
5.4.1 Summary of Test Results ............................................................................. 122
5.5 Results Discussion ............................................................................................. 125
5.5.1 Convergence Capability and Accuracy ....................................................... 125
5.5.2 Performance Comparison ............................................................................. 131
5.5.3 The Proposed Algorithm’s Limitations ...................................................... 134
5.6 Conclusions ......................................................................................................... 134

6.1 Introduction ......................................................................................................... 136
6.2 Statement of the Multi-objective Optimization Problem .................................... 137
6.3 Pareto Frontier and Pareto Efficiency .............................................................. 138
6.4 Fitness Function ......................................................................................... 139
6.5 The Proposed Approach ........................................................................... 140
6.6 Sampling Guidance Function ..................................................................... 140
6.7 Adaptive Approximation Models Fitting .................................................. 141
6.8 Termination Criteria ................................................................................... 141
6.9 Steps of the Proposed Approach ............................................................... 141
6.10 Numerical Examples and Results ............................................................. 143
  6.10.1 Test Problem 1: Kita’s Function ............................................................. 143
  6.10.2 Test Problem 2: Veldhuizen and Lamont’s Function ............................. 145
  6.10.3 Test Problem 3: Comet Function .......................................................... 145
  6.10.4 Test Problem 4: The Concave Pareto Front Function ......................... 146
  6.10.5 Test Problem 5: Practical Example ...................................................... 146
6.11 Results and Discussions ............................................................................ 147
6.12 Summary of Test Results .......................................................................... 154
6.13 Conclusions ............................................................................................... 155

Chapter 7. Automotive Magnetorheological Brake (MRB) System Design
Optimization ....................................................................................................... 157
  7.1 Introduction ................................................................................................ 157
  7.2 MRB Model ............................................................................................... 159
  7.3 Formulation of the Optimization Problem ............................................... 162
  7.4 Optimization Results and Discussions ..................................................... 164
  7.5 Conclusions ............................................................................................... 169

Chapter 8. Application of the New SEUMRE Global Optimization Tool in High
Efficiency EV/PHEV/EREV Electric Mode Operations ...................................... 170
  8.1 Introduction ................................................................................................ 170
  8.2 The Vehicle Model ..................................................................................... 173
    8.2.1 Vehicle Description ............................................................................. 173
    8.2.2 Configuration ....................................................................................... 174
  8.3 The Optimization Problem Formulation ................................................... 175
  8.4 Optimization Results .................................................................................. 177
  8.5 Discussion of The Results ......................................................................... 179
8.6 Conclusions ................................................................................................................. 182

Chapter 9. Conclusions and Recommendations for Future Research .................. 184

9.1 Conclusions .............................................................................................................. 184
  9.1.1 New Region Elimination Optimization Algorithm .............................................. 184
  9.1.2 New Space Exploration Optimization Algorithm .............................................. 185
  9.1.3 New Mixed Surrogate Models Optimization Algorithm ............................... 186
  9.1.4 New Multi-objective Optimization Algorithm .................................................. 187
  9.1.5 Real Life Engineering Design Optimization Problems .................................. 187

9.2 Research Contributions ............................................................................................ 188

9.3 Future Work ............................................................................................................. 189
  9.3.1 Dividing and Partitioning the Design Space ...................................................... 189
  9.3.2 Introducing Efficient and Robust Global Optimization Algorithms .............. 190
  9.3.3 Enhancing Computer Experiments Sampling Techniques ............................... 190
  9.3.4 Testing of MOO Design Applications ............................................................. 190

References ..................................................................................................................... 191

Appendix A. Tested Benchmark Problems ................................................................. 207
  A.1 Unconstrained Test Problems ............................................................................... 207
  A.2 Constrained Test Problems .................................................................................. 211

Appendix B. Sample of SEUMRE Optimization Results on the Tested Benchmark
  Problems ....................................................................................................................... 214

Appendix C. Tested Multi-objective Optimization Benchmark Problems Results’
  Plots ............................................................................................................................... 219
List of Tables

Table 1-1. Representative Global Optimization Techniques ........................................... 23
Table 2-1. Representative Sampling Techniques ................................................................. 54
Table 2-2. Radial Basis Function Forms ................................................................................. 64
Table 2-3. Representative Metamodeling Techniques ......................................................... 65
Table 3-1. Test Results on Alpine function .......................................................................... 84
Table 3-2. Test Results on Rosenbrock’s Valley or Banana Function .................................. 85
Table 3-3. Test Results on the Beak Function ........................................................................ 86
Table 3-4. Test Results on Goldstein and Price Function ..................................................... 87
Table 3-5. Test Results on Branin Function .......................................................................... 88
Table 3-6. Test Results on Schaffer’s F6 Function ............................................................... 89
Table 3-7. Test Results on Griewank Function ...................................................................... 90
Table 3-8. Test Results on Generalized Polynomial Function ............................................ 91
Table 3-9. Algorithm Performance Comparison (Relative Computation Time) .................. 94
Table 3-10. Algorithm Performance Comparison (Number of Objective Function Evaluations) .......................................................................................................................... 94
Table 4-1. Test Results of the New SEUMRE Method ......................................................... 107
Table 4-2. Performance Comparisons between SEUMRE and other Space Exploration Optimization Methods on Selected Benchmark Test Problems .......................................................................................................................... 109
Table 4-3. Performance comparison between AUMRE and SEUMRE algorithms 110
Table 4-4. Percentage reduction of number of function evaluations and CPU time 110
Table 4-5. Optimization results on constrained test problems using SEUMRE .... 111
Table 5-1. Optimization Test Results Using Mixed Surrogates, Quadratic Response Function and Kriging........................................................................................................... 123
Table 5-2. Optimization Test Results using Mixed Surrogates, Radial Basis Function and Kriging.................................................................................................................... 123
Table 5-3. Optimization Test Results using Mixed Surrogates, Quadratic Response Function and Radial Basis Function ................................................................................ 124
Table 5-4. Optimization Test Results using Mixed Surrogates Quadratic Response Functions, Radial Basis Function and Kriging ........................................................................ 124
Table 5-5. Summary of Benchmark Test Results of the Proposed Algorithm ........ 132
Table 5-6. Performance Comparison on Selected Benchmark Test Problems........ 133
Table 6-1. Test Results of the Proposed Approach on Test Problem 1 ............................ 148
Table 6-2. Test Results of the Proposed Approach on Test Problem 2 ............................ 148
Table 6-3. Test Results of the Proposed Approach on Test Problem 3 ............................ 148
Table 6-4. Test Results of the Proposed Approach on Test Problem 4 ............................ 148
Table 6-5. Test Results of the Proposed Approach on Test Problem 5 ................... 149
Table 6-6. Summary of the Best Test Results of the Proposed Approach ............ 154
Table 6-7. Summary of Test Results of the Proposed Approach ......................... 155
Table 7-1. The Calculated Optimum Parameters and Results Comparison .......... 165
Table 7-2. Optimized MRP Prototype Specifications ........................................... 167
Table 8-1. Vehicle Specifications ............................................................................ 174
Table 8-2. Control Parameter List ............................................................................ 175
Table 8-3. Optimum Electrical/Mechanical Energy Efficiency Results Obtained from (SEUMRE) .......................................................... 177
Table 8-4. Optimum Power Split Ratio Results Obtained from SEUMRE ............ 178
Table 8-5. Summary of Optimization Algorithms Results ...................................... 179
Table B-1. Optimization Results on Six-hump Camel-back (SC) Function ............ 214
Table B-2. Optimization Results on Levy Function .............................................. 214
Table B-3. Optimization Results on Hartmann (H6) Function ............................. 215
Table B-4. Optimization Results on Six-hump Camel-back (SC) Function ............ 215
Table B-5. Optimization Results on Levy Function .............................................. 216
Table B-6. Optimization Results on Hartmann (H6) Function .............................. 216
List of Figures

Figure 1-1. Unimodal and Multimodal Functions .................................................. 13
Figure 2-1. Metamodels Replacing Time Consuming Computer Simulations ...... 35
Figure 2-2. Region Elimination Procedures ......................................................... 37
Figure 2-3. Optimization Based Metamodeling Process ................................. 57
Figure 3-1. Field of Interest and Points of Design Variable .............................. 75
Figure 3-2. Reduction of Identified Unimodal Area ........................................... 81
Figure 3-3. Flow diagram of the proposed method (AUMRE) ......................... 83
Figure 3-4. The Objective Function of Benchmark Test 1 ................................. 84
Figure 3-5. The Objective Function of Benchmark Test 2 ................................. 85
Figure 3-6. The Objective Function of Benchmark Test 3 ................................. 86
Figure 3-7. The Objective Function of Benchmark Test 4 ................................. 87
Figure 3-8. The Objective Function of Benchmark Test 5 ................................. 88
Figure 3-9. The Objective Function of Benchmark Test 6 ................................. 89
Figure 3-10. The Objective Function of Benchmark Test 7 ................................ 90
Figure 3-11. The Objective Function of Benchmark Test 8 ............................... 91
Figure 4-1. Dividing the Design Space Based on Obtained Function Values ...... 100
Figure 4-2. Identifying the Promising Unimodal Region Boundaries ............... 101
Figure 4-3. The Flow Diagram of the Proposed Algorithm ............................. 102
Figure 4-4. Convergence Process of SEUMRE for Sphere Function ................. 108
Figure 4-5. Convergence process of SEUMRE for Hartmann (H16) Function ...... 108
Figure 4-6. Performance Comparison of Space Exploration Optimization Methods for the Tested Benchmark Problems .................................................. 109
Figure 5-1. The Flow Diagram of the Proposed Algorithm ............................. 121
Figure 5-2. Convergence Trends of Mixed Surrogates for SC Test Problem ...... 126
Figure 5-3. Convergence Trends of Mixed Surrogates for H6 Test Problem ...... 127
Figure 5-4. Convergence Trends of Mixed Surrogates for Shubert Test Problem .. 127
Figure 5-5. Convergence Trends of Mixed Surrogates for Levy Test Problem ..... 128
Figure 5-6. Convergence Trends of Mixed Surrogates for Trid Test Problem ...... 128
Figure 5-7. Convergence Trends of Mixed Surrogates for H16 Test Problem ...... 129
Figure 5-8. Optimization Runs on H6 Test Problem using Mixed Surrogates, Quadratic Polynomial Function (QPF), Radial Basis Function (RBF) and Kriging ............................................................................. 130
Figure 5-9. Optimization Runs on Trid Test Problem using Mixed Surrogates, Quadratic Polynomial Function (QPF), Radial Basis Function (RBF) and Kriging ............................................................................. 130
Figure 5-10. The Proposed Algorithm Performance Comparison for Different Surrogates ........................................................................................................... 132
Figure 5-11. Performance Comparison with Other Optimization Algorithms

Figure 6-1. Flowchart of the Proposed Algorithm

Figure 6-2. Performance Space and Evaluated Points for Test Problem 1

Figure 6-3. Pareto Set Points for Test Problem 1

Figure 6-4. Pareto Frontier of Five Runs for Test Problem 1

Figure 6-5. Performance Space and Evaluated Points for Test Problem 2

Figure 6-6. Pareto Set Points for Test Problem 2

Figure 6-7. Performance Space and Evaluated Points for Test Problem 3

Figure 6-8. Performance Space and Evaluated Points for Test Problem 4

Figure 6-9. Pareto Frontier of Five Runs for Test Problem 5

Figure 6-10. Performance Space and Evaluated Point of Five Runs for Test Problem 5

Figure 7-1. Basic MRB Design (Park et. al. [159])

Figure 7-2. Chosen MRB Based on the Design Criteria

Figure 7-3. Dimensional Parameters Related to Magnetic Circuit Design

Figure 7-4. Process of Computing the Cost Function for a Random Design

Figure 7-5. Computation Cost Comparison

Figure 7-6. MR Brake System Optimization Results

Figure 8-1. The Hybrid Electric Vehicle Model

Figure 8-2. The Optimum Energy Efficiency Obtained Using SEUMRE

Figure 8-3. Computational Time Required by Optimization Algorithms to Converge to Global Solutions

Figure 8-4. Optimization Algorithms Performance Comparison at Different Vehicle Power Values (a) -60Kw; (b) -30 Kw; (c) 40 Kw; and (d) 80Kw

Figure B-1. Convergence Process of SEUMRE for Levy Function

Figure B-2. Convergence Process of SEUMRE for Hartmann (H6) Function

Figure B-3. Convergence Process of SEUMRE for C3 Function

Figure C-1. Performance Space and Evaluated Points of Five Runs for Test Problem 2

Figure C-2. Performance Space and Evaluated Points of Five Runs for Test Problem 3

Figure C-3. Pareto Frontier of Five Runs for Test Problem 3

Figure C-4. Performance Space and Evaluated Points of Five Runs for Test Problem 4

Figure C-5. Pareto Frontier of Five Runs for Test Problem 4

Figure C-6. Pareto Frontier of Five Runs for Test Problem 5
Acknowledgments

I am greatly indebted to my supervisor, Prof. Zuomin Dong, for many things. I thank him for accepting me to be one of his students at the University of Victoria, for his suggestion on this fascinating research area, for his direct supervision, and for his continuous support during this research. I also owe special thanks to him for carefully reading and invaluable suggestions and corrections of the manuscripts of this dissertation.

I would like to thank my colleagues, Leon Zhou and Kerem Karakoc for the good scientific atmosphere they offered to me during my study in University of Victoria.

I owe thanks to The Libyan Ministry of Higher Education for supporting my study in Canada. I am grateful for all research facilities that have been provided to me at the University of Victoria to carry out this work.

Although I was very happy to be in a great scientific environment represented at the University of Victoria, it was hard to achieve the needed research atmosphere without my wife’s support. I am very grateful for her continual supports and encouragement. I owe great thanks to my great parents for all things that they gave to me and taught me. Without their supports, I would have never made any success.

I have been brought up in an Islamic culture and environment in which my parents taught me the importance of appreciation to others and primarily to Allah, the Creator, and the Ultimate Source of all gifts in life. I have been always grateful to their teaching so lastly and above all, I would thank Him, the Almighty, for all His gifts, guidance and helps.
Dedication

I dedicate this work to my great parents
## Acronyms

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACO</td>
<td>Ant Colony Optimization</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Networks</td>
</tr>
<tr>
<td>ARSM</td>
<td>Adaptive Response Surface Method</td>
</tr>
<tr>
<td>AUMRE</td>
<td>Approximated Unimodal Region Elimination</td>
</tr>
<tr>
<td>CAE</td>
<td>Computer Aided Engineering</td>
</tr>
<tr>
<td>CAM</td>
<td>Computer Aided Manufacturing</td>
</tr>
<tr>
<td>CCD</td>
<td>Central Composite Designs</td>
</tr>
<tr>
<td>CFD</td>
<td>Computational Fluid Dynamics</td>
</tr>
<tr>
<td>CHB</td>
<td>Conventional Hydraulic Brake</td>
</tr>
<tr>
<td>CNC</td>
<td>Computer Numerical Control</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit Computation Time</td>
</tr>
<tr>
<td>DACE</td>
<td>Design and Analysis of Computer Experiment</td>
</tr>
<tr>
<td>DIRECT</td>
<td>Dividing Rectangular</td>
</tr>
<tr>
<td>DOE</td>
<td>Design of Experiments</td>
</tr>
<tr>
<td>EREV</td>
<td>Extended Range Electrical Vehicle</td>
</tr>
<tr>
<td>EV</td>
<td>Electric Vehicle</td>
</tr>
<tr>
<td>EMO</td>
<td>Evolutionary Multi-objective Optimization</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithms</td>
</tr>
<tr>
<td>GHG</td>
<td>Green House Gases</td>
</tr>
<tr>
<td>GO</td>
<td>Global Optimization</td>
</tr>
<tr>
<td>FEA</td>
<td>Finite Element Analysis</td>
</tr>
<tr>
<td>ICE</td>
<td>Internal Combustion Engine</td>
</tr>
<tr>
<td>IL</td>
<td>Inductive Learning</td>
</tr>
<tr>
<td>Iter</td>
<td>Iterations</td>
</tr>
<tr>
<td>KRG</td>
<td>Kriging</td>
</tr>
<tr>
<td>LHD</td>
<td>Latin Hypercube Designs</td>
</tr>
<tr>
<td>MARS</td>
<td>Multivariate Adaptive Regression Surface</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>MDO</td>
<td>Multidisciplinary Design Optimization</td>
</tr>
<tr>
<td>MG</td>
<td>Motor Generator</td>
</tr>
<tr>
<td>MOP</td>
<td>Multi-objective Optimization</td>
</tr>
<tr>
<td>MRB</td>
<td>Magnetorheological Break</td>
</tr>
<tr>
<td>MRFs</td>
<td>Magnetorheological Fluids</td>
</tr>
<tr>
<td>MSSE</td>
<td>Mixed Surrogate Space Exploration</td>
</tr>
<tr>
<td>MPS</td>
<td>Mode Pursuing Method</td>
</tr>
<tr>
<td>PHEV</td>
<td>Plug In Hybrid Electric Vehicle</td>
</tr>
<tr>
<td>PSAT</td>
<td>Powertrain System Analysis Toolkit</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>QPF</td>
<td>Quadratic Polynomial Function</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>RSM</td>
<td>Response Surface Method</td>
</tr>
<tr>
<td>RTM</td>
<td>Rear Traction Motor</td>
</tr>
<tr>
<td>SA</td>
<td>Simulated Annealing</td>
</tr>
<tr>
<td>SCD</td>
<td>Small Composite Designs</td>
</tr>
<tr>
<td>SDP</td>
<td>Stochastic Dynamic Programming</td>
</tr>
<tr>
<td>SEUMRE</td>
<td>Space Exploration and Unimodal Region Elimination</td>
</tr>
<tr>
<td>SOC</td>
<td>State of Charge</td>
</tr>
</tbody>
</table>
Chapter 1. Introduction

1.1 Overview

With the rapid advances in Computer Aided Design, Engineering and Manufacturing (CAD/CAE/CAM), virtual prototyping of a new design by using computer modeling, analysis and simulation tools has become more common. The computational function modules in CAD/CAE/CAM, including finite element analysis (FEA), computational fluid dynamics (CFD), kinematics/dynamics analysis, motion animation and CNC tool path simulation, automatically evaluate and accurately predict the performance of a mechanical design. It is quite natural to further extend the practice to allow design optimizations be carried out using these virtual-prototyping “black-box” functions as the objective and constraint functions. These optimizations are used to identify the best combination of design parameters in the complex, multidisciplinary design problems. However, this type of optimizations often has multimodal objective function and non-convex feasible regions as well as discrete variables and discontinuous objective functions, requiring special global optimization search tools. Conventional optimization methods, such as conjugate gradient, quasi-Newton, and sequential quadratic programming, which perform very well on a typical local optimization problem, get often
trapped into local minima and are unable to identify the global minimum of a design problem. On the other hand, the computation intensive nature of engineering analysis and simulation software makes the use of many mature stochastic global optimization methods very difficult due to the need of extensive and costly evaluations of the objective and constraint functions [1]. Effective methods for identifying the global optimum with reduced number of objective function evaluations are seriously needed to make this new paradigm for design automation and optimization viable.

As the title of this dissertation implies, the work to be introduced and discussed is about the development of new global optimization algorithms for single and multi-objective black-box functions. In the next sections, an overview of several optimization techniques is introduced.

1.2 Optimization Problems

Optimization problems are found everywhere in our daily life. Many problems can be easily formulated and then optimized. Hence, optimization techniques are seriously needed to carry out the optimization search and find acceptable solutions for these problems.

Optimization involves finding the best solution with respect to specified criteria. In engineering design, this might typically be minimum cost or weight, maximum quality or efficiency, or some other performance indices pertaining to a disciplinary objective. Realistic optimal design involves not only an objective function to be minimized or maximized, but also constraints, which represent limitations on the design space. Numerical programming requires the mathematical representation of the design space, objective function and constraints, in terms of design variables (parameters that govern
some potential for change.) Generally, the problems of interest in engineering are of nonlinear nature, in that the dependence of the objective function and constraints on the design variables is nonlinear.

1.3 Local and Global Optimization

Local optimization algorithms generally depend on the derivatives of the cost function and constraints to aid in the search. Local optimization also depends on the search initial point. Even though the initial point is made optional for convenience, it should be provided for cases where there are multiple local minima. The minimum that is returned depends on the initial point provided.

Global optimization is defined as a branch of applied mathematics and numerical analysis which deals with the optimization of a function or a set of functions to some criteria. The easiest and common form is the minimization of one real-valued function in the parameter space $\bar{x} \in R$. There might be several constraints applied to the solution vectors $x^*$ that must be satisfied.

In real life problems and applications, functions of many variables have a huge number of local minima and maxima. Finding an arbitrary local optimum is relatively straightforward by using local optimization methods but finding the global maximum or minimum of a function is much more challenging and sometimes impossible. In addition, the discrete and discontinuous nature of the black box functions often precludes the use of efficient, mature gradient-based search algorithms.

A misconception with the global optimization is that "it always returns the best answer." A more accurate description is "it returns the best answer until the termination criteria are met."
Global optimization approaches could be deterministic, such as interval optimization and branch and bound methods, or stochastic, such as the Monte Carlo based algorithms which include simulated annealing, stochastic tunnelling, and parallel tempering. Or it could be heuristic and metaheuristics, such as evolutionary algorithms, swarm based optimization algorithms, mimetic or hybrid algorithms, and reactive search optimization.

1.4 The Nature of The Design Problems Considered in this Research

Generally, there are two types of design problems. They are either continues or discrete. Continuous design problems can be easily solved using gradient based methods. The gradient information for continues design problems can easily be obtained. When the design problems are discrete and calculating their gradient is impossible, gradient based algorithms fail and cannot solve these problems. The nature of the design problems presented in this research often have discrete or mixed variables and discontinuous objective functions, making the use of gradient information infeasible in many cases. Hence, gradient based algorithms cannot be used to carry out the optimization process. For that reason, new non-gradient optimization algorithms are applied to handle the challenge. This research introduces new global optimization algorithms based metamodeling techniques to be used to solve discrete optimization problems, such as the optimization of magnetorheological brake design and the optimization of the control strategy of hybrid electric vehicles.

1.5 Research Objectives

This work aims at introducing new global optimization algorithms. These algorithms are designed to solve complex optimization problems and specifically black-box
functions. This kind of optimization problems are computationally intensive in nature and require specific computational efforts and resources. The proposed algorithms can be classified within the category of search methods, where the solution is obtained by using the function evaluations. These optimization algorithms do not require gradient information which in most cases are not available or cannot be found. Approximation techniques are used in the proposed algorithms to replace the expensive black-box function with an approximation model (metamodel) to alleviate the burden of the computational cost. Efficient sampling techniques are also used in the new proposed algorithms. These sampling techniques generate sample points that covers the entire design space in an economic way and make those sample points equally and randomly distributed in the design space or in the region of interest to explore the design space in an efficient way. The main objectives of this dissertation are as follows:

1) **Introduce region elimination global optimization algorithm.** These optimization algorithms are seriously needed to search the design space looking for the global optima. This type of optimization algorithms start their search by either eliminating the unpromising regions from the design space or dividing the design space into subspaces and focus the search in the most promising regions. These optimization algorithms should be efficient and robust. These optimization algorithms will use approximation models and sampling techniques that will enable them to reach global solutions with less computation time. These algorithms will be used to solve real life and practical optimization problems.

2) **Introduce space exploration and unimodal region elimination global optimization algorithm.** A new space exploration optimization algorithm is introduced. The design
space is explored by sending out agents to explore and report back about their findings. Efficient sampling techniques and Kriging approximation model will be used to improve the efficiency and the robustness of this algorithm. The benefit of using space exploration optimization algorithms is that good and acceptable solutions can be reached with fewer resources, less computation time and better accuracy.

3) **Develop mixed surrogate model optimization algorithm.** A new algorithm for intensive computation optimization problem is developed. In this algorithm, mixed surrogate models are introduced to explore the design space. The benefit of using mixed surrogates is that every surrogate model has its own advantages, which will be combined to obtain an accurate response of the design problem. Every surrogate model will contribute towards building the approximate model in the promising regions by a weight factor which is determined ahead to know how much every surrogate should contributes towards building the approximation model. These optimization algorithms further reduce the computation cost for intensive computation optimization problems and yield global optimum solutions with comparably good accuracy. The performance of the proposed algorithm will be demonstrated with simulation results.

4) **Develop an efficient Pareto front finder algorithm for multi-objective optimization Problems.** Identifying the efficient Pareto set for black-box multi-objective optimization functions involves fitness evaluations that are expensive to perform. Using approximation models might help reducing the burden of these expensive evaluations. A new adaptive multi-objective approach based approximation models will be introduced to reduce the computation cost and identify the efficient Pareto
frontier. A practical application will be used to test the proposed algorithm to reveal its pros and cons.

5) **Implement the introduced and developed optimization algorithms in solving real life practical engineering applications.** The space exploration and region elimination optimization algorithm will be tested and used to optimize two practical engineering applications. The first application is the optimization of the control strategy of hybrid electrical vehicles (HEV) in which the electrical/mechanical energy conversion efficiency of electric vehicles (EV) and plug-in hybrid electric vehicles/extended range electric vehicles (PHEV/EREV) in EV mode is modeled using MATLAB Simulink based powertrain component models. In particular, a new 2 mode-plus EREV design is used as a design example. The mechanical/electrical energy efficiency needs to be maximized. The optimal vehicle control scheme needs to be generated which will help determining the speed and torque of the Motor Generators (M/Gs) of the vehicle without violating their physical constraints and achieving the overall maximum efficiency of the hybrid powertrain system. The second application is applying the proposed algorithm to solve a highly nonlinear and complex real-life engineering design optimization problem – the optimal design of automotive magnetorheological brake (MRB). The chosen design configuration of MRB will be optimized for higher braking torque and lower weight. In setting up such an optimization problem for the MRB, a cost function will be defined by including the braking torque and weight as functions of the dimensional parameters of the magnetic circuit.
1.6 General Background

1.6.1 Design Optimization

Problems that seek to maximize and minimize a mathematical function of a number of variables, subject to certain constraints, form a unique class of problems, which may be called optimization problems. Many real-world and theoretical problems can be modelled in this general framework.

The common term “optimize” is usually used to replace the terms “maximize” or “minimize”. The mathematical function that is to be optimized is known as the objective function, containing usually several variables. An objective function can be a function of a single variable for some practical problems; however, a single variable function may not be a challenge from an optimization point of view. Optimization problems may involve more than one objective function and are known as multi-objective optimization problems.

Depending on the nature of the problem, the variables in the model may be real or integer or mix of both. The optimization problem could be either constrained or unconstrained.

The following is to introduce the readers to some concepts which will be used extensively in this dissertation. These are crucial definitions that need to be understood before an optimization problem can be formulated.

Design Variables

The formulation of an optimization problem begins with identifying the underlying design variables, which are primarily varied during the optimization process. Any engineering system or component is defined by a set of quantities some of which are
viewed as variables during the design process. In general, certain quantities are usually fixed at the outset and these are called \textit{pre-assigned parameters}. All the other quantities are treated as variables in the design process and are called \textit{design or decision variables} $x_i = 1, 2, \ldots, n$. The design variables are collectively represented as a design vector

$$x^T = [x_1, x_2, \ldots, x_n]$$

A design problem usually involves many design parameters, of which some are highly sensitive to the proper working of the design. These parameters as previously defined are called design variables in the parlance of optimization procedures. There is no rule to choose a prior the parameters which may be important in a problem. The choice of the important design variables largely depends on the user. However, it is important to understand that the efficiency and speed of optimization algorithms depend on the number and type of chosen design variables.

**Constraints**

Having chosen the design variables, the next task is to identify the constraints associated with the optimization problem. In many practical problems, the design variables cannot be chosen arbitrarily; rather, they have to satisfy certain specified functional and other requirements. The restrictions that must be satisfied to produce an acceptable design are collectively called \textit{design constraints}. Constraints that represent limitations on the behavior or performance of the system are termed \textit{behavior or functional constraints}. Constraints that represent physical limitations on design variables such as availability, fabricability, and transportability are known as \textit{geometric or side constraints}. 
There are usually two types of constraints that emerge from most considerations. Either the constraints are of an inequality type or of an equality type. Inequality constraints state that the functional relationships among design variables are either greater than, smaller than, or equal to, a resource value. In engineering design problems, most of the constraints encountered are inequality constraints. Equality constraints state that the functional relationship should exactly match a resource value. Equality constraints are usually difficult to handle and, therefore, need to be avoided whenever possible. If the functional relationships of equality constraints are simpler, it may be possible to reduce the number of design variables by using the equality constraints.

**Objective Function**

After both the design variables and the constraints are decided, the last piece in the formulation procedure is to find the objective function in terms of the design variables and other problem parameters. The conventional design procedures aim at finding an acceptable or adequate design which merely satisfies the functional and other requirements of the problem. In general, there will be more than one acceptable design, and the purpose of optimization is to choose the best one of the many acceptable and available designs. Thus a criterion has to be chosen for comparing the different alternative acceptable designs and for selecting the best one. The criterion, with respect to which the design is optimized, when expressed as a function of the design variables, is known as the *criterion of merit* or *objective function*. The choice of objective function is governed by the nature of the problem. In engineering, the common objectives involve either minimization or maximization. One might minimize the production time in a manufacturing process and other might maximize the total life of that product. The
objective function for minimization is generally taken as weight in aircraft and aerospace structural design problems. In civil engineering structural designs, the objective is usually taken as the minimization of cost. The maximization of mechanical efficiency is the obvious choice of an objective in mechanical engineering systems design. Some objectives can easily be expressed in a mathematical form other might be difficult to express in a mathematical form; in these cases an approximating mathematical expression is used.

In some situations, there may be more than one criterion to be satisfied simultaneously. For example, a gear pair may have to be designed for minimum weight and maximum efficiency while transmitting a specified horsepower. An optimization problem involving multiple objective functions is known as a multi-objective programming problem. Usually users avoid formulating their optimization problem in multi-objective form because that adds more complexity and requires the optimization algorithms more time to converge to solutions. With multiple objectives there arises a possibility of conflict, and one simple way to handle the problem is to construct an overall objective function as a linear combination of the conflicting multiple objective functions. Thus, in most optimal design problem multiple objectives are avoided. Instead, the designer chooses the most important objective as the objective function of the optimization problem, and the other objectives are included as constraints by restricting their values within a certain range. Nowadays, many multi-objective optimization algorithms are being introduced to alleviate the burden of the intensive computation associated with multi-objective optimization problems.
Statement of an Optimization Problem

An optimization problem in terms of their structure can be classified into two main categories, namely constrained and unconstrained optimization problems. A general structure of a mathematical model of an optimization problem can be represented as follows:

\[
\text{Find } x^T = [x_1, x_2, \ldots, x_n] \text{ that minimizes } f(x) \quad (1-1)
\]

Subject to the constraints

\[
g_i(x) \leq 0; \quad i = 1, 2, \ldots, m
\]
\[
h_j(x) = 0; \quad j = 1, 2, \ldots, p
\]

Where \( x \) is an \( n \)-dimensional vector called the design vector, \( f(x) \) is called the objective function, and \( g_i(x) \) and \( h_j(x) \) are known as inequality and equality constraints respectively. The number of variables \( n \) and the number of constraints \( m \) and/or \( p \) need not to be relating in any way. The problem stated in Equation (2-1) is called a constrained optimization problem. Some optimization problems do not involve any constraints and can be stated as:

\[
\text{Find } x^T = [x_1, x_2, \ldots, x_n] \text{ which minimizes } f(x)
\]

Such problems are called unconstrained optimization problems.

Unimodal and Multimodal Functions

The surface structure or the shape of the objective function usually differs from one to another. Functions that need to be optimized have only one peak or valley are known as unimodal functions and those which have many peaks and valleys are known as multimodal functions. A function that has only one relative maximum is said to be
unimodal, consider the function \( f(x_1, x_2, \ldots, x_n) \) which reaches a maximum at \((x_{10}, x_{20}, \ldots, x_{n0})\) or, using vector notation, at \(X_0\). If \(X_1\) and \(X_2\) are any two points in a neighbourhood of \(X_0\) with \(|X_1 - X_0| < |X_2 - X_0|\), then a path passing successively through \(X_0, X_1,\) and \(X_2\) is unimodal if \(f(x_2) < f(x_1) < f(x_0)\). If any point \(X\) can be connected to \(X_0\) by a unimodal path defined by \(f(x)\) for all points \(X \neq X_0\), then the function \(f(x)\) is unimodal. If \(X\) and \(X_0\) are connected by unimodal straight-line paths for all \(X\), then the function \(f(x)\) is said to be strongly unimodal. An example of a one-dimensional unimodal function is shown in Figure 1-1a. An example of a one-dimensional multimodal function is shown in Figure 1-1b.

![Figure 1-1. Unimodal and Multimodal Functions](image)

**Classification of Optimization Problems**

Over the past years, considerable progress has been made in developing more capable, efficient, and robust global optimization methods. Many optimization problems that were considered difficult to solve and intractable even in recent years can now be successfully solved. The resulting wide variety in global optimization techniques means that many
users are unaware of the latest developments, and hence cannot make an adequate informed choice of the optimization method. Through this overview on the features, trends, pros and cons of existing global optimization algorithms and promising directions of new algorithm development for the computer modelling and simulation based design optimization are revealed.

Global optimization (GO) methods, in general, can be classified into two main categories: deterministic and stochastic or heuristic methods. Deterministic methods solve an optimization problem by generating a deterministic sequence of points converging to a globally optimal solution, such as the branch and bound, clustering, and tunnelling methods. These methods converge quickly to the global optimum, however they require the optimization problem having certain mathematical characteristics that may not exist in most computer analysis and simulation based global optimization problem.

The stochastic or heuristic methods are based on a random generation of feasible points, or sampled points, and nonlinear local optimization search procedures using these points. Typical stochastic optimization methods include Genetic Algorithms (GAs), Simulated Annealing (SA), Statistical Algorithms, Tabu Search, Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO). Genetic Algorithms are search methods that are inspired by the natural selection and survival of the fittest in the biological world, and are different from traditional optimization techniques since GAs involve a search from a population of solutions, rather than a single point. Simulated annealing is a generalization of the Monte Carlo method for examining the equations of state and frozen states of n-body systems. The algorithm emulates the annealing process on how liquid
freezes or metal recrystalizes in cooling. Simulated annealing is easy to implement, although the method converges slowly and it is difficult to find an appropriate stopping rule. Statistical algorithms employ a statistical model of the objective function to bias the selection of a new sample point. One of the challenges in using statistical methods is the verification on the appropriateness of the statistical model for the class of problems under consideration. Tabu search, as described by Glover [2], is a meta-heuristic superimposed on another heuristic. The approach is characterized by forbidding or penalizing moves which take the solution in the next iteration to points in the solution space previously visited to avoid redundancy in cycles [2]. As a relatively new search method, Tabu search has traditionally been used on combinatorial optimization problems. Under active research, the method continues to evolve and improve. Ant Colony Optimization is a swarm intelligence technique which was originally proposed for combinatorial problems and lately for difficult and discrete optimization problems. Particle Swarm Optimization is a population based stochastic optimization technique, inspired by social behaviour of bird flocking or fish schooling and developed by [3]. Stochastic optimization methods are capable of solving complex optimization problems, but the algorithms usually suffer from certain inefficiency in dealing with continuous multimodal functions due to two major drawbacks, namely premature convergence and weak exploitation capability. Occurrence of premature convergence often leads to a local optimum instead of global optimum, while weak exploitation capability often causes slow convergence.

1.6.2 Traditional Global Optimization Methods

Many optimization methods have been around for many years and have gained excellent reputation due to their outstanding performance and ability to identify the
design optimum, whenever used properly, in different engineering applications. An overview of these traditional optimization methods is presented to prepare for the later performance comparison of various optimization methods.

**Genetic Algorithms (GA)**

*Genetic Algorithms* (GAs) are a class of search procedures based on the mechanics of natural genetics and natural selection [4]. The idea appears first in 1967 in J. D. Bagley’s thesis on “The Behaviour of Adaptive Systems, which employs Genetic and Correlative Algorithms” [5]. The theory and applicability were then strongly influenced by J. H. Holland [6], who laid the basic principles of genetic algorithms and has been considered as the pioneer of GAs. Since then, this promising field has witnessed a tremendous development and attracted the attention of many researchers. The search process of GAs involves selection, crossover and mutation. *Selection* is the mechanism for choosing individuals or strings for reproduction according to their fitness which is measured by objective function value. *Crossover* is a method of merging the genetic information of two individuals. GAs work with a population of individuals, each of these individuals could be a possible solution to a given problem. Each individual is assigned a fitness score to judge how good it is as a solution to the problem. Those highly-fit individuals are “selected” to reproduce, by cross breeding or “crossover” with other individuals in the population. The production creates new individuals as offspring, which carry some features taken from each parent. The least fit members of the population are less likely to get selected for reproduction and are forced out. A whole new population of possible solutions is thus produced by selecting the best individuals from the current “generation”, and mating them to produce a new set of individuals with a higher proportion of the
characteristics possessed by the good members of the previous generation. By favouring the more fit individuals, the most promising areas of the search space are explored. Eventually the population might converge to an optimal solution to the problem. Finally, mutation is realized as a random deformation of the strings with a certain probability. This allows the search preserves genetic diversity, thus avoiding local maxima.

Compared with traditional continuous optimization methods, such as Newton or Gradient Descent methods, almost all conventional methods start their search from a single point, GAs always operate on a whole population of points or strings, leading to the robustness of the algorithm, improving the chance of reaching the global optimum and reducing the risk of being trapped within a local stationary point. Most genetic algorithms do not use any auxiliary information about the objective function, such as derivatives. Therefore, the algorithm can be applied to any continuous or discrete optimization problem. However, there is no certainty of the convergence as opposed to gradient-based methods.

**Simulated Annealing (SA)**

Simulated annealing (SA) can be briefly defined as a robust probabilistic optimization method mimicking the solidification of a crystal under slowly decreasing temperature; applicable to a wide class of problems such as travelling salesman problem [7], image reconstruction [8], and integrated circuit (IC) designs [9]. As a random-search technique which exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure, SA searches for a minimum in a more general system, forming the basis of an optimization technique for combinatorial and other problems.
Simulated annealing was developed in 1983 by Kirkpatrick [10] to deal with highly nonlinear problems. SA approaches the global maximum of a problem similarly to using a bouncing ball that can bounce over mountains from valley to valley. It begins at a high temperature which enables the ball to bounce higher over any mountains to access any valley. As the temperature declines, the ball loses its bouncing power so it can settle in a relatively small region of the valley. From the design objectives, possible valleys or states to be explored are generated. Acceptance criteria, based upon the difference between the depths of the presently explored valley and the last saved lowest valley, are used to determine probabilistically whether to stay in the new lower valley or to jump to another one. By carefully controlling the rate of cooling or the temperature, SA can effectively locate the global optimum over time. Fast annealing and very fast simulated re-annealing (VFSR), or adaptive simulated annealing (ASA) [11] can exponentially speed the cooling and the convergence of the algorithm. The strength of SA lies in its capability to deal with optimization problems with highly nonlinear, chaotic and noisy data in its objective with a large number of constraints; and in its capability to allow parameter tuning for enhanced performance. The versatile algorithm does not rely on any restrictive properties of the optimization model. A major drawback of SA is the lack of a clear trade-off between the quality of a solution and the time required to locate the solution, leading to longer computation time to converge.

1.6.3 Established Global Optimization Methods

A number of relatively new and representative stochastic global optimization methods are studied in this work. With novel ideas behind and impressive performance, nowadays these optimization algorithms, mostly based on stochastic or heuristic techniques, are
used extensively in many applications, particularly those using black-box simulations or analysis programs to provide measures on design objectives. Through a systematic review and tests, this work intends to reveal their capabilities and put their performance into perspective. These representatives, established GO tools include Particle Swarm Optimization, Ant Colony Optimization, Mode Pursuing Sampling, Region Elimination, and Approximated Unimodal Region Elimination.

Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a recently introduced global optimization technique that has been used with great success in the area of computational intelligence. PSO is a remarkable algorithm for many reasons. It has a very simple formulation which makes it easy to implement, apply, extend and hybridize, and it is a constant source of complex and emergent phenomena which are at the essence of swarm intelligence. Many people around the world are exploring PSOs and their applications. Inspired by the social behaviour patterns of organisms that live and interact within large groups, such as flocks, swarms, or herds, the method is a population based stochastic optimization technique introduced by Kennedy and Eberhart [3] after studying the social behaviour of birds. PSO shares many similarities with evolutionary computation techniques, such as GAs. The system is initialized with a population of random solutions and searches for the optima by updating generations. However, unlike GAs, PSO has no evolution operators such as crossover and mutation. The connection to a search problem is made by assigning direction vectors and velocities to each point in a multi-dimensional search space, where the individuals interact locally with their neighbours, leading to global dynamic behaviour and search patterns within the overall population. To search for food, each
member in a flock of birds determines its velocity based on their personal experience as well as information gained through interaction with other members of the flock. Each bird, a particle, flies through the solution space of the optimization problem searching for the optimum solution and its position represents a potential solution. In particle swarm terminology, the available solution in each iteration is called the swarm, which is equivalent to the population in GAs.

PSO has been applied to solve practical optimization problems and proved to be one of the promising and successful methods. Many researchers have contributed to the further development and enhancement of the method for different types of optimization problems. For example, Rahimi-Vahed et al. [12] proposed a hybrid multi-objective algorithm based on PSO and Tabu search (TS). Zhang and Chen [13] have used PSO to determine the optimal tool size for NURBS profile milling efficiently.

**Ant Colony Optimization (ACO)**

*Swarm intelligence* is a relatively new discipline that deals with the study of self-organizing processes both in nature and in artificial systems. *Ant Colony Optimization* (ACO) is one of the most successful techniques in *swarm intelligence*. The first ACO algorithms have been proposed more than fifteen years ago. Since then, significant advances in algorithmic variants, challenging applications, and theoretical foundations have been developed, establishing ACO as a mature, high-performing meta-heuristic for the solution of difficult and discrete optimization problems.

ACO algorithms are multi-agent systems in which the behaviour of each ant is inspired by the foraging of the real ants to solve optimization problems [14]. The idea of imitating the behaviour of ants during their search for food was initiated to find good solutions to
combinatorial optimization [15]. It is known that ants can communicate in a “chemical language”. Ants move randomly and leave chemical trails called pheromone on their paths. The pheromone trails guide other ants towards the place with food. Pheromone is volatile and it evaporates over time thereby preventing the trailing ants from taking the same path all the time, while maintaining the same general direction. This principle forms the basis of ACO, which is carried out in three major steps: a) initialization of the pheromone trail to guide the search; b) generation of competing solutions, which are created largely according to probabilistic state transition rules, similar to each ant constructs its path based on the state of the pheromone; c) update on the quantity of pheromone following global updating rules, in which an evaporation phase causes a fraction of the pheromone to evaporate, and a reinforcement phase makes each ant, or solution, deposit an amount of pheromone proportional to the fitness of its solution. This iterative process ends when a stopping criterion is satisfied.

In real world, ants initially wander randomly looking for food and once they find food these ants return to their colony while laying down pheromone trails. If other ants find the trial, they are likely to follow the trial, returning and reinforcing it if they eventually find food. Over time, the pheromone trail starts to evaporate, thus reducing the strength of attractiveness. The longer the time it takes for an ant to travel down the path and back, the more time the pheromones will evaporate. A short path, by comparison, gets marched over faster, leading to higher pheromone density. Thus, when one ant finds a good (i.e. short) path from the colony to a food source, other ants are more likely to follow that path, and positive feedback eventually leads all the ants following a single path. Pheromone evaporation has also the advantage of avoiding the convergence to a locally
optimal solution. If there were no evaporation at all, the paths chosen by the first ants would be excessively attractive to the following ones. In that case, exploration of the solution space would be constrained. The idea of the ant colony algorithm is to mimic this behaviour with "simulated ants" walking around the graph representing the problem to solve.

**Mode Pursuing Sampling (MPS)**

*Mode Pursuing Sampling* (MPS) method was introduced by Wang et al. [16]. MPS was proposed to serve as a new global optimization method for black-box functions with computationally challenging engineering design problems. Based on a novel mode-pursuing sampling method that systematically generates more sample points in the neighbourhood of the function mode while statistically covers the entire search space, a quadratic regression is used to detect the region containing the global optimum. The sampling and detection process iterates until the global optimum is obtained. The method is found to be effective, efficient, robust, and applicable to both continuous and discontinuous functions through intensive testing. The method can also be used as a standalone global optimization method since it does not use any existing optimization routines.

To summarize, a summary of deterministic and stochastic global optimization methods are introduced in Table 1-1. The classification of the optimization methods and the characteristics of each category are summarized. Examples of optimization methods that belong to each category are given.
Table 1-1. Representative Global Optimization Techniques

<table>
<thead>
<tr>
<th>Global Opt.</th>
<th>Classification</th>
<th>Characteristics</th>
<th>Method</th>
</tr>
</thead>
</table>
| Deterministic Methods | Gradient Based Methods | • Popular and effective.  
• More efficient than non-gradient based search methods.  
• Requiring derivative information.  
• Sensitive to the shape of the objective function.  
• Improving the efficiency of the algorithm and enhance the rate of convergence.  
• The determination of gradient components is one of the main problems in all gradient-based optimization methods. | Quasi-Newton and conjugate gradient methods for unconstrained problems, and the penalty function, gradient projection, augmented Lagrangian and sequential quadratic programming (SQP) methods for constrained problems. DFPM; BFGS; Steepest Descent; Conjugate Gradient search |
| Non-Gradient Based Methods |                        | • Working better with ill behaved objective functions.  
• Less efficient than gradient based methods.  
• Converging to a solution that may be the global optimum.  
• Requiring the optimization problem to have certain special math form that may not exist in computer simulation based global optimization problem. | Univariate search; Direct Search; Random search; Conjugate direction search (Powell); Hooks and Jeeves Method; |
| Stochastic or Heuristic Methods | Evolutionary Methods | • Following evolutionary control strategy.  
• Can be used for optimization in continues multi-dimensional space.  
• Useful for multimodal functions  
• Capable of exploring and exploiting promising regions of the search space.  
• Taking a relatively long time to locate the exact local optimum in a region of convergence and sometimes not finding the optimum with sufficient precision.  
• Random generation of feasible points | Particle Swarm Optimization (PSO); Genetic Algorithms (GA); Evolutionary Programming (EP); Evolution Strategies (ES); Simulated Annealing (SA); Sequential Kriging Optimization (SKO) |
| Region Elimination And Space Reduction Methods |                         | • Useful in multi-dimensional space.  
• Reducing the number of the design variables by dimensionality reduction.  
• Reducing the design space by eliminating non-promising regions.  
• Dramatically reducing the number of function & constraint evaluation.  
• Not expensive computationally if using the right sampling method.  
• Efficient and robust.  
• Some methods combine stochastic and | AUMRE; Fuzzy Clustering for Design Space Reduction; DIRECT; Domain Optimization algorithm (DOA) |
1.7 Dissertation Overview

This dissertation is composed of many chapters. A brief description of each chapter’s contents is introduced. The structure of this dissertation is as follows:

Chapter 1 presented a general background on optimization techniques. It briefly presented and discussed many types of optimization techniques.

Chapter 2 presents a thorough literature review and discusses directly related work that is related to global optimization methods and specifically those designed for black-box functions and computationally intensive design problems.

Chapter 3 introduces a new global optimization algorithm based on design experiments, region elimination and response surface model, namely Approximated Unimodal Region Elimination (AUMRE) method [17]. The approach divides the field of interest into several unimodal regions using design experiment data; identifies and ranks the regions that most likely contain the global minimum; forms a response surface model using additional design experiment data over the most promising region; identifies its minimum, removes this processed region, and moves to the next most promising region. By avoiding redundant searches, the approach identifies the global optimum with reduced number of objective function evaluations and computation effort. The new algorithm was tested using a variety of benchmark global optimization problems and compared with several widely used global optimization algorithms. The experiments results present comparable search accuracy and superior computation efficiency, making the new
algorithm an ideal tool for computer analysis and simulation based global design optimization.

Chapter 4 introduces a new meta-modeling, space exploration and region reduction search algorithm. This algorithm, namely Space Exploration and Unimodal Region Elimination (SEUMRE) [18], divides the design space into key unimodal regions using design experiment data; identifies the regions that most likely contain the global minimum; fits Kriging models with additional design experiments using Latin Hypercube designs over these regions; identifies their local minima, and then the global optimum. By identifying promising unimodal regions of the objective and reducing searching space, the method can find the global optimum effectively and efficiently, particularly suited for optimization problems that require extensive computation through engineering analyses and simulations. Comparisons with existing space exploration and region elimination/reduction methods using benchmark test problems have been carried out to demonstrate the advantages of the new method.

Chapter 5 introduces the mixed surrogate models approach. The custom in surrogate-based modeling of complex engineering problems is to fit one or more surrogate models and select the one that performs best. In this chapter, a new global optimization algorithm, namely Mixed Surrogates and Space Elimination Search method (MSSE) [19], is introduced. In the introduced approach, more than one surrogate model is introduced into the design space and fitted to the available data. The approach divides the field of interest into several unimodal regions; identifies and ranks the regions that likely contain the global minimum; fits a mixed surrogate models over each promising region with additional design experiments data points using Latin Hypercube Designs; identifies its
minimum and removes the processed region; and moves to the next most promising region until all regions are processed and the global optimum is identified. The proposed algorithm was tested using several benchmark problems for global optimization and compared with several widely used space exploration global optimization algorithms, showing reduced computation efforts, robust performance and comparable search accuracy, making the proposed method an excellent tool for computationally intensive global design optimization problems.

Chapter 6 describes the use of approximation models in multi-objective optimization problems that involve expensive analysis and simulation processes such as finite element analysis (FEA) and computational fluid dynamics (CFD). The use of approximation models has become more popular and more attractive. Approximation models are found to be a promising tool for multi-objective Optimization problems because of their efficiency in imitating the real model besides their compatibility for intensive computation problems. Many optimization applications, particularly those arising in engineering, involve fitness evaluations that are expensive to perform. This becomes even worse if multiple objective black-box functions need to be dealt with and evaluated. In this chapter, a new adaptive multi-objective optimization approach based meta-modeling techniques is introduced [20]. The approach can easily identify the Pareto front for multi-objective optimization problems with high accuracy. The computational cost associated with identifying the Pareto front for expensive black-box functions is reduced. The proposed approach was tested using benchmark test problems and practical example. The obtained results are found to be promising and very encouraging.
Chapter 7 describes the practical use of the proposed algorithms in optimizing real life applications. These applications include the optimization of magnetorheological brake system [21]. Unlike the conventional brake, an MRB employs the interaction between a magnetorheological fluid and an applied magnetic field to generate the retarding braking torque. The SEUMRE design optimization algorithm was used to maximize the braking torque and minimize the weight of the brake structure.

Chapter 8 describes the second application which is the optimization of the control strategy of hybrid electric vehicles (HEV). The optimal vehicle electrical/mechanical energy conversion efficiency under various powertrain component operation parameters is obtained using three alternative global optimization tools. The new SEUMRE global optimization tool used to obtain equally accurate results much efficiently. A rough look-up surface is created to demonstrate the difference in computational efficiency. The application of the SEUMRE global optimization tool allows refined and more accurate vehicle electrical/mechanical energy conversion efficiency map being created for the optimal operation of the EV/PHEV/EREV. Optimal vehicle control schemes can then be generated in determining the speed and torque of the M/Gs of the vehicle without violating their physical constraints and achieving the overall maximum efficiency of the hybrid powertrain system. Results of the design optimization are presented and compared. New design guidelines are provided in [22].

The optimized design parameters in magnetorheological brake system (MRB) and the overall maximum efficiency of the hybrid powertrain system obtained using SEUMRE illustrates SEUMRE’s capabilities in converging to global solutions with acceptable accuracy and less computation cost.
Chapter 9 summarizes the main achievements and research findings and discusses topics needing further research.
Chapter 2. Literature Review and Related Work

2.1 Introduction

Computation-intensive simulations and analyses are frequently used to support engineering design and design optimization. The high computation costs associated with these simulations and analyses preclude design optimization methods that require extensive evaluations of the objective and constraint functions. Efficient, effective and robust optimization methods are needed. Computer analysis and simulation based design optimization requires more computationally efficient global optimization tools. In computation intensive optimization problems, such as with black-box functions, global optimization techniques suffer and might fail in converging to global solutions within a specific limit of time or resources. Problems that are computational demanding can be found in many fields. They can be found in engineering, economics, mathematics, health sciences and so on. Recently, development of global optimization techniques for handling computationally challenging engineering design problems with computationally expensive black-box functions have attracted many researchers’ attention. In this chapter, a thorough literature review, pertaining to engineering optimization algorithms based
metamodelling techniques, design of experiments, space exploration and region elimination techniques, is carried out.

2.2 Global Optimization Algorithms

Efficient global optimization algorithms have found a wide range of applications in sciences and engineering. Much research has already been carried out in their continuous development and improvements. Widely known global optimization techniques include various stochastic based algorithms, such as Simulated Annealing (SA), Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), and Ant Colony (AC) optimization methods, as well as meta-model and approximation based global optimizations.

Another group of efficient global optimization algorithms is based on objective-oriented sequential sampling. Among them, DIRECT is representative. The name stands for Dividing Rectangles, which describes the way the algorithm moves towards the optimum. The algorithm was first introduced by Jones in 1993 [23] by modifying the Lipschitzian optimization to solve difficult global optimization problems. As a sampling algorithm, DIRECT requires no gradient information and decides the subsequent search based on collected data. The algorithm normalizes the search space to a unit hypercube, starts the search by sampling at the centre of the unit hypercube, and identifies the set of potentially optimal hyperrectangles which are then sampled and subdivided until a pre-specified stopping criterion is achieved.

Many other global optimization methods explicitly take into account the high computation cost involved with the evaluation of the objective and constraint functions. These techniques can be roughly divided into two groups, sequential and non-sequential
methods. Non-sequential methods are aimed at modeling the whole design space with the help of dedicated Design Of Experiment (DOE) techniques, using Response Surface Models (RSM) [24]. This approximation-based optimization method has attracted much attention in recent years. The approach approximates computation-intensive functions with a simple analytical model, or meta-model. Meta-modeling evolves from classical DOE theory, in which polynomial functions are used as response surfaces or metamodels to considerably reduce the number of objective and constraint function evaluations. A recursive approach is often used to improve the accuracy of the modeling and the search. Today, solving complex design and design optimization problems using computational intensive computer models that are constructed using advanced CAD/CAE/CAM systems is becoming increasingly common. However, when the dimension of the search increases and/or when the feasible region of the search presents irregular shape, the application of this method becomes increasingly difficult. A sequential method, on the other hand, divides the entire design space into a number of smaller and more manageable regions and identifies the optimum in each region based on their likelihood of containing the global optimum. This approximated unimodal region removing scheme is more suitable for multidimensional design spaces containing many infeasible areas. Both metamodeling and region elimination are promising computationally efficient global optimization methods that focus on the accurate identification of the global optimum with as few objective function evaluations as possible.

Response Surface Method (RSM) is used for solving a complex optimization problem through approximation, in which a regression model is used to fit a series of planned design of experiments data points to estimate the complex relationship between the
design variables and the objective functions [25]. RSM was initially developed to represent the relationship between the input and the output of a physical experiment by a simple mathematical expression. Later the use of RSM has been extended to engineering analyses that involve the execution of complex computer analysis codes, where RSM found many applications to alleviate the computational burden of such analyses [26]. Having been used effectively as metamodels [27], RSM considers the correlation between the parameters of a process and the obtained results as surfaces in the dimensional space of the variables [28]. The simulation community has used metamodels to study the behaviour of computer simulations for over twenty-five years. The most popular techniques have been based on parametric polynomial response surface approximations [27]. RSM bears a number of appealing features for analysis and simulation based global optimization, including robustness, supporting distributed computation, providing variable sensitivities, and allowing both continuous and discrete variables [29]. An improved RSM algorithm, Adaptive Response Surface Method (ARSM) was introduced in the group’s earlier research Wang et al. [1], to improve the accuracy and efficiency of global optimal design. Tests on benchmark optimization problems and on industrial fuel cell component and system design optimizations showed considerable improvements. However, the robustness and capability of the method to handle more complex design problems need to be improved.

2.3 Different Approaches and Optimization Methods for Black-Box Functions

Recently, the development of global optimization techniques for handling computationally challenging engineering design problems with computationally
expensive black-box functions has attracted much attention. Researchers have been working on innovative ideas for new algorithms and improvements of existing methods. Depending upon the level of complexity and the details of the model, the analysis or simulation performed using computationally extensive black-box functions may require the solution of large systems of partial differential equations, which might take minutes to hours to solve [30]. Black-box based optimization algorithms are used to address problems with little or incomplete information. In general, black-box based optimization algorithms provide useful tools for discovering approximate solutions, but these algorithms cannot guarantee to find the global optimum. Due to the high computation cost of black-box based optimization, analysts are typically willing to perform only a small number of objective/constraint function evaluations. This motivates towards developing special global optimization algorithms, which will produce reasonably good solutions with modest number of function evaluations.

At present, considerable shortcomings remain with most of the existing optimization methods for computationally intensive black-box functions. Traditional gradient-based algorithms cannot be used in many cases simply because derivatives are not available and finite-difference approximations are too coarse to provide trustable solutions. An alternative is using non-derivative, heuristic optimization methods - the Direct Search methods. One example is the Simplex Reflection algorithm by Nelder-Mead [31] which was designed originally for deterministic problems and was recently revised by Barton and Ivery [32] to solve stochastic problems. Other Direct Search methods include the DIRECT method by [23] and the Parallel Direct Search algorithm by Dennis and Torczon [33]. The drawback of Direct Search methods is that a large number of function
evaluations are needed since these algorithms do not take advantage of the available continuity of the objective functions.

A widely used and more practical type of optimization methods for computationally expensive functions is the response surface model or surrogate model based optimization methods. The function of the response surface model is to provide an inexpensive approximation to the costly black-box functions to identify promising search points. The most well-known and widely used algorithm is the traditional Response Surface Methods (RSM) [34] which are based on low-order polynomial regression. RSM is easy to understand and to implement. Another class of methods is the derivative-free optimization approaches by Powell [35] and by Conn et al. [36] which utilize multivariate polynomial interpolation models within a trust-region framework. Originally introduced for unconstrained optimization problems, these methods can be easily modified to deal with black-box constrained optimization problems, and they are generally more efficient than Direct Search methods. Other response surface methods for costly optimization problems include those that rely on Kriging models [37-39] and Radial Basis functions [40-42].

Methods for stochastic black-box systems have also been studied continuously. Development on the alternatives to response surface methods using the Bayes theorem was done by Kushner [43] and Zilinskas [44]. Kushner used a heuristic optimization method to select the next point for sampling based on simplified models, and Zilinskas considered one-dimensional problems. Pettunen et al. [45] and Gablonsky and Kelly [46] developed DIRECT algorithm.
Figure 2-1 shows the advantage of using metamodeling to replace black-box functions. The computation time needed to get a solution from conventional black-box function can be considerably reduced by an approximate solution from metamodeling, making computation intensive global optimization feasible.

**Figure 2-1. Metamodels Replacing Time Consuming Computer Simulations**

Heuristic methods, such as the GA, SA and *Tabu search*, can also be applied to solve stochastic black-box problems. A major advantage of these methods is their universal applicability, as they usually assume very little about the objective functions. Heuristic methods are particularly useful when the evaluation of the objective function is inexpensive. For continuous and smooth nonlinear problems, the heuristic methods require more objective function evaluations than other classes of optimization methods.

In the recent years, promising new search algorithms with their roots in the *Bayesian* global optimization method were developed with a primary focus on deterministic problems. Jones [38] proposed the *Efficient Global Optimization* (EGO) algorithm, and Zhang and Chen [13] proposed a formulation to extend the EGO scheme to stochastic systems. Named *Sequential Kriging Optimization* (SKO), the algorithm differs from the
original EGO in implementing the Kriging meta-model and in formulating the Expected Improvement (EI) function.

2.4 Region Reduction and Elimination

In metamodeling based global optimization involving computation intensive numerical analysis/simulation and multiple objectives, a major challenge arises when the unknown system has a large design space and a large number of design variables, especially when the evaluation of the objective function is costly and the number of available samples is limited. The exponentially increased computation burden presents a major obstacle to the solution of the problem. Many region elimination schemes have been introduced to keep the size of the design space under control (Figure 2-2). Among those, the most representative group is the multivariable elimination procedures, which reduce the design space or feasible region by discarding the unpromising regions that do not likely contain the optimum. These methods are similar to minimax single variable search methods in eliminating unpromising intervals on each of the independent variables. Such methods are restricted to certain types of functions, e.g., unimodal functions. In addition, to locate the best value of the objective function using these methods, the reduction in the range of the independent variables increases exponentially as the number of variables increases. This is what is known as curse of dimensionality, and why single-variable minimax interval elimination methods are not useful in multidimensional problems, since a very large number of 1-D line segments need to be eliminated in an N-D plane. Another approach, metaheuristics, uses a complex probability model to identify the promising regions with relatively higher chance to contain the design optimum. However, the
strategy is computationally less efficient due to the complexity of the probabilistic models used.

Modelling efficiency and accuracy are directly related to the design space. Some research efforts in reducing design space focused on trimming down the dimension of the design space or the number of the design variables. Box and Draper [47] introduced a method to gradually refine the response surface to better capture the real function by “screening” out unimportant variables, thus reducing the dimension of the problem and breaking the curse of dimensionality. However, the reduction of design variables or dimension is a very difficult task especially for complex, multidisciplinary problems. Alternatively, other researchers intended to reduce the size of the design space when the reduction of the dimension is difficult, since a broad range of design variables leads to large design space, making construction of the approximation models difficult and costly. Due to the lack of knowledge on the behaviour and interaction between the objective and constraint functions at the early design stage, conservative and broad bounds of design variables are often used in setting up the optimization problem.

Figure 2-2. Region Elimination Procedures
2.5 Space Exploration and Region Elimination for Computationally Intensive Design Problems

Design space exploration, region reduction and region elimination optimization methods have gained much attention due to their superior performance in converging to global solutions and in solving many complex engineering problems. Metamodeling based search, space exploration, and region reduction/elimination methods are effective optimization schemes for computation intensive global design optimization problems.

Many research efforts have been done related to this research area. Jin et al. [48] investigated the general applicability of sequential sampling for creating global metamodels which might help in exploring the design space efficiently. In their work, various sequential sampling approaches were reviewed and new approaches were proposed. A set of test problems were used to investigate the performance of these methods against onstage approach. Giunta and Watson [49] used analytical test problems to compare response surface approximations and Kriging. Forrestor and Kean [50] reviewed the present state of the art of constructing metamodels and their use in optimization strategies. Recent studies on metamodel based design space exploration method in design applications include [27, 39, 51-53]. A generic framework for integrating metamodels from multiple subsystems was presented in Meckesheimer et al. [54].

Other research efforts in reducing design space focused on trimming down the dimension of the design space or reducing the number of the design variables. These methods intend to keep the sensitive design variables and remove the insensitive ones, based upon sensitivity analysis tests. However, problem dimension is very difficult to
reduce especially for multidisciplinary design problems. Wang and Dong [1] developed a method in which the size of the design space reduces systematically by discarding the portions that correspond to objective function values larger than a given threshold at each modelling iteration, gradually reducing the design space to the neighbourhood of the global optimum. An intuitive method to systematically reduce the design space to a relatively small region is also introduced by Wang and Simpson [55]. The method is found to be able to intuitively capture promising design regions and efficiently identify the global or near global design optimum in the presence of highly nonlinear constraints. Shan and Wang [56] presented a new method that can help reducing the design space for multi-objective optimization problems. The method assumes all design goals are realistic and helps design engineers to focus on a smaller design space for further optimization. Melo and Delbem [57] have developed the search space reduction based global optimization algorithm, Domain Optimization Algorithm, with a search strategy using simple models to identify non-promising regions. The approach is iteratively applied to eliminate regions that likely contain no global optimum. Martin and Gonzales [58] proposed an approach for optimizing the dimension of the linkages for the synthesis of planar linkages. A combined gross-to-fine search method is used as the optimization strategy. Zhuge and Xue [59] proposed an integrated framework for design optimization and space minimization (IDOM), and significantly reduced design space and search cost in a set of DSP benchmark tests.

Jones and Schonlau [38] proposed Dividing Rectangles (DIRECT) algorithm by modifying the Lipschitzian optimization to solve difficult global optimization problems. Another global optimization algorithm was recently introduced by the author of this
dissertation; the algorithm is based on design experiments, region elimination and response surface model, namely *Approximated Unimodal Region Elimination* (AUMRE) method [17].

Sampling is the first, yet one of the most important steps in building a surrogate model. Sampling techniques can be generally classified into two main categories: classical designs and space filling designs. *Classical designs* are used in various experiment designs. Full factorial design is its most basic form. Central Composite Design (CCD) has been widely used even though the number of points in CCD increases exponentially with the number of design variables, making it less efficient for high dimensional problems. Besides CCD, alphabetical optimal designs are also widely used, especially D-optimal designs [25]. *Space filling* designs spread experiment points evenly throughout the design space, and largely take four different forms: Orthogonal Arrays [60], various Latin Hypercube designs [61] and [62], Hammersly Sequences [63], and Uniform Designs [64]. Space filling designs are used extensively due to their capability to well cover the design space or the region of interest by spreading the sample points evenly.

### 2.6 Global Optimization Using Mixed Surrogate and Space Elimination in Computation Intensive Designs

Metamodeling based design optimization has become an emerging research area due to their high efficiency, robustness and ease of implementation. Many new search methods have been introduced over the past decades. Simpson et al. [65] reviewed many different surrogates and recommended the usage of different surrogates for different problems. Barthelemy and Haftka in [66] presented an earlier review on the application of metamodeling techniques in structural optimization. Rikards et al. [67] and Queipo et al.
have reviewed the use of different surrogate models in aerospace industry. More recently, Forrester and Keane in [50] have presented a review on the state of the art of surrogate modeling techniques and their use in optimization. Among these approaches, the traditional sequential method that fits generated experiment data using a global approximate model, and uses this “metamodel” as a surrogate of the expensive function. In the second approach, validation and/or optimization is being carried out in loop to decide the re-sampling and remodeling strategy [69]. Sample data were iteratively generated to update the approximation and make the model as accurate to the objective function as possible. Schonlau et al. [70] developed a sequential algorithm to balance local and global searches using approximations for constrained optimization. Osio and Amon [71] developed a multistage Kriging strategy to sequentially update and improve the accuracy of surrogate approximations.

Many researchers also contributed to the better understanding of different surrogates and their effective use in design optimization. Jin et al. [72] compared different metamodeling techniques based on different criteria such as efficiency, accuracy, robustness, transparency and simplicity, and suggested the use of polynomial response surfaces for low order nonlinear problems, Kriging for low order nonlinear problems and/or high dimension problems, and radial basis function for nonlinear, high-order problems. Sasena at al. [73] used Kriging models for disconnected feasible regions. Kayman and McMathon [26] introduced what is called ADAPRES, in which a weighted regression method is applied in place of normal regression. The experimental points from the region where the design point is most likely to exist are selected using the algorithm.
In metamodeling based global optimization involving computationally intensive analysis and simulation, modeling efficiency and accuracy are directly related to the size of the design space. A large design space of the unknown system demands larger number of sample points, leading to exponentially increased computation burden.

At present, limited work has been done in using mixed or combined approximate models in the metamodel based design optimization. Most of the researches have been focused on the identification and use of the appropriate surrogate model from multiple candidates until recently when Zerpa et al. [74] presented the use of an ensemble of surrogates to construct weighted average surrogate model for the optimization of alkali–surfactant–polymer flooding process, and concluded that the weighted average surrogate model has better modeling capabilities than individual surrogates. Goel et al. [75] have also explored different approaches in which the weights associated with each surrogate model are determined based on a global cross-validation error measure, called prediction sum of squares (PRESS). Viana et al. [76] investigated how cross validation errors can help obtain the best predictor. They have suggested that fitting multiple surrogates and choosing one based on cross validation errors is a promising strategy, and cross validation errors may also be used to create a weighted surrogate.

Appropriate generation of design experiment data points is also critical to the efficiency and effectiveness of the appropriate model. Space filling designs are used extensively due to their capability to well cover the interested region by spreading the sample points uniformly and evenly. For this reason, Latin Hypercube design sampling method is adopted in the mixed surrogate approach introduced in this dissertation.
The work done and introduced in this dissertation combines the efforts of using mixed surrogates, effective reduction of design space and appropriate data sampling into one algorithm. The surrogate models, the mechanism for mixing different surrogates and combining it with unimodal region identification and space filling sampling schemes are also introduced.

2.7 Multi-objective Optimization Based Metamodeling Techniques for Computationally Intensive Design Problems

Many engineering problems have more than one objective function to optimize. These problems can be formulated as multi-objective optimization problems. Efficient, effective, and robust techniques are highly in demand to solve these complex multi-objective optimization problems. Many algorithms were introduced and more are being proposed to explore the design space of multi-objective optimization problems. It is clearly noticeable that most of the available algorithms are based on evolutionary algorithms, such as GA, PSO, and others. These algorithms are known as evolutionary multi-objective optimization (EMO). Few are those that are proposed for black-box functions.

For engineering multi-objective optimization problems (MOPs) with expensive high fidelity computational models such as complicated simulations and detailed analyses, the standard multi-objective optimization methods are hard to perform directly because of their high computational cost of functions and constraint evaluations. For this reason, the use of approximation models, which are the corresponding low fidelity models, becomes more attractive in these problems. However, due to the loss of the accuracy, it cannot guarantee the same Pareto optimal set that could be obtained from the high fidelity
models. Therefore, some efforts were made to minimize the loss arising from using the approximation models.

Two scenarios exist for solving multi-objective optimization problems with black box functions. The first scenario is by directly approximating the Pareto optimal [77-79] and this can be seen in many evolutionary algorithms (EAs). These methods are usually computationally expensive because a massive number of non-Pareto set points have to be evaluated. The other scenario is using approximate models in which each single objective function is evaluated. It is obvious that the accuracy of the Pareto optimal frontier will depend on the approximation models. Li et al. [80] used a hyper-ellipse surrogate to approximate the Pareto optimal frontier for bi-criteria convex optimization problems. If the approximation is not sufficiently accurate, then the Pareto optimal frontier obtained using the surrogate approximation will not be a good approximation of the actual Pareto optimal frontier. Wilson et al. [78] used the surrogate approximations (response surface and Kriging models) for computationally expensive analyses to explore the multi-objective design space and identify Pareto optimal points, the Pareto set, using the approximation models. Their approach minimizes the loss of the accuracy by validating the approximations before the optimization procedure. Proos et al. [81] developed a new algorithm incorporating multiple criterion optimizations into evolutionary structural optimization (ESO) using the weighting and global criterion method. Yang et al. [79] proposed a strategy named - adaptive approximation in multi-objective optimization (AAMO) for multi-objective optimization problems that need expensive computer simulations. They employed MOGA and approximation models and specifically they used Kriging. MOGA is used to improve the accuracy of the approximation models.
Yang et al. [82] also proposed the first framework to manage approximation models in MOO. In the framework, a GA based method is employed with a sequentially updated approximation model. In their approach, they update the approximation model in the optimization process and not before the optimization procedure as in [78]. The fidelity of the identified frontier solutions, however, is still built upon the accuracy of the approximation model. The work done in [82] also suffers from the problems of the GA-based MOO algorithm, i.e., the algorithm has difficulty in finding frontier points near the extreme points (the minimum obtained by considering only one objective function).

Shan and Wang [83] proposed a new method named Pareto Set Pursuing (PSP) in which a sampling guidance functions based on approximation models were developed. They claimed that PSP demonstrates considerable promises in efficiency, accuracy and robustness. Nain and Deb [84] combined an artificial neural network approximation with the NSGA-II to enable the use of GAs on computationally expensive problems and to achieve a computationally effective search and optimization procedure. Their objective is to enable the use of Gas on computationally expensive problems while retaining their basic robust search capabilities. Knowles [85] proposes an extended efficient global optimization algorithm (ParEGO) for MOO problems using the design and analysis of computer experiments (DACE) models which are sequentially updated in the iteration procedure. Kim and Chung [86] developed a multi-objective design optimization framework by combining GAs and Kriging approximation techniques and have tested it on wing platform design problem and demonstrated that the results obtained from the developed approach were efficient and applicable. Liu et al. [87] suggested a novel multi-objective optimization method based on an approximation model management technique.
Response surface approximations were used to predict the Pareto optimal sets and design of experiments DOE were used to generate the sample points. It is an iterative method in which the approximations gets updated in every single iteration by using of the trust region move limits updating strategy, which guarantees the accuracy of the approximations in the interest region of the design space where the actual Pareto optimal solutions may exist. The presented method focused on obtaining the actual Pareto optimal which makes it less dependent on the accuracy of the approximation models. Lim et al. investigated the use of surrogate models in evolutionary search. Recently, Jang et al. [88] employs adaptive approximation framework to resolve the computation burden of the full stochastic fatigue analysis in the optimization.

2.8 Design Optimization of Real Life Applications

Two real engineering design examples using metamodeling based optimization algorithms are presented. The first is the optimization of the design parameters of Automotive Magnetorheological Brake (MRB) System and the second is related to electric vehicles in which the energy conversion efficiency is to be maximized against powertrain component operation parameters using high fidelity model and simulation.

2.8.1 Optimal Design Parameters of Automotive Magnetorheological Brake System

Today hydraulic brakes (CHB) are used to provide required braking torque in stopping a vehicle. The CHB system consists of brake pedal, hydraulic fluid, transfer pipes and brake actuators (disk and drum brakes). When the driver presses on the brake pedal, the hydraulic brake fluid provides the pressure in the brake actuators that squeezes the brake pads onto the rotor. Unlike their conventional counterparts, electromechanical brake (EMB) systems offer potential improvements such as faster response, easy controller
implementation and parameter control, reduced weight and wiring, reduced maintenance etc, due to their purely electrical nature. Magnetic particle/fluid brake (or MRB) is one of the commonly known electromechanical brakes introduced as possible substitutes of the conventional hydraulic brake (CHB) systems.

MRB employs magnetorheological fluids (MRFs) that have controllable rheological characteristics through variation in magnetic field. MRFs are created by adding micron-sized iron particles to an appropriate carrier fluid such as oil, water or silicon. Their rheological behaviour is almost the same as that of the carrier when no external magnetic field is present. However, when exposed to a magnetic field, the iron particles acquire a dipole moment aligned with the applied magnetic field to form linear chains parallel to the field [89, 90]. This reversibly changes the liquid to solid-like that has a controllable yield strength, which its magnitude depends directly on the magnitude of the applied magnetic field.

The literature presents a number of MR fluid-based brake designs. In [91, 92], Carlson of Lord Corporation proposed and patented general purpose MRB actuators, which subsequently became commercially available [93]. In [94], an MRB design was proposed for exercise equipment (e.g., as a way to provide variable resistance). Recently, an MRB was designed and prototyped for a haptic application as well [95].

2.8.2 Design Optimization of High Efficiency EV/PHEV/EREV Electric Mode Operations

In recent years, hybrid vehicle powertrain has emerged as a promising technology that can help reduce petroleum energy consumption and green house gas (GHG) emissions. The powertrain of hybrid and electric vehicles uses one or multiple electric drives.
Depending on the vehicle configurations, propulsion from the electric drives either solely drives the vehicle or is combined with the power from internal combustion engine (ICE). High performance electric drives for vehicles are expected to be one of the key technologies for future cleaner and fuel efficient vehicles. Interest in advanced and alternative vehicle powertrain technologies continues to grow in response to concerns about continuing increases in the cost of oil and greenhouse gas emission rates. Because of complex nature of hybrid powertrain, developing a control system that can optimal manage multiple power plants and maximize vehicle energy efficient is a challenging task. Approaches that employ efficient control strategies could improve HEV’s efficiency at potentially minimal additional cost [96]. The control strategy is defined as the high level controller that decides in which operating mode to run according to the driver command, the driving conditions, the components’ characteristics and the feedback from the components in order to meet the torque, fuel consumption, emissions and drivability requirements [97].

HEVs use an energy storage system (normally a battery pack) coupled with an electric motor to absorb excess power and to filling the surge power needs, improving vehicle efficiency and emissions. The alternative power path provided by the battery and motor allows the engine to shut down when at a stop and during periods of otherwise inefficient low power operation. It also allows a smaller, more efficient engine to be used while retaining the vehicle’s peak power capability. To maintain the electrical energy needed to provide these functions, the battery is recharged by accepting excess power during efficient engine operation and by recapturing kinetic energy during regenerative braking to slow down the vehicle.
In response to the driving demand, the HEV controller selects how to split the draw of energy between the battery and the engine primarily to minimize fuel consumption. However, this fuel minimization goal must be pursued within the constraints imposed by the vehicle’s components, particularly the battery. To avoid costly battery replacements during the expected life of the vehicle, the batteries of today’s commercial HEVs are restricted to a narrow region of operation, avoiding the life-reducing impacts of operating at very high and low states of charge (SOCs) and cycling between them. Reaching a defined SOC constraint (and/or related battery current and voltage limits) during the operation of the vehicle will restrict the efficiency improvement benefit of the hybrid system. For instance, if the control strategy allows the battery to reach an upper limit on a particular cycle, the vehicle will no longer be able to capture and store regenerative braking energy. If the battery should reach a lower limit, the vehicle will no longer be able to operate in electric or electric-assist mode. These scenarios present an opportunity for a modified vehicle control strategy to use less fuel over the same cycle if it can use the hybrid functions of the vehicle in a way that avoids reaching such hard battery limits.

The U.S. Department of Energy (DOE) - has continuously supported research and development on HEVs and Plug-in Hybrid Electric Vehicle (PHEV) technologies, due to the potential fuel economy improvement that these technologies can offer. The DOE’s PHEV research and development plan [98], which is aiming at reducing the dependence on foreign oil by diversifying the sources for automobile fuels, describes the different activities required to achieve its goals. DOE’s Argonne National Laboratory’s (ANL) Powertrain System Analysis Toolkit (PSAT) [99] provides a useful tool to guide for the
design and evaluation of HEV/PHEV with various “primary electric” ranges, considering all-electric and charge-depleting strategies.

### 2.8.3 Optimization Algorithms Used in HEV Design

In recent years optimal EV/HEV/PHEV design is becoming a hot spot as a viable way to meet requirements on both vehicle performance and environment protection. Apparently, the coordination between the powertrain and control strategy has significant impacts on the operating performance of EVs. The EV powertrain, consisting of one or two electric motors, power battery and transmission, etc., is a complicated driving system with integrated mechanical, electrical devices. Hence, the coordination and control strategy for the components of a powertrain have a significant influence on dynamic performance, fuel economy, emissions, among other things [100]. To counteract this issue, many researchers have strived to find optimal solutions with various schemes. A number of studies have paid attention to optimizing powertrain size according to dynamic performance [101, 102]. Some researchers have applied traditional optimization methods to optimize powertrain size [103], but these methods require too many assumptions in the objective function including continuity, differentiability, satisfaction of the Lipshitz condition, etc. Therefore, a genetic algorithm was used to optimize the component size of the HEV powertrain [104, 105], the results showed the suitability and effectiveness of a genetic algorithm for this nonlinear optimization problem. However, the effect of control strategy parameters on vehicle performance was also ignored. This is not so important, though, as many researchers concentrate their research on the optimization of the control strategy parameters. Delprat et al. [106] and Antonio et al. [107] used optimal control theory and genetic algorithms respectively, to optimize the parameters of the control
strategy. Unfortunately, in their work the vehicle powertrain was fixed during the optimization of the control strategy.

Sampling based algorithms, such as DIRECT, were also used for the optimization of control strategy of electric vehicles. DIRECT requires no knowledge of the objective function gradient. The algorithm sets sample points of the domain, and uses the sample information to decide search direction. The algorithm will globally converge to the minimal value of the objective function. DIRECT is suitable for difficult optimization problems with bound constraints and a real-valued objective function, as the HEV power control problem. In Keith Wipke’s opinion, DIRECT is the best non-gradient based method to find the global optimum solution for HEV power control strategy [108].

Traditional optimization techniques such as linear programming and nonlinear programming are neither good nor efficient at dealing with hybrid electric vehicle design, considering the high complexity of HEV systems. Fortunately, the stochastic search techniques are particularly suitable for complex engineering design. For instance, such algorithms require little a priori knowledge about the search space. Furthermore, the stochastic nature of various evolutionary algorithms can prevent convergence upon local optima. Adaptive search and optimization algorithms include evolutionary algorithms, ant-colony, population-based incremental learning, simulated annealing (SA), Tabu search, particle swarm optimization (PSO), scatter search, and many more. Among them, evolutionary algorithm is a type of well-proved optimizer which is capable of seeking out the optimal solutions in an efficient fashion.

Wang in [109] discussed the optimal design of a series hybrid electric vehicle using adaptive stochastic strategy. He used a multi-objective optimization evolutionary
algorithm (MOEA) based on genetic algorithms (GA) to find the optimal parameters for the Series-HEV. He concluded that the proposed approach can be applied to parallel hybrids. Lin et al. [110] developed a design method for the power management control algorithm for HEV by using Markov chain modeling and stochastic dynamic programming (SDP) techniques. They compared their SDP simulation results with rule-based control strategy on different driving cycles and claimed that SDP performs better. Montazeri et al. [111] describes the application of GA for the optimization of control strategy in parallel HEV. Perez et al. [112] have implemented an algorithm based dynamic programming approach to find the split between two sources using a priori knowledge of the driving cycle. However, their solutions are suboptimal because of the treatment of the constraint on the consumed energy from the energy storage system ESS.

In this dissertation, an investigation was made to apply optimization based approaches to the determination of the optimal control parameters, in order to maximize the energy efficiency. Due to problem complexity, most existing global optimization methods such as GA and PSO cannot fully satisfy the design need. Therefore, the author used his recently introduced global optimization algorithm, SEUMRE, to carry out the search process to optimize the efficiency of 2-mode EV.

2.9 Identification and Use of Appropriate Sampling Techniques

Properly designed experiments are essential for exploring the properties of a physical system. In engineering, traditionally a single parameter is varied or perturbed and the effects are observed. Alternatively, combinations of factor settings are assigned either systematically (e.g. grid search) or randomly to provide alternatives for comparison. These experimental design techniques developed for physical experiments are now being
applied to the design of computer experiments to increase the efficiency of these analyses and simulations [65]. Sampling, as the first step in design experiments, is crucial in exploring the characteristics of the physical system or black-box computer analysis and simulation model, efficiently and effectively. In general, experimental designs can be classified into two categories: *classical* and *space filling* designs.

**2.9.1 Classical Designs**

Classical designs are used in various experiments designs. Full factorial design is its most basic form, in which every setting of every factor appears with every setting of every other factor. Classical designs include, *Central Composite Design* (CCD), alphabetical optimal designs, especially *D-optimal designs* [25]. In practice, it is desirable to use the smallest number of factor levels in an experimental design. One common class of such designs is the Box-Behnken designs. Sacks et al. [113] stated that in the presence of systematic rather than random error, a good experiment design tends to fill the design space rather than to concentrate on the boundary, and standard designs such as CCD and *D-Optimality designs* can be inefficient or even inappropriate for deterministic computer codes. Jin et al. [72] confirmed that experimental designs for deterministic computer analyses should be space filling.

**2.9.2 Space Filling Designs**

Space filling designs spread experiment points evenly throughout the design space, and largely take four different forms: *Orthogonal arrays* [60], various *Latin Hypercube designs* [61, 62], *Hammersly sequences* [63], and *uniform designs* [63, 64]. *Orthogonal arrays*, which are usually fractional factorial designs in two or three levels, are constructed to reduce the number of needed design points. *Hammersly sampling* is found
to provide better uniformity than Latin hypercube, while Latin hypercube performs worst due to its uniform sampling in 1-D projection. The appropriate sample size depends on the complexity of the function to be approximated. Increased sample points offer more information of the function at a higher expense. Wang in [114] found that for low-order functions, increasing the number of sample points after reaching a certain sample size does not contribute much to the approximation accuracy. Covering the entire design space with less sample points will reduce the objective and constraint function evaluations and lower computation cost. The present sampling schemes largely focus on the initial sampling to achieve certain space filling properties. Not knowing the shape of the functions as well as the desirable distribution and the number of needed sample points, these approaches are much less efficient than the seriously needed intelligent sampling techniques, in which minimum number of sample points are generated to effectively represent the unknown function using metamodeling techniques. Table 2-1 shows the characteristics of different sampling techniques and where they are used.

Table 2-1. Representative Sampling Techniques

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical Sampling</td>
<td></td>
<td>Stochastic methods; ARSM; MPS; AUMRE</td>
</tr>
<tr>
<td>Fractional Factorial</td>
<td>• Increasing exponentially with the increase of number of factors [25].</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Used when experiments are costly.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Used to screen for important factors [25].</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Typically placing points at regular intervals in the design space [34].</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Used for calibrating linear models in experimental settings where factors are relatively unconstrained in the region of interest.</td>
<td></td>
</tr>
<tr>
<td>Central Composites and Box-Behnken Designs</td>
<td>Central Composite Designs</td>
<td>Stochastic methods; ARSM; MPS; AUMRE</td>
</tr>
<tr>
<td></td>
<td>• Not suitable for high dimensional problems.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Recommended for designs with sequential experimentation since these designs can incorporate information from properly planned factorial experiments.</td>
<td></td>
</tr>
</tbody>
</table>
## Box Behnken designs
- Independent quadratic design that does not contain an embedded factorial or fractional factorial design[115],[123],[125].
- Rotatable.
- Requiring three levels of each factor.
- Requiring fewer treatment combinations than a CCD in cases involving three or four factors.
- Having limited capability for orthogonal blocking compared to the central composite designs.
- Used when performing non-sequential experiments (only planning to perform the experiment once).
- Allowing efficient estimation of the first- and second-order coefficients.
- Requiring fewer runs than CCDs.
- Used to calibrate full quadratic models.

## D-Optimal Designs
- Useful when classical designs do not apply.
- Design matrices are usually not orthogonal and effect estimates are correlated [116].
- Serving as an option regardless of the type of model the experimenter intends to fit.
- Particularly useful when resources are limited or constraints are on factor settings.

## Orthogonal Sampling (Orthogonal arrays)
- Simply fractional factorial design in two or three levels [25].
- Reducing the number of design points.
- Identical to Plackett-Burnam designs.
- Used for sampling deterministic computer experiments.
- Treating all region of design space equally.
- Efficient but more difficult to implement since all random samples must be generated simultaneously.
- Design space must be sampled evenly.
- Orthogonal sampling ensures that the ensemble of random numbers is a very good representative of the real variability.

## Space Filling Sampling
- Covering the design space in a uniform way.
- Costly in terms of computation time for high dimensional problems.
- Easy to generate and implement.
- Efficient for low dimensional problems (when n≤2).

## Uniform Designs
- Good for deterministic and stochastic computer experiments.
- Treating all regions equally and covers the design space in a good way [64].
- Considered as a good design in filling the design space.
- LHS, random sampling belongs to uniform designs.
- Particularly useful when identifying threshold or

## Stochastic methods
- ARSM; AUMRE; MPS

## Stochastic methods
- DIRECT; AUMRE
fitting nonparametric surfaces [117].
• Not requiring sampling orthogonally, but difficult to generate [118].

| **Latin Hypercube Sampling (LHS)** | Good for large scale problems [25].
| | Providing good uniformity and flexibility on the size of the sample.
| | Often applied in uncertainty analysis [61].
| | Independent - the sampling scheme does not require more samples for more dimensions (variables).
| | Random samples can be taken one at a time, remembering which samples were taken.
| | Ensuring ensemble of random numbers is representative of the real variability.
| | Can handle sampling where input variables have specified probability distribution.
| | Same point can be chosen multiple times.

| **Hammersly Sequence Sampling (HSS)** | Good for large scale problems.
| | Deterministic and generating a uniformly distributed and stochastic-looking sampling pattern, at low computational cost [63].
| | Efficient and robust.
| | Used extensively in many engineering applications.

| **Random Sampling** | New sample points are generated without taking into account the previously generated sample points.
| | Traditional random sampling (sometimes called brute force) is just an ensemble of random numbers without any guarantees.
| | Can handle irregular design spaces.
| | Easy to implement.

2.10 Identification and Use of Appropriate Metamodels

Approximation models, or metamodels, play a major role in the meta-models based global design optimization. The metamodel in simple, easy to calculate form, is used to replace the original, black-box computer analysis and simulation model. The introduced metamodel also provides an insight to the optimization problem by visualizing the interactions among design variables, objective functions and constraints. The overall objective is to reduce the computation cost of computationally intensive design simulations and analyses, using inexpensive surrogates of these analyses and simulations. 

Computational efficiency which refers to the computational effort required to construct
the metamodel and to predict response values for a set of new points using the metamodel and transparency which refers the capability for providing information concerning selection of different variables and interactions among these variables are two major concerns in generating these metamodels [119]. Modelling efficiency and accuracy are directly related to the design space, regardless of specific metamodelling techniques for a specific problem.

Many metamodelling techniques have been introduced for function approximation, but these methods show different accuracy, robustness, computational efficiency and transparency. An overview of several well known and representative metamodelling methods, including *Response Surfaces Model* (RSM), *Kriging* models, *Radial Basis Functions* (RBF) are discussed.

Figure 2-3 briefly describes the optimization process based metamodelling techniques.

![Figure 2-3. Optimization Based Metamodelling Process](image)

Figure 2-3. Optimization Based Metamodelling Process
2.11 Metamodels and Computer Simulations

Today’s engineering methods are strongly based on complex computer codes and numerical analyses (like nonlinear finite element analyses) which solely provide pointwise (discrete) information about the underlying relationship. Only in few applications, an analytic relationship can be established between input variables and output of a system under investigation. As a consequence, the solution of stochastic optimization problems requires many evaluations of the governing equations, for instance to compute the worst case or to evaluate the integral over the probability density. Especially for problems that can only be studied by means of time-consuming numerical analyses, stochastic optimization of the original problem becomes prohibitive. The permanent efforts in enhancing the underlying analysis codes and the increasing complexity of the structures under investigation countervail the steady enhancements in processor speed and computing power. Hence, it may not be expected that stochastic optimization problems will be easily manageable in the near future.

To reduce the computational burden, solution methods using global approximations are presented in this thesis. These global approximations are also termed metamodels or surrogate models since they are used as temporary substitution for the original code [27]. A metamodel replaces the true functional relationship \( y(x) \) by a mathematical expression \( \hat{y}(x) \) that is much cheaper to evaluate. Usually, an individual metamodel is established for each single response value \( y \). In general, the metamodel can be set up to depend on selected inputs and noise variables only omitting those variables with negligible or no impact on the selected response \( y \). To shorten the notation, the two different types of input variables (design variables and noise variables) are assembled to one input vector \( x \).
\[ x^T = [x_1, x_2, \ldots, x_n] \]

The individual components of \( x \) are subsequently addressed by \( x_i \) with \( i = 1, 2, \ldots, n \).

The metamodel concept is illustrated in Figure 2-3.

For the generation of a metamodel, an appropriate number of sampling points is needed. These points can be selected via design of experiments (DOE) techniques to gain the maximum number of information about the characteristics of the underlying relationship between input and output.

A suitable DOE technique must be carefully chosen since each type of surrogate model has need of different attributes with respect to the distribution of the sampling points.

Accordingly, diverse DOE techniques are also summarized and discussed in this dissertation. After selecting a compatible combination of global approximation method and DOE technique, the original simulation is performed for the designs appointed by the coordinates of the sampling points. With the information obtained from these computer experiments, the metamodel can be fit to provide an efficient estimate for the original function [120].

The main benefits of metamodels can be summarized as follows:

1. It is much cheaper to evaluate a metamodel than to perform a complex computer simulation. This yields a reduction in computational effort where many function evaluations are necessary (e.g. in optimization or stochastic analyses).
2. By the use of metamodels, the designer can easily explore the entire design space to get a more profound understanding of the system under investigation.
3. Metamodels can be used to combine information gathered from different sources, for instance analysis codes for different disciplines (e.g. fluids, structures, or thermo dynamical problems), or physical experiments and computer simulations.

4. Parallel computing is simple, since in general the individual sampling points are appointed simultaneously. Hence, the necessary computer experiments can be performed independently and in parallel.

5. Metamodels can be used to smooth response values if noise is present in the observations.

In the next sections, a selection of metamodeling approaches which were used in this work will be introduced. Comparisons of different metamodels formulations are also presented.

2.11.1 Response Surface Method

Response surface methods (RSM) have been used effectively for over thirty years as metamodels. Originally it was developed for the analysis of physical experiments [121]. Polynomial RSM have been used effectively for building approximations in a variety of applications. RSM approximates functions by using the least squares method on a series of points in the design variable space. Low order polynomials, such as the first and second order polynomials in Equations (2-1) and (2-2) are widely used as the response surface approximating functions.

\[ \hat{y}(x) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i \]  
(2-1)
\[
\hat{y}(x) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \sum_{j=1}^{k} \beta_{ij} x_i x_j
\]  
\text{(2-2)}

where, parameters, \(\hat{\beta}\), are computed using least squares regression by minimizing the sum of the squares of the deviations of predicted function values, \(\hat{y}(x)\), from the actual function values, \(y(x)\), using Equation (2-3).

\[
\hat{\beta} = (F^T F)^{-1} F^T \hat{y}
\]  
\text{(2-3)}

where \(F\) is the design matrix of sample data points, and \(\hat{y}\) contains the values of the response at each sample point. Polynomial response surface models can be easily constructed, and its smoothing capability allows quick convergence of noisy functions in the optimization. However, this over-simplification may be troublesome for modelling highly nonlinear or irregular behaviours [65]. Response surfaces have a variety of applications, including robust design, multidisciplinary optimization, adaptive strategies for global optimization, and manufacturing analysis.

### 2.11.2 Kriging Metamodels

Based on the work of Daniel G. Krige [122] addressing problems in geostatistics [123], kriging models are today a widespread global approximation technique. Their ability to exactly interpolate response values obtained at sampling points makes them particularly attractive for approximating deterministic simulations. Cressie first made the other scientists aware of the method, but it was not until later when a paper written by four statisticians on the topic that the true power of the method is realized [124]. The work has been considered as an attempt to link the relatively disjoint fields of statistical
analysis and deterministic computer experiments [125]. Sacks et al. [126] were the first to use this approach to model deterministic output of computer codes. Referring to the title of their contribution “design and analysis of computer experiments”, this model type is also called DACE model. A kriging model approximates the original relationship by

\[ y = f(x) + Z(x) + \epsilon \]  

(2-4)

where \( f(x) \) is a polynomial with free parameters \( \beta \) for the response surface approach. \( Z(x) \) represents the realization of a stationary, normally distributed Gaussian random process with mean zero, variance \( \sigma^2 \) and non-zero covariance. The term \( \epsilon \) describes solely the approximation error (bias) since random errors are excluded in this formulation.

The unknown parameters \( \beta \) and \( \sigma^2 \) can be estimated as

\[ \hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} \bar{y} \]  

(2-5)

\[ \hat{\sigma}^2 = \frac{1}{n} (\bar{y} - F \hat{\beta})^T R^{-1} (\bar{y} - F \hat{\beta}) \]  

(2-6)

The expression \( f(x) \) provides a global trend for the system behavior as in the standard response surface approach. The second part of the formulation \( Z(x) \) guarantees the interpolation of the observations \( \bar{y} \) at the sampling points \( x_i \) as it creates a “localized deviation” from the polynomial part of the model.

The Gaussian random process is characterized by the covariance matrix of \( Z(x) \) defined as

\[ \text{Cov}(Z(x^k), Z(x^l)) = \sigma^2 R ; \quad 1 \leq (k, l) \leq n \]  

(2-7)
with the correlation matrix

\[
R = \begin{bmatrix}
R(x_1, x_2) & \cdots & R(x_1, x_n) \\
\vdots & \ddots & \vdots \\
R(x_n, x_1) & \cdots & R(x_n, x_n)
\end{bmatrix}
\]  (2-8)

For kriging models, the predicted response values are assumed to be spatially correlated with the observations made at neighboring sampling points. Hence, the correlation function is expected to decrease with the distance to the sampling points. Suitable one-dimensional correlation functions are e.g. the Gaussian correlation function

\[
R(x_i, x_j) = \exp\left(-\sum_{k=1}^{p} \theta (x_i^k - x_j^k)^2\right)
\]  (2-9)

Using likelihood estimation, the unknown parameters of the correlation function have to be estimated. For the estimation of these parameters, the best linear unbiased prediction of the response is

\[
\hat{y} = \hat{\beta}^T f(x) + r^T(x)R^{-1}(\tilde{y} - F\hat{\beta})
\]  (2-10)

where \(r^T(x)\) is a vector representing the correlation between an unknown set of points \(x\) and all known experimental points.

\[
r(x) = [R(x, x_1), R(x, x_2), \ldots, R(x, x_n)]^T
\]  (2-11)

The second term in Equation (2-11), \(r^T(x)R^{-1}(\tilde{y} - F\hat{\beta})\), represents an interpolation of the residuals of the regression model \(\hat{\beta}^T f(x)\). All response data will thus be exactly predicted. Second order polynomial regression models are used in most of the work done in this dissertation. Gaussian correlation models are used due to their ability to control both the range of influence and the smoothness of the approximation function. Due to the
wide range of correlation functions available, Kriging methods can provide accurate predictions of highly nonlinear or irregular behaviours.

2.11.3 Radial Basis Function

Originally introduced by Hardy [127] and further improved by Dyn [128], Radial Basis function (RBF) is an effective algorithm for smoothing and interpolating experiment data. The form of the approximate model is a basis function of the Euclidean distance between the sampled data point and the point to be predicted. Developed as an analytical method for representing irregular surfaces, RBF uses linear combinations of radial symmetric functions of the Euclidean distance to build approximation models.

Mathematically, the model can be expressed as

$$\hat{y}(x) = \sum_{i=1}^{N} w_i \varphi(\|x - c_i\|)$$  \hspace{1cm} (2-12)

where the approximated function $\hat{y}(x)$ is represented as a sum of $N$ radial basis functions $\varphi$, each associated with a difference center, $c_i$, and weighted by an appropriate coefficient, $w_i$. RBF approximation is capable of producing good fits to arbitrary contours of both deterministic and stochastic response functions [129].

The radial function $\varphi(r)$ can take many forms as shown in Table 2-2.

<table>
<thead>
<tr>
<th>Function Type</th>
<th>Radial Basis Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Function</td>
<td>$\varphi(r) = r$</td>
</tr>
<tr>
<td>Cubic Function</td>
<td>$\varphi(r) = r^3$</td>
</tr>
<tr>
<td>Thin Plate Function</td>
<td>$\varphi(r) = r^2 \log r$</td>
</tr>
<tr>
<td>Multiquadratic Function</td>
<td>$\varphi(r) = \begin{cases} \sqrt{r^2 + a^2}, &amp; r &gt; 0 \ 0, &amp; r = 0 \end{cases}$</td>
</tr>
<tr>
<td>Gaussian Function</td>
<td>$\varphi(r) = e^{-ar^2}$</td>
</tr>
</tbody>
</table>
The method has been used successfully in many engineering applications, including ocean depth measurement, altitude measurement, rainfall interpolation, surveying, mapping, geographic and geology, image warping, and medical imaging. A summary of the characteristics of the previously mentioned and other metamodels are presented in Table 2-3.

Table 2-3. Representative Metamodeling Techniques

<table>
<thead>
<tr>
<th>Metamodels</th>
<th>Characteristics</th>
<th>Method</th>
</tr>
</thead>
</table>
| **Response Surface Method (RSM)** | • Easy to understand and implement.  
• Works well for applications with few design variables.  
• Best suited for applications with random errors [65].  
• Performing well when approximating low-order (Polynomial) nonlinearity problems.  
• Able to produce efficient and accurate solutions to many engineering problems.  
• Possible to identify the significance of different design factors directly from the coefficients in the normalized regression mode in Polynomial Regression (PR)[72].  
• For multidimensional problems, important to use linear or second-order polynomial models to narrow the design variables to the most critical ones.  
• Might be instable when higher-order polynomials is used [130].  
• May be too difficult to take sufficient sample data to estimate all of the coefficients in the polynomial equation, particularly in large dimensions [72]. | Deterministic and Stochastic methods |
| **Kriging** | • Accurate for nonlinear models.  
• Extremely flexible due to the wide range of the correlation function but complex to construct.  
• Well suited for deterministic applications.  
• Good for applications with \( \leq 50 \) factors [65].  
• Providing a basis for a stepwise algorithm to determine the important factors.  
• Major disadvantage - model construction can be very time-consuming [72].  
• Complexity and lack of commercial software may hinder this technique from being popular. | Deterministic and Stochastic methods; ARSM; AUMRE; MPS |
| **Radial Basis Function (RBF)** | • Superior to thin plate splines [129].  
• Producing good fits to arbitrary contours of both deterministic and stochastic response functions [129].  
• Successfully used in many engineering design applications [65].  
• Based on interpolation [131].  
• Effective metamodels for electronic circuit’s simulation models [129]. | Deterministic methods; Stochastic methods; ARSM; AUMRE; MPS |
• Restricted to a small region like Kriging.
• Most appropriate to approximate scattered data in several dimensions.
• Used to approximate multivariate functions.
• Highly efficient in practice because of their easily adjustable smoothness and their powerful convergence properties.
• Very suitable to so-called learning by neural networks.

<table>
<thead>
<tr>
<th>Multivariate Adaptive Regression Splines (MARS)</th>
<th>Deterministic methods; Stochastic methods; ARSM; AUMRE; MPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Very flexible as it can adapt to any functional form.</td>
<td></td>
</tr>
<tr>
<td>• Providing better results for sparse inputs [72].</td>
<td></td>
</tr>
<tr>
<td>• Relatively new compared to the other techniques [72].</td>
<td></td>
</tr>
<tr>
<td>• High accuracy and major reduction in computational cost associated with constructing the metamodel [72].</td>
<td></td>
</tr>
<tr>
<td>• Attempting to approximate complex relationships by a series of linear regressions on different intervals of the independent variable ranges or sub-regions of the independent variable space.</td>
<td></td>
</tr>
<tr>
<td>• Performing well when large sample sizes are given.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Artificial Neural Networks (ANN)</th>
<th>Deterministic and Stochastic methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Excellent for high dimensional problems [132].</td>
<td></td>
</tr>
<tr>
<td>• Limited applications in expensive black-box functions.</td>
<td></td>
</tr>
<tr>
<td>• Best suited for approximating functions in regression-type applications [65].</td>
<td></td>
</tr>
<tr>
<td>• Eliminating the possibility of pre-selecting an incorrect functional form, due to its ability to universally model any relationship through a neural network with non linear transfer function [72].</td>
<td></td>
</tr>
<tr>
<td>• Not responsive to deviations from traditional statistic model hypothesis</td>
<td></td>
</tr>
<tr>
<td>• Can model a combination of continuous and discrete numerical variables.</td>
<td></td>
</tr>
<tr>
<td>• Most paradigms are global models [133].</td>
<td></td>
</tr>
<tr>
<td>• Accuracy and precision is based on the quality and quantity of the data set used in modelling.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Learning Inductive (LI)</th>
<th>Decision Trees; Bayes Nets; Machine Learning Algorithms (SVM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Most appropriate when input and output factors are primarily discrete valued or grouped.</td>
<td></td>
</tr>
<tr>
<td>• Forming the model as rules or decision tree.</td>
<td></td>
</tr>
<tr>
<td>• Requiring much data to train the system.</td>
<td></td>
</tr>
<tr>
<td>• Not the best for engineering applications and having limited applications associated with expensive black-box functions.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NURBS</th>
<th>GA; SA; PSO; ACO Genetic Programming; Cultural Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Generally following the shape of the governing control point network.</td>
<td></td>
</tr>
<tr>
<td>• Not oscillating about any straight line more often than the defining polygon NURBs curves.</td>
<td></td>
</tr>
<tr>
<td>• Invariant with respect to affine transformations.</td>
<td></td>
</tr>
<tr>
<td>• Hyper Models are based on NURBs.</td>
<td></td>
</tr>
</tbody>
</table>

Selecting the appropriate metamodel for the problem of interest makes a huge difference. There is no specific metamodel that can be recommended. It depends on the
shape and nature of the objective function. Some metamodels might perform well with specific optimization problems but might not do well with other optimization problems. In this dissertation we have used many of the previously mentioned metamodels. Every metamodel has its own advantages and disadvantages. The author of this dissertation tested many benchmark test problems using different metamodels and reported the results in [134].

2.12 Statistical Validation Methods

Validating the approximation models is a crucial step in metamodeling. What is meant by validation here is how good is the approximate model in imitating the expensive or black-box function. In other words, how accurate is the approximation model. Metamodels should be validated before being used as approximate models for computation intensive processes [135]. There are many statistical validation techniques introduced in the last few decades and still being used by many researchers. The accuracy of the approximation models can be estimated by using these statistical methods. *Root mean square error* (RMSE), *relative maximum absolute error* (RMAE) and *R square* are among these methods. The equations for these methods are presented here.

2.12.1 Root Mean Square Error (RMSE)

Equation (2-13) represents the mathematical formula for calculating RMSE

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}
\]

The smaller the value of RMSE, the more accurate the metamodel is.
2.12.2 Relative Maximum Absolute Error (RMAE)

Equation (2-14) represents the mathematical formula for calculating RMAE

\[
RMAE = \frac{\max(|y_1 - \hat{y}_1|, |y_2 - \hat{y}_2|, \ldots, |y_n - \hat{y}_n|)}{\sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2}}
\]

(2-14)

2.12.3 R-Square

Equation (2-15) represents the mathematical formula for calculating R-square

\[
R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}
\]

(2-15)

where \(\hat{y}_i\) is the corresponding predicted value for the observed value \(y_i\); \(\bar{y}\) is the mean of the observed values. The larger the value of R Square, the more accurate the metamodel is. In this dissertation RMSE is used to verify and validate the accuracy of the constructed metamodels.

2.13 Multiple Objectives

Real world engineering design problems are usually characterized by the presence of many conflicting objectives. Therefore, it is obvious to look at the engineering design problems as multi-objective optimization problems (MOP). MOP can be found in various fields: product and process design, finance, aircraft design, the oil and gas industry, automobile design, or wherever optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Multi-objective optimization can be defined as the process of simultaneously optimizing two or more conflicting objectives subject to certain constraints. Maximizing performance and minimizing fuel consumption of a vehicle; and minimizing weight while maximizing the strength of a particular
component are examples of multi-objective optimization problems. There is no acceptable definition for the optimum as in single objective optimization. The decision is completely left to the decision maker to decide what the best is. In multi-objective optimization, several conflicting and non-commensurate objective (criterion) functions have to be optimized over a feasible set determined by constraint functions. Due to the conflicting nature of the criteria, a unique feasible solution optimizing all the criteria does not exist. Based on the commonly used Pareto concept of optimality, one has to deal with a rather large or infinite number of efficient solutions. Two different efficient solutions are characterized by the fact that each of them is better in one criterion but worse in another. The fact that improvement of one criterion results in a loss in another is known as the trade-off between the solutions.

The primary goal of multi-objective optimization is to seek efficient solutions and/or Pareto outcomes of multi-objective programs (MOPs), and if possible, supports the decision maker to choose a final preferred solution. It is therefore of interest to design methods for obtaining a complete (or partial) description of the Pareto and efficient sets referred to as the solution sets. Unlike single objective optimization, the solution to this problem is not a single point, but a family of points known as the Pareto-optimal set. Each point in this surface is optimal in the sense that no improvement can be achieved in one cost vector component that does not lead to a degradation in at least one of the remaining components.

Since the exact solution set is very often not attainable, an approximated description of the solution set becomes an appealing alternative. Many approximating approaches have been developed to represent the solution set when this set is numerically available.
2.14 Test Benchmark Problems and Industrial Design Case Studies

To verify the performance of optimization algorithms, various benchmark problems are used. Unfortunately there are no mature benchmarks for metamodel based optimization algorithms. In this work, a number of well-known benchmark functions that are normally used to test conventional, unimodal and multimodal optimization search algorithms are tested against classical global optimization methods, to illustrate the capability of the newly introduced algorithms. These benchmark functions are widely used in testing new global optimization search algorithms today to fill in the vacuum within the research community. Every benchmark problem has its own features that make it difficult and quite challenging for an algorithm to reach its global optimum. To demonstrate the special capability of the proposed search methods, two practical industrial design optimization examples, employing “black-box” analysis and simulation are also used. Optimization results and computation time using the newly proposed algorithms and well know global optimization methods, such as GA and SA, are also presented.

The benchmark tests are used to compare the performance of different approaches, and to find out how efficient, effective, and robust these methods are. Some algorithms might suffer in converging to a global optimum with some benchmark problems, but they might succeed with other benchmark problems. The more and different benchmark problems are used in testing the algorithm, the better. These benchmark problems are different in terms of their shape of objective function, dimension (number of design variables), and complexity. Low and high dimensional, as well as unconstrained and constrained benchmark problems are used in these tests, including Alpine Function, Banana Function, Beak Function, Goldstein and Price Function, Branin Function, Schaffer’s Function,
Griewank Function, Generalized Polynomial Function, Shubert Function, Six-hump Camel-back Function (SC), Levy Function, Shekel Function, Sphere function, Hartman functions with 6 and 16 design variables (H6 & H16) and several others. Also included are nonlinear objective function with inequality ($C_1$) and equality ($C_2$) constraints and the tension compression string problem. The detailed form of these benchmark test problems are given in Appendix A.

All these benchmark problems are continuous and gradient based search methods can used to solve these problems more efficiently. At present not appropriate and efficient discrete and discontinuous benchmark problems are available to test the non-gradient based optimization algorithms introduced in this work.

These benchmark problems consist of three major categories: traditional optimization benchmark test problems; high dimensional benchmark problems; and black-box function based design optimization problem. Usually, global optimization algorithms suffer from converging to global solutions if the number of the dimensions is high. The optimization algorithms introduced in this search are tested using high dimensional benchmark problems and the results obtained are promising. Industrial design case studies were used to test and reveal the pros and cons of the proposed algorithms and the efficiency of the new algorithms in dealing with real life design applications. The engineering design problems include the design optimization of the magnetorheological brake system and the optimization of the control strategy of electric vehicles to reach the maximum vehicle efficiency and minimum emissions.

In addition, to validate the newly introduced algorithms, the results obtained from the algorithms proposed in this research are compared with two well known and proven
global optimization search algorithms SA and GA. Their results and computation time
are included for sake of comparison.

In this dissertation, an extensive review and benchmark testing on the state of the art
metamodel-based global optimization techniques and multidisciplinary optimization
search algorithms was carried out.
Chapter 3. Approximated Unimodal Region Elimination for Global Design Optimization

3.1 Introduction

Computer analysis and simulation based design optimization requires more computationally efficient global optimization tools. In computation intensive optimization problems such as black-box functions global optimization techniques suffer and might fail in converging to global solutions within a specific limit of time or resources. Problems that are computationally demanding can be found in many fields. They can be found in engineering, economics, mathematics, health sciences and so on. That draw the researcher’s attention and made them aware of the importance of developing and introducing global optimization methods that are efficient and capable in solving complex optimization problems. Replacing the black-box functions with metamodels is a promising alternative in alleviating the computation burden. Since metamodels are found to be very promising in handling computationally intensive optimization problems, low order polynomial response surface approximation model are adapted in the proposed approach which will be introduced in this chapter. RSM are easy to understand and easy to construct metamodels. In the proposed approach the search is being conducted and focused on the most promising regions within the design space.
In this chapter, a new global optimization algorithm based on design experiments, region elimination and response surface model, namely Approximated Unimodal Region Elimination (AUMRE) method, is introduced. The approach divides the field of interest into several unimodal regions using design experiment data; identifies and ranks the regions that most likely contain the global minimum; forms a response surface model using additional design experiment data over the most promising region; identifies its minimum, removes this processed region, and moves to the next most promising region. By avoiding redundant searches, the approach identifies the global optimum with reduced number of objective function evaluations and computation effort. The new algorithm was tested using a variety of benchmark global optimization problems and compared with several widely used global optimization algorithms. The experiment results present comparable search accuracy and superior computation efficiency, making the new algorithm an ideal tool for computer analysis and simulation based global design optimization.

3.2 Generic Global Optimization Problem

In the multidisciplinary design optimization, the objective function of the optimization is evaluated through complex, black-box computer analysis and simulation. These evaluations lead to a group of experiment data

\[ S \triangleq \{(x_i, f(x_i)) : x_i \in A\} \tag{3-1} \]

where \( x_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \), and \( n \) is the dimension of the design problem under study; \( x_i = (i = 1, \ldots, n) \) is the \( ith \) value of the design variables in the \( n \)-dimensional space with \( n \) vector components; \( y_i = f(x_i) \) is the value of the objective function for
point \( x_i \); \( S \) is a set of experiment data, and \( A \) denotes the field of interest or the feasible region of the design optimization problem. If \( n=2 \), the field of interest and several points of the design variable \( (x_i) \) can then be illustrated as shown in Figure 3-1.

Note that \( S \) forms our design space, and several design points appear in the design space randomly. In addition, the objective function \( f(x_i) \) is normally expressed in an implicit form, multimodal, and expensive to calculate since complex computer analysis and simulation are needed to obtain its value. The objective of this work is to identify the global optimum of \( f(x_i) \) with a small or considerably reduced number of these black-box numerical function evaluations.

![Figure 3-1. Field of Interest and Points of Design Variable](image)

### 3.3 Major Steps of the Proposed Algorithm

The proposed search algorithm consists of the following major steps:

1) Generate a set of DOE data points, \( x_i = (x_{i1}, x_{i2}, \ldots, x_{in})(i = 1, \ldots, m) \) over the field of interest;

2) Divide the field of interest \( S \) into many unimodal regions; in the subsequent search of the local minimum, the identified unimodal area may overlap with
previously examined area. For the ease of search algorithm, these minor overlaps that cause repeated search over small areas are ignored.

3) Identify the region that most likely contains the global minimum, and rank the others, based on the smallest value of the objective function within these regions;

4) Refine the most promising region by adding more experiment data points within the region, and carry out an RSM approximation over the region;

5) Conduct optimization in the most promising unimodal region to obtain the minimum of the region;

6) Add the obtained minimum as a new data point, remove the furthest away point from this approximated minimum, carry out a new RSM approximation, and go to step 5) until a local minimum has been identified over this most promising unimodal region;

7) Remove the processed region from the field of interest;

8) Repeat the previous steps (1-7) until the local optima of all promising unimodal regions are located, and the global optimum is identified from these local optima.

3.4 Metamodeling for the Divided Region

Properly designed experiments are essential for effective data sampling. In engineering, traditionally a single parameter is varied (perturbed) and its effects on the objective function are observed. Alternatively, combinations of factor settings are assigned either systematically (e.g. grid search) or randomly to provide an alternative for comparison. Experimental design techniques which were developed for physical experiments are being applied to the design of computer experiments to increase the efficiency of these analyses [65].
In this work, due to the implicit form of the black-box computer analysis and animation tools, a systematic collection and evaluation on the values of the objective function in the global design optimization are carried out using DOE method. RSM is used for solving a complex optimization problem through approximation, in which the regression model is used to fit a series of planned experiments to estimate the complex relationship between design variables and objective functions.

Suppose that we are concerned with a system involving a vector $y$ of responses which depends on the given input variables $x_1, x_2, \cdots, x_n$ and their relations can be formulated as

$$y = f(x_1, x_2, \cdots, x_n)$$  \hspace{1cm} (3-2)

where the form of $f(x_1, x_2, \cdots, x_n)$ is unknown and perhaps very complicated. The response surface method assumed that $f$ can be approximated by a low order polynomial function. For quadratic approximation, the second order response surface model has the form:

$$\hat{y}(x) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \sum_{j=1}^{k} \beta_{ij} x_i x_j + \epsilon$$  \hspace{1cm} (3-3)

where $\epsilon$ is the disturbance or model error, and the estimators $\beta_0, \beta_i$ and $\beta_{ij}$ are determined using the least squares regression analysis by fitting the response surface to experiment data.

The main aim in the formation of the RSM is to fit a response surface as closely as possible. The coefficients of the response surface using the least square method are

$$\hat{\beta} = (F^T F)^{-1} F^T \hat{y}$$  \hspace{1cm} (3-4)
where $F$ denotes the design matrix comprising the experimental points, and $\hat{y}$ represents the response vector obtained from the performance function corresponding to the experimental points [136]. The use of RSM has many advantages. The method works very well for unimodal functions, as it reduces the number of function evaluations, and considerably increases the computation efficiency.

### 3.5 Region Elimination Algorithm

The following are the algorithm steps that explain in details how the proposed algorithm works.

**Step 1:** introduce a finite number of grids in the space of interest, and obtain a set of design experiment data, $f(x_i)(i = 1, \cdots, m)$ using DOE method on the selected grid points; set unimodal region counter $l=1$.

**Step 2:** for unimodal region, $l=1$ carry out the following

**Step 2.1:** find the minimum function value and its coordinates

$$y^l = \min\{f(x_i): (x_i, f(x_i)) \in S\} \tag{3-5}$$

and denote

$$x^{(l)} = (x_1^{(l)}, x_2^{(l)}, \cdots, x_n^{(l)}); f(x^l) = y^{(l)} \tag{3-6}$$

If there are several $x$’s corresponding to $y^{(l)}$, just choose one.

Put $(x^{(l)}, f(x^{(l)}))$ into $C_l$, and $x^{(l)}$ into $C_l^x$ and carry out the following partition:

$$S = \bigcup_{i=1} C_l \tag{3-7}$$

If $f$ is continuous and the number of elements in $C_l^x$ is greater than 2, then in the corresponding area of $C_l^x$ exists at least one minimum. At each round, these partitions can
be ranked based on their possibility to contain the global minimum from high to low as $C^x_1, C^x_2, \cdots C^x_l$

**Step 2.2:** Use the coordinates of the minimum function value $x^l_k (k = 1: n)$ as the centre point to start the search, and assign the function value at this centre point as $q = y^{(l)}$.

**Step 2.3:** Identify the turning points (when $f(x) < q$, go back in the opposite direction) of the unimodal region through a search for the next point $x$ in $S$ by moving along the positive and the negative directions of the coordinates (e.g. moving right, left, up, and down if $n = 2$), comparing the function value at each point with the previous function value.

If $f(x) \geq q$, put $(x, f(x))$ into $C_i$ and $x$ into $C^x_i$, and let $q = f(x)$; go to Step 2.3 for the next design space coordinate.

**Step 2.4:** Subtract the identified unimodal region $C_i$ from the design space $S$ until all of the points in the design space $S$ are explored. Specifically, let $S = S \setminus C_i$. If $S \neq \emptyset$, let $l = l + 1$, and go to Step 2.

**Step 3:** After exploring the entire space of interest $S$, and obtaining divided regions $C^x_i$ by dividing $S$ into $C^x_i$, generate a field $A_i$ which covers the area of $C^x_i (i = 1, 2, \cdots)$. Consider the first several $C^x_i$, which are most likely to contain the global minimum

$$l_i = \min\{x^l_i: (x^l_1, \cdots, x^l_i, \cdots x^l_n) \in C^x_i\} \quad (3-8)$$

$$r_i = \max\{x^l_i: (x^l_1, \cdots, x^l_i, \cdots x^l_n) \in C^x_i\} \quad (3-9)$$

where $l_i$ and $r_i$ are the minimum and the maximum values of the points in $C^x_i$, which can be determined using the relations given previously.
\[ A_i = [l_i, r_i] \times \cdots \times [l_n, r_n] \tag{3-10} \]

Let

\[ A = A_i \cap A \tag{3-11} \]

Carry out new design experiments in \( A \), and denote the set of experiment data contained in \( A \) as \( S \), repeat Step 2.

Terminate the process when the sub region \( A \) can no longer be changed.

**Step 4:** For \( C_i^x \), denote

\[ \alpha = \min \{|x_i^- - x_i^1|, |x_i^+ - x_i^1|, i = 1, \ldots, n\} \tag{3-12} \]

where

\[ x_i^- = \min \{x_i : x_i \text{ are the } i\text{th coordinate of } x, x \in C_i^x\} \tag{3-13} \]

and

\[ x_i^+ = \max \{x_i : x_i \text{ are the } i\text{th coordinate of } x, x \in C_i^x\} \tag{3-14} \]

let

\[ x_{ij} = x^{(i)} \pm \frac{\alpha}{2}(e_i \pm e_j) \tag{3-15} \]

where \( e_i(i = 1, \ldots, n) \) is a unit vector of the \( i\)th coordinate direction in \( R \).

If \( f(x_{ij}) \geq f(x^{(i)}) \), next else end.

Let \( \alpha = 0.5\alpha \), generate new sample points in the identified unimodal area and repeat the previous step until \( \alpha \leq \varepsilon \) (\( \varepsilon \) is a pre-specified small positive number defined by the designer).
For a two dimensional problem, \( n = 2 \), the process is illustrated in Figure 3-2. The example shows how a convex and unimodal region is obtained by from a large feasible area. Since we have the centre point, \( x_i \), which represents roughly the minimum point in that sub-region. By obtaining \( x_i^- \) and \( x_i^+ \) we can find the initial value of \( \alpha \) using \( \alpha = \min\{|x_i^- - x_i^+|, |x_i^+ - x_i^+|, i = 1, \ldots, n\} \). By drawing a circle (since \( n = 2 \)) with radius \( \alpha \), one can obtain the four extreme coordinate points, two in each dimension, as the circle intersects with the axes of the coordinates, represented as \( X_1, X_2, X_3 \) and \( X_4 \). It is desirable to have more data points inside the circle so that the approximated fitted model will be more accurate. More points can be obtained using the equation for calculating \( x_{ij} \) as previously mentioned. These are the middle points on the straight lines that connects \( X_1, X_2, X_3, \) and \( X_4 \), as shown in Figure 3-2. By evaluating the function values at all of these points and compare them with the function value at the centre point, a decision can be made whether further reduction of \( \alpha \) and repeat the same procedure are needed, or the current value of \( \alpha \) can be accepted. For a problem with dimension \( n > 2 \), \( n \) is the radius of an \( n \)-D spherical region.

![Figure 3-2. Reduction of Identified Unimodal Area](image)
Step 5: Create spheres with radius $\alpha$ that hold the unimodal region inside and use RSM to construct the response surface and find its local minimum using local optimization methods. Specifically, denote

$$Sp_i^x = \{x: \|x - x^i\| \leq \alpha\} \quad (3-16)$$

find the local minimum in each sphere $Sp_i^x$ using the second order response surface model.

Step 6: Find the absolute minimum $f^*(x^*)$ among the local minimum $f_{c_i}^*(x^*)$ and identify it as the global minimum of the optimization problem.

The flow diagram of the proposed algorithm (AUMRE) is shown in Figure 3-3.

3.6 Testing Using Benchmark Problems

To access the performance of the newly introduced AUMRE algorithm, in terms of its ability to provide correct search results and computation efficiency, the method has been tested using a variety of commonly used benchmark optimization problems. Furthermore, its performance is compared with other well-known global optimization and objective-oriented sequential sampling search methods, including SA, GA, PSO, DIRECT and MPS. The analytical global optimum point (coordinates) and its objective function value of all tested benchmark problem in this dissertation are reported in Appendix A. The calculated results are presented in this section.
3.6.1 Benchmark Test 1 - Alpine Function

First the Alpine function is used as the objective function of the optimization in the tests as it contains many local minima and one distinct global minimum. The function given by Equation (3-17) is illustrated in Figure 3-4. The results are presented in Table 3-1.
\[ f(x_1, \ldots, x_D) = \sin x_1 \times \ldots \times \sin x_D \sqrt{x_1 \cdots x_D} \]  
\[ (x_1, \ldots, x_D) \in [0, x_{\text{max}}]^D \]

Figure 3-4. The Objective Function of Benchmark Test 1

Table 3-1. Test Results on Alpine function

<table>
<thead>
<tr>
<th>Algorithm Used</th>
<th>Calculated Optimum, ( X^* ) and ( f(X^*) )</th>
<th>Number of Objective Function Evaluations</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical Minimum</td>
<td>( x_1 = 7.9171 ), ( x_2 = 4.8151 ), ( f(x_1, x_2) = -6.1295 )</td>
<td>125</td>
<td>19.738</td>
</tr>
<tr>
<td>PSO</td>
<td>( x_1 = 7.9180 ), ( x_2 = 4.8046 ), ( f(x_1, x_2) = -6.1291 )</td>
<td>125</td>
<td>13.649</td>
</tr>
<tr>
<td>GA</td>
<td>( x_1 = 7.9040 ), ( x_2 = 4.8151 ), ( f(x_1, x_2) = -6.1295 )</td>
<td>1000</td>
<td>13.649</td>
</tr>
<tr>
<td>MPS</td>
<td>( x_1 = 7.9040 ), ( x_2 = 4.8356 ), ( f(x_1, x_2) = -6.1277 )</td>
<td>45</td>
<td>20.39</td>
</tr>
<tr>
<td>SA</td>
<td>( x_1 = 7.9171 ), ( x_2 = 4.8158 ), ( f(x_1, x_2) = -6.1295 )</td>
<td>1391</td>
<td>12.10</td>
</tr>
<tr>
<td>DIRECT</td>
<td>( x_1 = 7.9218 ), ( x_2 = 4.8171 ), ( f(x_1, x_2) = -6.1294 )</td>
<td>267</td>
<td>10.02</td>
</tr>
<tr>
<td>AUMRE</td>
<td>( x_1 = 7.908 ), ( x_2 = 4.8244 ), ( f(x_1, x_2) = -6.1290 )</td>
<td>125</td>
<td>7.92</td>
</tr>
</tbody>
</table>
3.6.2 Benchmark Test 2 - Banana Function

Rosenbrock’s valley or Banana function is a classic test problem for optimization algorithms due to its challenge on the convergence and robustness of the algorithm. The global minimum is inside a long, narrow, parabolic shaped flat valley, and converges to the solution at point (1, 1) is well known to be difficult. The Banana function given by Equation (3-18) is illustrated in Figure 3-5. The results are presented in Table 3-2.

\[
f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2
\]  

(3-18)

![Banana Function](image)

Figure 3-5. The Objective Function of Benchmark Test 2

Table 3-2. Test Results on Rosenbrock’s Valley or Banana Function

<table>
<thead>
<tr>
<th>Algorithm Used</th>
<th>Calculated Optimum, X* and f(X*)</th>
<th>Number of Objective Function Evaluations</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical Minimum</td>
<td>(x_1: 1.0000, x_2: 1.0000, f(x_1, x_2): 0.0000)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSO</td>
<td>(x_1: 0.9653, x_2: 0.9308, f(x_1, x_2): 0.0013)</td>
<td>146</td>
<td>14.79</td>
</tr>
<tr>
<td>GA</td>
<td>(x_1: 1.0796, x_2: 1.1656, f(x_1, x_2): 0.0063)</td>
<td>1000</td>
<td>10.36</td>
</tr>
<tr>
<td>MPS</td>
<td>(x_1: 1.0244, x_2: 1.0448, f(x_1, x_2): 0.0027)</td>
<td>227</td>
<td>58.14</td>
</tr>
<tr>
<td>SA</td>
<td>(x_1: 0.9531, x_2: 0.9085, f(x_1, x_2): 0.0022)</td>
<td>2153</td>
<td>31.95</td>
</tr>
<tr>
<td>DIRECT</td>
<td>(x_1: 0.9931, x_2: 0.9858, f(x_1, x_2): 0.0001)</td>
<td>1347</td>
<td>19.01</td>
</tr>
<tr>
<td>AUMRE</td>
<td>(x_1: 1.0476, x_2: 1.1000, f(x_1, x_2): 0.0029)</td>
<td>381</td>
<td>10.23</td>
</tr>
</tbody>
</table>
3.6.3 Benchmark Test 3 - Beak Function

The Beak function is used as a benchmark problem due to its challenge to find its global minimum. The Beak function given by Equation (3-19) is illustrated in Figure 3-6. The results are presented in Table 3-3.

\[
f(x_1, x_2) = 3(1 - x_1)^2 e^{(-x_1^2 - (x_2 + 1)^2)} - 10 \left(\frac{x_1}{5} - x_1^3 - x_2^5\right) e^{(-x_1^2 - x_2^2)} - \frac{1}{3} e^{(-(x_1 + 1)^2 - x_2^2)}
\]

(3-19)

Figure 3-6. The Objective Function of Benchmark Test 3

Table 3-3. Test Results on the Beak Function

<table>
<thead>
<tr>
<th>Algorithm Used</th>
<th>Calculated Optimum, (X^<em>) and (f(X^</em>))</th>
<th>Number of Objective Function Evaluations</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical Minimum</td>
<td>0.2283, -1.6255, -6.5511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSO</td>
<td>0.2283, -1.6255, -6.5511</td>
<td>241</td>
<td>38.15</td>
</tr>
<tr>
<td>GA</td>
<td>0.2303, -1.6261, -6.5511</td>
<td>1000</td>
<td>23.92</td>
</tr>
<tr>
<td>MPS</td>
<td>0.2398, -1.614, -6.5486</td>
<td>34</td>
<td>16.63</td>
</tr>
<tr>
<td>SA</td>
<td>0.2283, -1.6255, -6.5511</td>
<td>2524</td>
<td>19.39</td>
</tr>
<tr>
<td>DIRECT</td>
<td>0.2222, -1.7778, -6.2407</td>
<td>195</td>
<td>9.63</td>
</tr>
<tr>
<td>AUMRE</td>
<td>0.2467, -1.7863, -6.1928</td>
<td>136</td>
<td>9.57</td>
</tr>
</tbody>
</table>
3.6.4 Benchmark Test 4 - Goldstein and Price Function (GP)

Goldstein and Price function is often used as a benchmark function due to the difficulty to find its global minimum; the GP function given by Equation (3-20), is illustrated in Figure 3-7. The results are presented in Table 3-4.

\[
f(x_1, x_2) = \frac{[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]}{\times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]}
\]

(3-20)

Figure 3-7. The Objective Function of Benchmark Test 4

Table 3-4. Test Results on Goldstein and Price Function

<table>
<thead>
<tr>
<th>Algorithm Used</th>
<th>Calculated Optimum, X* and f(X*)</th>
<th>Number of Objective Function Evaluations</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Analytical Minimum</strong></td>
<td>0.0000  -1.0000  3.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSO</td>
<td>-0.0003  -0.9999  3.0000</td>
<td>125</td>
<td>19.83</td>
</tr>
<tr>
<td>GA</td>
<td>0.0005   -0.9999  3.0001</td>
<td>1000</td>
<td>11.84</td>
</tr>
<tr>
<td>MPS</td>
<td>-0.0004  -1.0006  3.0000</td>
<td>227</td>
<td>33.21</td>
</tr>
<tr>
<td>SA</td>
<td>-0.0000  -1.0000  3.0000</td>
<td>2081</td>
<td>19.63</td>
</tr>
<tr>
<td>DIRECT</td>
<td>0.0000   -1.0005  3.0001</td>
<td>191</td>
<td>4.98</td>
</tr>
<tr>
<td>AUMRE</td>
<td>-0.0002  -0.9999  3.0000</td>
<td>129</td>
<td>8.16</td>
</tr>
</tbody>
</table>
3.6.5 Benchmark Test 5 - Branin function (BR)

Branin function is also used as a benchmark function due to the difficulty to find its global minimum; the BR function given by Equation (3-21), is illustrated in Figure 3-8. The results are presented in Table 3-5.

\[
f(x_1, x_2) = \left( x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 \\
+ 10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10
\]  

(3-21)

Figure 3-8. The Objective Function of Benchmark Test 5

Table 3-5. Test Results on Branin Function

<table>
<thead>
<tr>
<th>Algorithm Used</th>
<th>Calculated Optimum, (X^<em>) and (f(X^</em>))</th>
<th>Number of Objective Function Evaluations</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical Minimum</td>
<td>-3.1416, 12.2750, 0.3978</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>PSO</td>
<td>-3.1403, 12.2790, 0.3979</td>
<td>184</td>
<td>28.78</td>
</tr>
<tr>
<td>GA</td>
<td>-3.1418, 12.2767, 0.3979</td>
<td>1000</td>
<td>15.11</td>
</tr>
<tr>
<td>MPS</td>
<td>-3.1444, 12.0996, 0.4311</td>
<td>26</td>
<td>14.67</td>
</tr>
<tr>
<td>SA</td>
<td>-3.1416, 12.2750, 0.3979</td>
<td>3201</td>
<td>20.19</td>
</tr>
<tr>
<td>DIRECT</td>
<td>-3.1241, 12.2165, 0.3996</td>
<td>159</td>
<td>0.02</td>
</tr>
<tr>
<td>AUMRE</td>
<td>-3.1040, 12.2119, 0.4054</td>
<td>168</td>
<td>14.30</td>
</tr>
</tbody>
</table>
3.6.6 Benchmark Test 6- Schaffer’s F6 Function

Schaffer’s function, a non linear two variable, has been considered due to the challenge to converge to its global minimum. The F6 function given by Equation (3-22) is illustrated in Figure 3-9. The results are presented in Table 3-6.

\[
f(x_1, x_2) = 0.5 + \frac{\sin^2 \left( \sqrt{x_1^2 + x_2^2} \right) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))}
\]

(3-22)

![Schaffers-F6 Function](image)

Figure 3-9. The Objective Function of Benchmark Test 6

<table>
<thead>
<tr>
<th>Algorithm Used</th>
<th>Calculated Optimum, (X^<em>) and (f(X^</em>))</th>
<th>Number of Objective Function Evaluations</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical Minimum</td>
<td>0.0000 0.0000 0.0000</td>
<td>417</td>
<td>17.036</td>
</tr>
<tr>
<td>PSO</td>
<td>0.0000 0.0000 0.0000</td>
<td>1040</td>
<td>12.644</td>
</tr>
<tr>
<td>GA</td>
<td>0.0041 0.0072 0.0001</td>
<td>34</td>
<td>16.590</td>
</tr>
<tr>
<td>MPS</td>
<td>0.0000 0.0000 0.0000</td>
<td>1015</td>
<td>10.594</td>
</tr>
<tr>
<td>SA</td>
<td>0.0000 0.0000 0.0000</td>
<td>69</td>
<td>0.9105</td>
</tr>
<tr>
<td>DIRECT</td>
<td>0.0000 0.0000 0.0000</td>
<td>31</td>
<td>2.1371</td>
</tr>
<tr>
<td>AUMRE</td>
<td>0.0000 0.0000 0.0000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.6.7 Benchmark Test 7- Griewank function (GN)

Griewank function is another benchmark problem that we have tested in this work. Griewank function given by Equation (3-23) is illustrated in Figure 3-10. The results are presented in Table 3-7.

\[
f(x_1, x_2) = \frac{x_1^2 + x_2^2}{200} - \cos x_1 \cos \frac{x_2}{\sqrt{2}} + 1
\]  

(3-23)

Figure 3-10. The Objective Function of Benchmark Test 7

Table 3-7. Test Results on Griewank Function

<table>
<thead>
<tr>
<th>Algorithm Used</th>
<th>Calculated Optimum, X* and f(X*)</th>
<th>Number of Objective Function Evaluations</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical Minimum</td>
<td>0.0000 0.0000 0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSO</td>
<td>-0.0006 0.0014 0.0000</td>
<td>411</td>
<td>41.910</td>
</tr>
<tr>
<td>GA</td>
<td>0.0015 -0.0015 0.0000</td>
<td>1000</td>
<td>21.451</td>
</tr>
<tr>
<td>MPS</td>
<td>-0.0146 0.0705 0.0014</td>
<td>108</td>
<td>30.766</td>
</tr>
<tr>
<td>SA</td>
<td>-0.0000 -0.0000 0.0000</td>
<td>1011</td>
<td>26.668</td>
</tr>
<tr>
<td>DIRECT</td>
<td>- - -</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>AUMRE</strong></td>
<td>-0.0009 0.0007 0.0000</td>
<td>154</td>
<td>12.562</td>
</tr>
</tbody>
</table>
3.6.8 Benchmark Test 8- Generalized Polynomial Function (GF)

Generalized polynomial function is used in the test also due to the challenge of convergence to the global minimum. The GF function given by Equation (3-24) is illustrated in Figure 3-11. The results are presented in Table 3-8.

\[ f(x_1, x_2) = (1.5 - x_1(1 - x_2))^2 + (2.25 - x_1(1 - x_2^2))^2 + (2.625 - x_1(1 - x_2^3))^2 \]  

(3-24)

![Figure 3-11. The Objective Function of Benchmark Test 8](image)

Table 3-8. Test Results on Generalized Polynomial Function

<table>
<thead>
<tr>
<th>Algorithm Used</th>
<th>Calculated Optimum, $X^<em>$ and $f(X^</em>)$</th>
<th>Number of Objective Function Evaluations</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical Minimum</td>
<td>2.0000 0.1700 0.5233</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSO</td>
<td>2.0000 0.1703 0.5233</td>
<td>85</td>
<td>13.990</td>
</tr>
<tr>
<td>GA</td>
<td>2.0000 0.1701 0.5233</td>
<td>1000</td>
<td>18.967</td>
</tr>
<tr>
<td>MPS</td>
<td>2.0000 0.1702 0.5233</td>
<td>383</td>
<td>272.991</td>
</tr>
<tr>
<td>SA</td>
<td>2.0000 0.1701 0.5233</td>
<td>3465</td>
<td>16.694</td>
</tr>
<tr>
<td>DIRECT</td>
<td>2.0000 0.1701 0.5234</td>
<td>7931</td>
<td>11.792</td>
</tr>
<tr>
<td>AUMRE</td>
<td>2.0000 0.1500 0.5261</td>
<td>132</td>
<td>9.3440</td>
</tr>
</tbody>
</table>
3.7 Summary and Discussions of Test Results

The results from the performance tests carried out in this work have demonstrated that the newly introduced AUMRE method has many advantages. The algorithm is capable of solving the challenging benchmark problems in global optimization; locating the optimum with comparable accuracy; and obtaining the results with much reduced computation time.

The use of DOE and RSM, and the avoidance of redundant examination of searched areas have contributed to the improved performance. Historically, the benchmark problems commonly used, and adopted here, are all limited to two design variables for the ease of illustration. It is expected that the computation efficiency advantage of the AUMRE approach can be more significant with the increased number of design variables and/or more computation demanding objective functions. Specifically the comparison studies showed that:

- **Performance or convergence accuracy**

  The new AUMRE method can locate the global minimum more accurately or reach comparable accuracy in most of the test cases (7 out of 8) and obtain slightly less accurate result in one of the 8 cases.

- **Computational efficiency**

  The CPU time needed by the new AUMRE method is only 53.30% of GA, 47.82% of SA, 33.74% of PSO, 100.13% of DIRECT, and 14.80% of MPS in average. These shows that AUMRE requires less computation time to converge to global solutions. It is also indicate that AUMRE is a computationally efficient algorithm.
3.7.1 The Proposed Algorithm Present Limitations and Continuous Work

The AUMRE algorithm has been recently introduced. Full implementation and computation tests/comparisons in high dimensional space have not yet been completed. AUMRE so far can deal with optimization problems in 2-D design space. Also AUMRE is not a standalone algorithm because it uses another local gradient based algorithm in its last step to help find the optimum point. AUMRE cannot handle constrained problems because of the classical sampling technique used to generate sample points as well as the and because of the way that AUMRE searches. The continuous research on this subject will be devoted to address this critical issue and the solution to problems encountered in the process.

To summarize the results obtained from AUMRE, in Table 3-9, the relative strength of each competing algorithm is illustrated. The minimum objective function value reached by all algorithms in the global minimization and the computation (or CPU) times needed by the SA algorithm to solve the eight benchmark problems are used as the references. Table 3-10, shows the number of function evaluations needed by each algorithm in the search. The new AUMRE algorithm converges to the global minimum of the benchmark test functions with less number of objective function evaluations in most cases.
### Table 3-9. Algorithm Performance Comparison (Relative Computation Time)

<table>
<thead>
<tr>
<th>Method</th>
<th>PSO</th>
<th>GA</th>
<th>MPS</th>
<th>SA</th>
<th>DIRECT</th>
<th>AUMRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpine</td>
<td>-6.1291</td>
<td>-6.1295</td>
<td>-6.1277</td>
<td>-6.1295</td>
<td>-6.1294</td>
<td>-6.1290</td>
</tr>
<tr>
<td>Rel. Comp. Time</td>
<td>1.63</td>
<td>1.13</td>
<td>1.66</td>
<td>1.00</td>
<td>0.83</td>
<td>0.65</td>
</tr>
<tr>
<td>Banana</td>
<td>0.0013</td>
<td>0.0063</td>
<td>0.0027</td>
<td>0.0022</td>
<td>0.0001</td>
<td>0.0029</td>
</tr>
<tr>
<td>Rel. Comp. Time</td>
<td>1.28</td>
<td>0.32</td>
<td>1.82</td>
<td>1.00</td>
<td>0.56</td>
<td>0.32</td>
</tr>
<tr>
<td>Rel. Comp. Time</td>
<td>1.97</td>
<td>1.23</td>
<td>0.86</td>
<td>1.00</td>
<td>0.50</td>
<td>0.49</td>
</tr>
<tr>
<td>GP</td>
<td>3.0000</td>
<td>3.0001</td>
<td>3.0000</td>
<td>3.0000</td>
<td>3.0001</td>
<td>3.0000</td>
</tr>
<tr>
<td>Rel. Comp. Time</td>
<td>1.01</td>
<td>0.60</td>
<td>1.69</td>
<td>1.00</td>
<td>0.25</td>
<td>0.42</td>
</tr>
<tr>
<td>BR</td>
<td>0.3979</td>
<td>0.3979</td>
<td>0.4311</td>
<td>0.3979</td>
<td>0.396</td>
<td>0.4054</td>
</tr>
<tr>
<td>Rel. Comp. Time</td>
<td>1.43</td>
<td>0.75</td>
<td>0.73</td>
<td>1.00</td>
<td>0.001</td>
<td>0.71</td>
</tr>
<tr>
<td>Schaffer F6</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Rel. Comp. Time</td>
<td>1.61</td>
<td>1.20</td>
<td>1.57</td>
<td>1.00</td>
<td>0.07</td>
<td>0.20</td>
</tr>
<tr>
<td>GN</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0014</td>
<td>0.0000</td>
<td>---</td>
<td>0.0000</td>
</tr>
<tr>
<td>Rel. Comp. Time</td>
<td>1.57</td>
<td>0.80</td>
<td>1.15</td>
<td>1.00</td>
<td>---</td>
<td>0.47</td>
</tr>
<tr>
<td>GF</td>
<td>0.5233</td>
<td>0.5233</td>
<td>0.5233</td>
<td>0.5233</td>
<td>0.5234</td>
<td>0.5261</td>
</tr>
<tr>
<td>Rel. Comp. Time</td>
<td>0.84</td>
<td>1.14</td>
<td>16.35</td>
<td>1.00</td>
<td>0.71</td>
<td>0.56</td>
</tr>
</tbody>
</table>

Note: Rel. Comp. Time – Relative Computation Time; Obj. Fun. Value – Objective Function Value; GP–Goldstein and Price Function; BR – Branin Function; F6 – Schaffer’s Function; GN – Griewank Function; and GF – Generalized Polynomial Function; The tests were done on a Dell Dimension XPS T600 PC with Intel Pentium III Processor 598 Hz.

### Table 3-10. Algorithm Performance Comparison (Number of Objective Function Evaluations)

<table>
<thead>
<tr>
<th>Method</th>
<th>PSO</th>
<th>GA</th>
<th>MPS</th>
<th>SA</th>
<th>DIRECT</th>
<th>AUMRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpine</td>
<td>-6.1291</td>
<td>-6.1295</td>
<td>-6.1277</td>
<td>-6.1295</td>
<td>-6.1294</td>
<td>-6.1290</td>
</tr>
<tr>
<td># Func. Eval.</td>
<td>125</td>
<td>1000</td>
<td>45</td>
<td>1391</td>
<td>267</td>
<td>125</td>
</tr>
<tr>
<td>Banana</td>
<td>0.0013</td>
<td>0.0063</td>
<td>0.0027</td>
<td>0.0022</td>
<td>0.0001</td>
<td>0.0029</td>
</tr>
<tr>
<td># Func. Eval.</td>
<td>146</td>
<td>1000</td>
<td>227</td>
<td>2153</td>
<td>1347</td>
<td>381</td>
</tr>
<tr>
<td># Func. Eval.</td>
<td>241</td>
<td>1000</td>
<td>34</td>
<td>2524</td>
<td>195</td>
<td>136</td>
</tr>
<tr>
<td>GP</td>
<td>3.0000</td>
<td>3.0001</td>
<td>3.0000</td>
<td>3.0000</td>
<td>3.0001</td>
<td>3.0000</td>
</tr>
<tr>
<td># Func. Eval.</td>
<td>125</td>
<td>1000</td>
<td>227</td>
<td>2081</td>
<td>191</td>
<td>129</td>
</tr>
<tr>
<td>BR</td>
<td>0.3979</td>
<td>0.3979</td>
<td>0.4311</td>
<td>0.3979</td>
<td>0.396</td>
<td>0.4054</td>
</tr>
<tr>
<td># Func. Eval.</td>
<td>184</td>
<td>1000</td>
<td>26</td>
<td>3201</td>
<td>159</td>
<td>168</td>
</tr>
<tr>
<td>Schaffer F6</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td># Func. Eval.</td>
<td>417</td>
<td>1040</td>
<td>34</td>
<td>1015</td>
<td>69</td>
<td>31</td>
</tr>
<tr>
<td>GN</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0014</td>
<td>0.0000</td>
<td>---</td>
<td>0.0000</td>
</tr>
<tr>
<td># Func. Eval.</td>
<td>411</td>
<td>1000</td>
<td>108</td>
<td>1011</td>
<td>---</td>
<td>154</td>
</tr>
<tr>
<td>GF</td>
<td>0.5233</td>
<td>0.5233</td>
<td>0.5233</td>
<td>0.5233</td>
<td>0.5234</td>
<td>0.5261</td>
</tr>
<tr>
<td># Func. Eval.</td>
<td>85</td>
<td>1000</td>
<td>383</td>
<td>3465</td>
<td>401</td>
<td>132</td>
</tr>
</tbody>
</table>
Chapter 4. Metamodeling and Search Using Space Exploration and Unimodal Region Elimination

4.1 Introduction

Computation-intensive simulations and analyses are frequently used to support engineering design and design optimization. The high computation costs associated with these simulations and analyses preclude design optimization methods that require extensive evaluations of the objective and constraint functions. Efficient, effective and robust optimization methods are needed.

Metamodelling based search, space exploration, and region reduction/elimination methods are effective optimization schemes for computation intensive global design optimization problems. One way to avoid the repeated and redundant evaluations of computationally intensive objective and constraint functions is to explore the design space in an optimal design problem by removing the less promising and previously searched regions. Effective identification of promising regions that have a higher potential to contain design solutions will speed up the search process. These promising regions are identified by dividing the design space into many subspaces and carrying out search in each one of these subspaces or by tackling the most promising regions that are
identified based on heuristics, educated guess and preliminary tests. For multimodal objective functions with many peaks and valleys, reducing the search space to a number of more manageable, unimodal regions can solve the complex global optimization problem step by step and put more focus on the more promising regions, thus increasing the chance to quickly identify the global solution.

The space exploration and region reduction methods start their search by sending agents or sampling points to explore the entire design space. These agents will yield some information (values of objective and constraint functions) about the design space. The obtained information provide hints to rank different regions based on their potentials to contain the global optimum, allowing more emphasis being put on them during subsequent searches. More samples are normally created in these promising regions to further explore the regions and to refine the unimodal model over each region. Exploration of the design space is also very valuable for a better understanding of the design problem. A surrogate model, or metamodel, is easy to construct and computationally cheaper to calculate, comparing to the original objective function. Quick calculation using the relatively simple metamodel is often referred as calculation of a “cheap point”, while obtaining the value of the original objective using numerically intensive analysis and simulation produces an “expensive point.” For optimization problems in which the calculations of the objective and constraint functions require extensive numerical analysis and simulation, the introduction of a metamodel and the use of cheap points to replace expensive points can considerably reduce computation time and make global optimization feasible.
In a design space reduction method, the optimum is identified in two steps. First, a crude technique is used to find the lower and upper bounds of the design variables. Thereafter, a more sophisticated scheme is used to search within these limits and find the optimum with the desired accuracy.

Region elimination schemes intend to keep the size of the design space under control. Many region elimination methods have been recently proposed. Among those, the most representative group is the multivariable elimination procedures [137], which reduce the design space by discarding the unpromising regions that do not contain the optimum. Another approach, metaheuristics [138] which uses a complex probability model to identify the promising regions with a higher chance to contain the design optimum. However, the strategy is computationally less efficient due to the complexity of the probabilistic models used.

In this chapter, a new global optimization algorithm, namely Space Exploration and Unimodal Region Elimination (SEUMRE) method, is introduced. The approach divides the design space into several unimodal regions using design experiment data; identifies and ranks the regions that most likely contain the global minimum; fits a Kriging model with additional design experiments using latin hypercube designs over the most promising region; identifies its minimum, and moves to the next most promising region. Step by step, the method identifies the global optimum by examining the most promising unimodal regions with additional design experiments.

The limitations of AUMRE such as solving only 2-D optimization problems, depending on classical design of experiments and using low order polynomial surface model which lacks accuracy in imitating the black-box functions are handled in
SEUMRE. SEUMRE solve design problem in N-D design space, classical design of experiments are replaced with *latin hypercube designs* which generates sample points that are distributed equally in the design space which allows exploring the design space in an efficient way. Also a more accurate metamodel is used in SEUMRE and that is *Kriging* which is known for its accuracy in imitating the expensive functions (black-box functions). SEUMRE also efficiently solves constrained design problems. SEUMRE does not depend on other gradient based local optimization algorithms like AUMRE. SEUMRE is a standalone algorithm.

### 4.2 The Proposed Algorithm

As an extension to the design experiments, region elimination and response surface model based global optimization algorithm, AUMRE [17], the new *Space Exploration and Unimodal Region Elimination* (SEUMRE) algorithm is introduced. The algorithm consists of the following key elements: a) dividing the design space into several unimodal regions using design experiment data; b) identifying and ranking the regions that most likely contain the global minimum; c) fitting a Kriging model [124] with additional design experiments using Latin Hypercube designs over the most promising region and identifying its minimum; and, d) moving to the next most promising region. The processing carries on until all promising unimodal regions are processed, and the global optimum is obtained from identified local optima.

#### 4.2.1 Steps of the Proposed Algorithm

1) Generate a set of design data points, \( x_i, i = 1, \ldots, n \) over the design space;
Where \( p \) is the number of design variables. The number of initial design data point, \( n \), is normally selected to be 5~10\( n \), increasing with the number of design variables, \( p \).

2) Evaluate the values of the objective function and constraints using the selected design points;

\[
y^l = \min\{f(x_i); (x_i, f(x_i)) \in S\}
\]

and denote

\[
x^{(l)} = (x_1^l, x_2^l, \ldots, x_n^l); f(x^l) = y^{(l)}
\]

where \( l \) represents the number of the unimodal regions.

If there are several \( x \)'s corresponding to \( y^l \), just choose one.

3) Use the obtained objective function values as clues to identify the regions that most likely contain the global minimum and rank the other regions on their possibility to contain the global minimum from high to low (Figure 4-1). Special attention is given to the first three most promising regions;

Put \( (x^{(l)}, f(x^{(l)})) \) into \( C_l \), and carry out the following partition:

\[
S = \bigcup_{i=1}^{l} C_i;
\]

4) Identify the rough upper and lower bounds of the promising unimodal regions, starting from the “center” point and moving up and down along each coordinate to find the upper and lower limits of each design variable, points that are seized.
between these boundaries are considered neighboring design points (Figure 4-2), specifically

![Diagram of Search Space and Design Points]

**Figure 4-1. Dividing the Design Space Based on Obtained Function Values**

- Use the point with the minimum function value $x_k^l (k = 1:n)$ as the centre point to start the search; and assign the function value at this point as $q = f(x)$.

- Identify the turning points that mark the rough bounds of the unimodal region by searching for the next point $x$ in $S$ along the up and down directions of the coordinates (e.g. moving right, left, up and down for $n = 2$), and comparing the function value at each point with the function value of the previous point.

- If $f(x) \geq q$, then this point is considered as a neighboring point, add $(x, f(x))$ into $C_i$ and let $q = f(x)$; repeat the same procedure for the next design space coordinate. When $f(x) < q$, a turning point or bound is identified, and then search the bound in the opposite coordinate direction following the same procedure.
Subtract the identified unimodal region $C_t$ from the design space $S$ until all of the points in the design space $S$ are visited. Specifically, let $S = S / C_t$

If $S \neq \phi$, and move to the next promising region i.e. let $l = l + 1$.

**Figure 4-2. Identifying the Promising Unimodal Region Boundaries**

5) Once the boundaries of the approximated unimodal regions has been identified, refine the data set of this most promising region by adding more experiment or “expensive” points (points evaluated using the objective function) into the region by using *Latin Hypercube Designs*, and introduce a Kriging approximation model over the region;

6) Add more “cheap” design points that can be obtained using the easy to calculate approximation model. In present approach, around 300 cheap points are generated using the metamodel over the promising unimodal region to identify the local optimum before the region is removed from the list;

7) Repeat the previous steps until the local optima of all promising unimodal regions are located. The global optimum is then identified by comparing the values of all local optima.
\[ f_{\min_{(global)}} = \min(f_{\min_1}, f_{\min_2}, \ldots, f_{\min_i}) \quad i = 1, \ldots, n_r \]  

where \( n_r \) is the number of unimodal regions processed.

Figure 4-3 shows the flow diagram of the proposed algorithm.

**Figure 4-3. The Flow Diagram of the Proposed Algorithm**

### 4.3 Kriging Metamodel for Approximation

Kriging metamodels [124] and related Gaussian processes are widely studied and used in recent years due to their ability to provide insights to the search space, better modelling
accuracy, and good design space exploration efficiency [53, 139-141]. The modelling process consists of several steps:

- Generate a set of design points as

\[ x = \{x_1, x_2, ..., x_n\}, \ x_i \in \mathbb{R}^p \tag{4-6} \]

where \( p \) is the number of design variable.

- The resulting outputs from the function to be modeled are

\[ Y = \{y_1(x), y_2(x), ..., y_n(x)\} \tag{4-7} \]

Using likelihood estimation, the unknown parameters of the correlation function have to be estimated in this approach, using the best linear unbiased prediction of the response

\[ \hat{y} = \hat{\beta}^T f(x) + r^T(x)R^{-1}(\bar{y} - \hat{F}\hat{\beta}) \tag{4-8} \]

where \( r^T(x) \) is a vector representing the correlation between an unknown set of points \( \mathbf{x} \) and all known experimental points.

\[ r^T(x) = [R(x, x_1), R(x, x_2), ..., R(x, x_n)] \tag{4-9} \]

The second term in Equation (4-8), represents an interpolation of the residuals of the regression model \( \tilde{\beta}^T f(x) \). All responses data will thus be predicted. The regression model used is a zero order polynomial regression model. The unknown parameters \( \beta \) and \( \sigma^2 \) can be estimated using Equations (2-5) and (2-6).

The Gaussian correlation function is used due to its ability to control the range of influence and the smoothness of the approximation function.
\[ R(x_i, x_j) = \exp \left( -\sum_{k=1}^{p} \theta (x_i^k - x_j^k)^2 \right) \] (4-10)

4.4 Sampling Techniques

In this work, Grid sampling [25] is used first and random Latin Hypercube Sampling design (LHD) [61, 62, 142] is used afterwards to obtain the design experiment points.

*Grid sampling* is used due to its ease of implementation, broad coverage and ability for uniformly sampling over the entire design space. For an interested region of parallelepiped form, \( l_j \leq x_j \leq u_j, \ j = 1, ..., n \) the simplest distribution of design data points is defined by the different combinations of

\[ S_j^i \equiv l_j + k_j^i \frac{u_j - l_j}{v_j}, \quad k_j^i = 0, 1, ..., v_j \] (4-11)

where \( v_j \) are integers. For all \( v_j = v \), the number of the design points becomes \( (v + 1)^n \).

*Latin Hypercube Sampling Design* is used after the promising unimodal regions are identified to form the Kriging metamodel for exploring the design space. Specifically, LHD is used to select \( n \) different values from each of the \( k \) variables, \( X_1, X_2, ..., X_k \) in the following manner. The range of each variable is divided into \( n \) non-overlapping intervals on the basis of equal probability. One value from each interval is selected randomly based upon the probability density of the interval. The \( n \) values obtained for \( X_1 \) are paired randomly with the \( n \) values of \( X_2 \). These \( n \) pairs are further combined randomly with the \( n \) values of \( X_3 \) to form \( n \) triplets, and so on, until \( n k \)-triplets are formed. The objective function is then evaluated using these \( n k \)-triplets, \( n k \)-dimensional input
vectors, to obtain \( n \) responses values \( Y(x) \). This sampling method forms an \((n\times k)\) matrix of input, where the \( i^{th} \) row contains values of each of the \( k \) input variables to be used on the \( i^{th} \) run of the metamodel. In other words, the LHS strategy basically divides the interval of each dimension of the design variable into \( m \) non-overlapping intervals with equal probability; get a random sample from each interval in each dimension; and pair these samples randomly from each dimension.

### 4.5 Region Elimination

The region elimination scheme of the proposed algorithm emerged from the AUMRE algorithm, introduced previously by the author of this dissertation Younis et al. [17]. Similarly, the new SEUMRE algorithm divides the field of interest into several unimodal regions based on the function values, as discussed previously; identifies the promising regions, gets more design experiment points and introduces Kriging metamodels over these regions; and finds the local optimum of each region, then the global optimum of all regions. By removing previously explored areas and avoiding redundant searches, the approach can identifies global optimum with less computational cost.

### 4.6 Test Problems and Results

To illustrate the advantage of the new SEUMRE algorithm over other widely used region elimination and space exploration global optimization methods, a number of commonly used benchmark problems for testing the convergence capability and computational performance of the new global optimization algorithms are used in this work to gauge the capability of the newly introduced algorithm. These include for unconstrained optimization; Shubert Function; Six-hump Camel-back Function (SC);
Goldstein and price function (GP); Branin function; Banana function; Levy Function; Shekel Function; Sphere function; Hartman functions with 6 and 16 design variables (H6 & H16). And for constrained optimization problems, nonlinear objective function with inequality (C1 and C3) and equality (C2) constraints as well as the tension compression string problem.

The detailed form of these benchmark test problems are given in Appendix A. These benchmark problems are representative in terms of their structure and dimensionality. Both low and high dimensional, as well as unconstrained and constrained benchmark problems are used in the tests. The computational performance of SEUMRE is also compared against its predecessor, AUMRE, to gauge the improvements.

Results from global optimization runs using SEUMRE on the representative benchmark problems and the required computations represented by the number of objective function evaluations, number of search iterations and CPU time are shown in Appendix B, and summarized in Table 4-1. Computational performance comparisons between SEUMRE and a number of existing global optimization methods are presented in Table 4-2 and Figure 4-6, while the comparison to the recently introduced AUMRE algorithm is given in Table 4-3. Table 4-4 presents a comparison between the new SEUMRE method and its predecessor AUMRE in light of the reduced computation costs. Test results on constrained optimization problems are shown in Table 4-5 Validation results that show the accuracy of the constructed Kriging models are shown in Table 4-1. Root Mean Square Error (RMSE) validation technique is used in this work to validate the constructed Kriging metamodels.
Table 4-1. Test Results of the New SEUMRE Method

<table>
<thead>
<tr>
<th>Test Problem</th>
<th>n</th>
<th>Original Space</th>
<th>Most Promising Design Space</th>
<th>Global Optimum Solution</th>
<th># F. Evl.</th>
<th>CPU time</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shubert</td>
<td>2</td>
<td>[-10 ; 10]</td>
<td>[-7.71 5.48; -7.70 5.48]</td>
<td>-186.7309</td>
<td>56</td>
<td>4.10</td>
<td>1.2e-06</td>
</tr>
<tr>
<td>SC</td>
<td>2</td>
<td>[-2 ; 2]</td>
<td>[0.04 0.12; -0.78 -0.70]</td>
<td>-1.0320</td>
<td>33</td>
<td>2.00</td>
<td>6.0e-05</td>
</tr>
<tr>
<td>GP</td>
<td>2</td>
<td>[-2 ; 2]</td>
<td>[-0.14 0.04; -1.0 -0.80]</td>
<td>3.0000</td>
<td>112</td>
<td>12.4</td>
<td>4.1e-06</td>
</tr>
<tr>
<td>Branin</td>
<td>2</td>
<td>[-5 0 ; 5 10]</td>
<td>[-3.21 -3.01; 12.67 11.6]</td>
<td>0.3979</td>
<td>32</td>
<td>1.75</td>
<td>6.5e-05</td>
</tr>
<tr>
<td>Banana</td>
<td>2</td>
<td>[-2 ; 2]</td>
<td>[0.70 1.00; 0.60 1.00]</td>
<td>0.0000</td>
<td>40</td>
<td>2.98</td>
<td>0.0015</td>
</tr>
<tr>
<td>Levy</td>
<td>4</td>
<td>[-10 ; 10]</td>
<td>[0.88 1.06; 0.91 1.40; 0.79 1.41; 0.71 1.17]</td>
<td>0.0000</td>
<td>120</td>
<td>12.3</td>
<td>5.9e-05</td>
</tr>
<tr>
<td>Shekel</td>
<td>4</td>
<td>[0 ; 10]</td>
<td>[3.9 4.01; 4.00 4.02; 3.94 4.01; 3.89 4.01]</td>
<td>-10.1531</td>
<td>108</td>
<td>11.7</td>
<td>1.7e-09</td>
</tr>
<tr>
<td>Sphere</td>
<td>10</td>
<td>[-5.12; 5.12]</td>
<td>**</td>
<td>0.0000</td>
<td>282</td>
<td>158.3</td>
<td>2.8e-05</td>
</tr>
<tr>
<td>Hartmann, H6</td>
<td>6</td>
<td>[0 ; 1]</td>
<td>[0.10 0.20; 0.14 0.15; 0.47 0.49; 0.26 0.28; 0.30 0.31; 0.64 0.65]</td>
<td>-3.3220</td>
<td>79</td>
<td>14.2</td>
<td>0.0151</td>
</tr>
<tr>
<td>Hartmann, H16</td>
<td>16</td>
<td>[0 ; 1]</td>
<td>**</td>
<td>25.875</td>
<td>262</td>
<td>433.2</td>
<td>0.0042</td>
</tr>
</tbody>
</table>

Note: n = Problem’s number of dimensions; # F. Evl = Number of Function Evaluations; CPU in Sec.; The tests were done on a Dell OPTIPLEX 745 PC with Intel(R) Core(TM)2 CPU 6400 @ 2.13 GHz, 2.00GB of RAM.

The convergence process of the proposed method (SEUMRE) to the global optimum of each tested benchmark problem is shown in Appendix B. In this section we selected two benchmark problems as examples to show how SEUMRE converge to the global optima. Sphere function with 10 design variables and Hartmann function (H16) with 16 design variables. Figure 4-4 shows the convergence process for Sphere function and Figure 4-5 shows the convergence process for Hartmann (H16) function.
Figure 4-4. Convergence Process of SEUMRE for Sphere Function

Figure 4-5. Convergence process of SEUMRE for Hartmann (H16) Function
Figure 4-6. Performance Comparison of Space Exploration Optimization Methods for the Tested Benchmark Problems

Table 4-2. Performance Comparisons between SEUMRE and other Space Exploration Optimization Methods on Selected Benchmark Test Problems

<table>
<thead>
<tr>
<th>Benchmark Test Problem</th>
<th>Optimization Method</th>
<th>Analytical minimum</th>
<th>Global Optimum</th>
<th># F. Eval.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Global minimum $x^*$</td>
<td>Optimum $f^*$</td>
</tr>
<tr>
<td>Six-hump Camel-back Function (SC) $n=2$</td>
<td>AUMRE</td>
<td>-1.032 (0.0900, -0.7090)</td>
<td>-1.0315</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>SEUMRE</td>
<td>-1.032 (0.0896, -0.7123)</td>
<td>-1.0316</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>MPS</td>
<td>-1.032 (0.088, -0.7120)</td>
<td>-1.0316</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>DIRECT</td>
<td>-1.032 (0.0903, -0.7157)</td>
<td>-1.0136</td>
<td>4037</td>
</tr>
<tr>
<td>Levy Function $n=4$</td>
<td>SEUMRE</td>
<td>0.0000 (0.9939, 0.9907, 0.9985, 0.9834)</td>
<td>0.0001</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>MPS</td>
<td>0.0000 (0.9910, 1.0276, 0.9407, 0.9954)</td>
<td>0.0008</td>
<td>401</td>
</tr>
<tr>
<td></td>
<td>DIRECT</td>
<td>0.0000 (1.0000, 0.9999, 0.9999, 1.0000)</td>
<td>0.0000</td>
<td>1553</td>
</tr>
<tr>
<td>Shekel Function $n=4$</td>
<td>SEUMRE</td>
<td>-10.1531 (4.0000, 4.0009, 4.0003, 4.0009)</td>
<td>-10.1531</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>MPS</td>
<td>-10.1531 (4.0004, 3.9984, 4.0005, 4.0006)</td>
<td>-10.1531</td>
<td>172</td>
</tr>
<tr>
<td></td>
<td>DIRECT</td>
<td>-10.1531 (3.9986, 3.9986, 3.9986, 3.9986)</td>
<td>-10.1531</td>
<td>293</td>
</tr>
</tbody>
</table>
### Table 4-3. Performance comparison between AUMRE and SEUMRE algorithms

<table>
<thead>
<tr>
<th>Test Problem</th>
<th>Previous Approach (AUMRE)</th>
<th>Improved Approach (SEUMRE)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Calculated Optimum Solution</td>
<td>Calculated Optimum Solution</td>
</tr>
<tr>
<td></td>
<td>$x_1$</td>
<td>$x_2$</td>
</tr>
<tr>
<td>SC</td>
<td>0.0900</td>
<td>-0.7090</td>
</tr>
<tr>
<td>GP</td>
<td>-0.0002</td>
<td>-0.9999</td>
</tr>
<tr>
<td>Branin</td>
<td>-3.1040</td>
<td>12.2119</td>
</tr>
<tr>
<td>Banana</td>
<td>1.0476</td>
<td>1.1000</td>
</tr>
<tr>
<td>Alpine</td>
<td>7.9082</td>
<td>4.8244</td>
</tr>
<tr>
<td>Beak</td>
<td>0.2467</td>
<td>-1.7863</td>
</tr>
<tr>
<td>Schaffer's</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>GN</td>
<td>-0.0009</td>
<td>0.0007</td>
</tr>
<tr>
<td>GF</td>
<td>2.0000</td>
<td>0.1500</td>
</tr>
</tbody>
</table>

### Table 4-4. Percentage reduction of number of function evaluations and CPU time

<table>
<thead>
<tr>
<th>Test problem</th>
<th>Previous Approach (AUMRE)</th>
<th>Improved Approach (SEUMRE)</th>
<th>% Reduction #F. Evl.</th>
<th>% Reduction CPU time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#F. Evl.</td>
<td>CPU time (sec)</td>
<td>#F. Evl.</td>
<td>CPU time (sec)</td>
</tr>
<tr>
<td>SC</td>
<td>69</td>
<td>6.2</td>
<td>33</td>
<td>2</td>
</tr>
<tr>
<td>GP</td>
<td>129</td>
<td>12.9</td>
<td>112</td>
<td>12.38</td>
</tr>
<tr>
<td>Branin</td>
<td>186</td>
<td>14.3</td>
<td>32</td>
<td>1.75</td>
</tr>
<tr>
<td>Banana</td>
<td>381</td>
<td>10.2</td>
<td>40</td>
<td>2.98</td>
</tr>
<tr>
<td>Alpine</td>
<td>125</td>
<td>7.91</td>
<td>27</td>
<td>1.68</td>
</tr>
<tr>
<td>Beak</td>
<td>136</td>
<td>9.57</td>
<td>38</td>
<td>2.59</td>
</tr>
<tr>
<td>Schaffer's</td>
<td>31</td>
<td>2.13</td>
<td>24</td>
<td>1.24</td>
</tr>
<tr>
<td>GN</td>
<td>154</td>
<td>12.6</td>
<td>117</td>
<td>12.5</td>
</tr>
<tr>
<td>GF</td>
<td>132</td>
<td>9.34</td>
<td>61</td>
<td>4.95</td>
</tr>
</tbody>
</table>
Table 4-5. Optimization results on constrained test problems using SEUMRE

<table>
<thead>
<tr>
<th>Problem</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>Cal. F. Value.</th>
<th># F. Evl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C1$</td>
<td>14.095</td>
<td>0.8430</td>
<td>--</td>
<td></td>
<td></td>
<td>-6961.814</td>
<td>47</td>
</tr>
<tr>
<td>$C2$</td>
<td>-1.7171</td>
<td>1.5957</td>
<td>1.8272</td>
<td>-0.7636</td>
<td>-0.7636</td>
<td>0.0539</td>
<td>59</td>
</tr>
<tr>
<td>$C3$</td>
<td>78.8154</td>
<td>33.0201</td>
<td>30.0848</td>
<td>44.9199</td>
<td>36.7249</td>
<td>-30585</td>
<td>541</td>
</tr>
<tr>
<td>$T-C$ string</td>
<td>0.0517</td>
<td>0.3580</td>
<td>11.2139</td>
<td>--</td>
<td></td>
<td>0.01266</td>
<td>72</td>
</tr>
</tbody>
</table>

Note: $T-C$ string = Tension Compression String; Cal. F. Value = Calculated Function Value

4.7 Summary and Discussion of Test Results

Several different types of benchmark problems commonly used in measuring the performance of new optimization methods are used to examine the performance of the proposed method and its capability in handling low and high dimensional problems. Results from the presented tests showed that the newly introduced SEUMRE algorithm has outperformed existing global optimization methods. The new algorithm is capable of handling and solving the challenging benchmark problems in global optimization; locating the optimum with comparable accuracy; and obtaining the results with much reduced number of objective function evaluations and computation time. Constrained optimization problems were also tested with satisfactory results. The use of LHD space filling sampling method, the incorporation of Kriging metamodels, and the avoidance of redundant examination of searched areas have contributed to the improved performance. Specifically the comparison studies showed:

- **Performance or convergence accuracy**

  The new SEUMRE method can locate the global minimum more accurately or reach comparable accuracy in most of the test cases. Figure 4-4 and Figure 4-5 shows the convergence trends of the proposed method on different benchmark test problems.
and how the proposed method can reach global solutions with comparable accuracy and at less computational cost. The method normally converges after several function evaluations and uses a larger number of additional evaluations to ensure good accuracy on the identified global optimum.

- **Computational efficiency**

  The number of function evaluations needed by the new SEUMRE method to converge to the global minimum is way less than that of other space exploration and region elimination methods. For instance, SEUMRE took 33 function evaluations to identify the minimum for the Six-hump Camel-back (SC) function, while MPS, DIRECT, and AUMRE required 71, 4037, and 69 function evaluations, respectively (Table 5-2 and Figure 4-6).

  As shown in Table 4-4, the reductions in CPU time for SEUMRE, comparing with AUMRE, over SC; GP; Branin; Banana; Alpine; Beak; Schaffer’s; GN; and GF benchmark test problems are 67.7%; 4.0%; 87.8%; 70.8%; 78.8%; 72.9%; 41.9% 7.0%; and 47.0%, respectively. The new method’s computational efficiency is also better compared to MPS and other optimization methods.

4.7.1 The Proposed Algorithm’s Limitations

At present, the SEUMRE algorithm shows slow convergence to problems with very large number of design variables. Continuous efforts are given to extend the new algorithm to handle larger dimension and multi-objective optimization problems. A number of practical industrial design problems involving global optimization are being used to test the new algorithm, including the design optimization of two-mode hybrid powertrains for new hybrid vehicles.
4.8 Conclusions

In this chapter, a new global optimization algorithm, namely Space Exploration and Unimodal Region Elimination (SEUMRE) method, is introduced. The approach divides the field of interest into several unimodal regions using design experiment data; identifies and ranks the regions that likely contain the global minimum; fits a Kriging model over each promising region with additional design experiments data points using *Latin Hypercube designs*; identifies its minimum and removes the processed region; and moves to the next most promising region until all promising regions are processed and the global optimum is identified. By avoiding redundant searches, the approach identifies the global optimum with reduced number of objective function evaluations and computation effort. The new algorithm was tested using a variety of benchmark global optimization problems and compared with several widely used region elimination and space exploration global optimization algorithms. The experiment results showed robust performance, comparable search accuracy and superior computation efficiency, making the new SEUMRE method an excellent tool for computation intensive, computer analysis/simulation and black-box function based global design optimization problems.
Chapter 5. Mixed Surrogate and Space Elimination in Computationally Intensive Designs

5.1 Introduction

Surrogate-based modeling is an effective search method for global design optimization over well defined areas, using complex and computationally intensive analysis and simulation tools. However, identifying the appropriate surrogate models and their suitable areas remain a challenge that requires extensive human intervention. The custom in surrogate-based modeling of complex engineering problems is to fit one or more surrogate models and select the one that performs best. In this chapter, a new global optimization algorithm, namely Mixed Surrogate and Space Elimination (MSSE) method, is introduced. Representative surrogate models, including Quadratic Response Surface, Radial Basis function, and Kriging, are mixed with different weight ratio to form an adaptive metamodel with best tested performance. The approach divides the field of interest into several unimodal regions; identifies and ranks the regions that likely contain the global minimum; fits the weighted surrogate models over each promising region using additional design experiments data points from Latin Hypercube Designs and adjusts the weights according to the performance of each model; identifies its minimum
and removes the processed region; and moves to the next most promising region until all regions are processed and the global optimum is identified.

This work combines the efforts of using mixed surrogates, effective reduction of design space and appropriate data sampling into one algorithm. The approximate models (surrogates) used in the work is first introduced. The mechanism for mixing different surrogates and for combining it with unimodal region identification and space filling sampling schemes is then presented. Testing results of the new MSSE algorithm using several well known benchmark problems for regular and global optimization are presented. Computational efficiency of the new algorithm is compared with those of the widely used space exploration global optimization algorithms, DIRECT and MPS, to show the reduced computation efforts, robust performance and comparable search accuracy.

5.2 Used Surrogate Models

In this work, the proposed algorithm employs three different surrogates to fit the obtained design points. These are the Response Surface Model (RSM) of quadratic response function (QRF), the linear radial basis function (RBF) and Kriging. Extensively used in the past, QRF smoothes the sampled data points using a quadratic model and predicts the trend of the original function [34]. On the other hand, the RBF passes all the “expensive” points obtained using the original numerical analyses/simulations, prevents unnecessary curvature added to the unknown function, and preserves the minimum from these expensive points. Both QRF and RBF are simple, intuitive, and easy to construct. Although the third selection, Kriging, is expensive to construct, it is more accurate and promising for region elimination and space exploration techniques. An overview and
complete details on the surrogates which were used in this approach were previously introduced in Chapter 3.

5.3 The Proposed Approach

The newly proposed Mixed Surrogate and Space Elimination (MSSE) search method consists of the following three major components:

- A combination of three surrogate models, QRF, RBF and Kriging, to form the metamodel to acquire “cheap” design points, or approximates to the analysis/simulation results, or “expensive points”, which require intensive computation; and an adaptive mechanism that automatically adjusts the weighted contribution of each surrogate model based on the performance of different model weight ratio.

- The approximate unimodal region identification and region elimination scheme (AUMRE) for global optimization previously introduced by Younis et al. [17].

- Sampling method that effectively covers the interested design space/region which is Latin Hypercube Designs (LHD);

The new algorithm integrates these three components in a step-by-step and iterative search process.

5.3.1 Metamodel with Mixed Surrogates

The essence of this work is to fit available design points in the design space or promising regions, using an appropriate linear combination of the three surrogate models, QPF, RBF and Kriging (KRG), to produce a better metamodel that can approximate the computation model.

The mixed surrogate model takes the following form
\[ \hat{f}_{\text{mixed}}(X) = \alpha_{k1} Q(X) + \alpha_{k2} R(X) + \alpha_{k3} K(X) \]  
\[ \alpha_{k1} + \alpha_{k2} + \alpha_{k3} = 1 \]

where \( \hat{f}_{\text{mixed}}(X) \) is the metamodel approximate to the analysis/simulation function, \( f(X) \), at sample point \( x \); \( Q(x), R(x) \) and \( K(x) \) represents the QRF, RBF and KRG surrogate models respectively; \( \alpha_{k1}, \alpha_{k2} \) and \( \alpha_{k3} \) are weight coefficients that determine the contribution of the three models to the mixed metamodel with \( k \) different test combinations.

### 5.3.2 Steps of the Proposed Approach

1. Generating a set of design data points, \( X_i, i = 1, \ldots, p \) (roughly \( 3n \sim 5n \)) over the entire design space that needs to be explored using LHD sampling:

\[ \Phi_0 = \{X_1, X_2, \ldots, X_p\}, \ X_i \in S^n \]  

where \( n \) is the number of design variables and \( X_i \) is defined in the \( n \) dimensional space.

2. Running analysis/simulation to evaluate the objective function, \( f(X) \), using this small set of expensive design data points, \( \Phi_0 \).

3. Identifying the region that most likely contains the global minimum based on the obtained function values; special attention is given to the top three ranked promising regions, and finding its approximate unimodal region:

\[ y^{(i)} = \min\{f(X_i) : (X_i, f(X_i)) \in S\} \]
4) Setting the unimodal region counter \( l = 1 \).

Based on obtained function values;

Put \((X^{(l)}, f(X)^{(l)})\) into \(C_l\), and carry out the following partition:

\[
S = \bigcup_{i=1}^{l} C_i; 
\]

(5-5)

Where \(X^{(l)}\) represents the coordinates of the design points found in this region and \(f(X)^{(l)}\) represents the function values of the expensive function at these points; these can be represented mathematically as;

\[
X^{(l)} = (X_1^l, X_2^l, ..., X_p^l); \text{ and } f(X)^{(l)} = y^{(l)} 
\]

(5-6)

a) Identifying the upper and lower boundaries of the promising unimodal regions, using the center point as the point to start the search by moving left, right, up and down, or along each design variable coordinate in the design space, to find the upper and lower limits of each design variable. Figure 4-2 shows how the rough boundaries of the promising regions can be identified.

b) Use the coordinates of the point with the minimum objective function value \(X_k^l (k = 1: n)\) as the centre point to start the search; assign the function value at this centre point as \(q = y\).

c) To identify the turning points of the unimodal region boundaries, search for the next point \(X\) in \(S\) by moving along the positive and the negative directions of the coordinates (e.g. moving right, left, up, and down if \(n = 2\)), comparing the function value at each point with the objective function value of the previous point.
d) If $f(X) > q$, put $(X, f(X))$ into $C_i$ and let $q = f(X)$; repeat the same procedure for next design space coordinate.

if $f(X) < q$, a turning point is identified and the search continues in the opposite direction. The same procedures are carried out in all design variable coordinate directions until all boundary points are identified.

e) Subtract the identified unimodal region $C_i$ from the design space $S$ until all of the points in the design space $S$ are visited. Specifically, let $S = S \setminus C_i$

The most promising region, $j$, at this instance is identified. In this region, $r$ sample points with accurate analysis/simulation function values exist, and $r \in p$.

5) Adding $p$ additional “expensive” points using LHD and calculating their values using the analysis/simulation function.

6) Searching and identifying the optimum over the approximate unimodal region using

- the $r + p$ sample data points, $\Phi_j = \{X_1, ... X_r, X_{r+1}, ..., X_{r+p}\}$ with accurate objective function value, $f(X), ..., f(X_{r+p})$.

- Introducing the three metamodels (QPF, RBF and Kriging) to the identified most promising region $j$.

7) Generating a large number of “cheap” sample data points, $X_i, i = 1, ..., q$ (roughly 15×$p$) in the focused region using LHD sampling

$$\Omega_j = \{X_1, X_2, ..., X_q\}, \quad X_i \in S^n$$  \hspace{1cm} (5-7)

Calculate the approximates using the metamodel with mixed surrogates, and find
the minimum:

\[ \hat{f}_{\text{min}}^{(j)} = \min \{ \hat{f}(X_1), \ldots, \hat{f}(X_q); (X_i, \hat{f}(X_i)) \in S^n \} \] (5-8)

8) Adaptively adjust the weight coefficients of the surrogate models, \( \alpha_{k1}, \alpha_{k2}, \alpha_{k3} \) to find the best tested combination of these weight coefficients, \( \alpha_1^*, \alpha_2^*, \alpha_3^* \) with the minimum \( \hat{f}_{\text{min}}^{(j)} \) value.

Examples of weight coefficient tests are:

<table>
<thead>
<tr>
<th>( k )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_{k1} )</td>
<td>0.2</td>
<td>0.5</td>
<td>0.25</td>
<td>0.4</td>
<td>0.7</td>
<td>0.15</td>
<td>0.45</td>
</tr>
<tr>
<td>( \alpha_{k2} )</td>
<td>0.5</td>
<td>0.25</td>
<td>0.6</td>
<td>0.4</td>
<td>0.2</td>
<td>0.7</td>
<td>0.15</td>
</tr>
<tr>
<td>( \alpha_{k3} )</td>
<td>0.3</td>
<td>0.25</td>
<td>0.15</td>
<td>0.2</td>
<td>0.1</td>
<td>0.15</td>
<td>0.4</td>
</tr>
</tbody>
</table>

9) Update the mixed approximate model \( \hat{f}_{\text{mixed}}(X_k) \)

- the best performing surrogate model weight coefficients, \( \alpha_{k1}^*, \alpha_{k2}^*, \alpha_{k3}^* \) and the metamodel with mixed surrogates given by Equation (5-1) is used to represent the computation model in generating more promising points iteratively and to find the global optimum of this promising region.
- If \( S \neq \emptyset \), let \( l = l + 1 \); go to step 4), move to the next most promising region on the list to continue the search.

10) Repeat the previous steps until the local optima of all promising unimodal regions are identified. The global optimum is then determined by comparing the values of all local optima.

\[ \hat{f}_{\text{min (global)}} = \min(\hat{f}_{\text{min1}}, \hat{f}_{\text{min2}}, \ldots, \hat{f}_{\text{minj}}) \quad j = 1, \ldots, N \] (5-9)

where \( N \) is the number of unimodal regions processed.

The flow chart of the proposed method is shown in Figure 5-1.
Figure 5-1. The Flow Diagram of the Proposed Algorithm
5.4 Test Problems

To illustrate the advantage of the newly introduced approach over other widely used region elimination and space exploration global optimization methods, computation tests were carried out using a variety of benchmark global optimization problems. These benchmark problems are representative in terms of their structure and dimensionality. Both low and high dimensional, benchmark problems are used. Performance of the proposed method is also compared against other space exploration and region elimination common optimization methods such as DIRECT, and MPS, to gauge the proposed approach advantages and capabilities.

5.4.1 Summary of Test Results

In this section a summary of the obtained test results, summarized and shown in Table 5-1 to Table 5-4, are presented. The benchmark test problems that are frequently used in literature to assess the performance of global search algorithms, including the Six-hump camel-back, Shubert, Levy, Hartmann H6, Trid, and Hartmann H16 function, are used, and their forms are given in Appendix A. These test problems are representative in problem dimensions and shape characteristics of the objective function. The optimization tests are carried out by fitting a different mix of the three surrogates, QPF, RBF and KRG, to demonstrate the advantage of using mixed surrogates as the approximate model over the region of search.
Table 5-1. Optimization Test Results Using Mixed Surrogates, Quadratic Response Function and Kriging.

<table>
<thead>
<tr>
<th>Test Problem</th>
<th>Analytical F. Val.</th>
<th>Mixed QPF and KRG</th>
<th>Calculated Optimum Point</th>
<th>F. Val.</th>
<th>#F. Evl.</th>
<th>$\alpha_1^*$</th>
<th>$\alpha_3^*$</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC</td>
<td>-1.03160</td>
<td></td>
<td></td>
<td>0.0871</td>
<td>-0.7125</td>
<td>-1.03160</td>
<td>110</td>
<td>0.39</td>
</tr>
<tr>
<td>Shubert</td>
<td>-186.731</td>
<td></td>
<td></td>
<td>-1.4252</td>
<td>5.4829</td>
<td>-186.7309</td>
<td>247</td>
<td>0.35</td>
</tr>
<tr>
<td>Levy</td>
<td>0.00000</td>
<td></td>
<td></td>
<td>1.0147</td>
<td>1.0168</td>
<td>0.9825</td>
<td>1.0012</td>
<td>0.0005</td>
</tr>
<tr>
<td>Levy</td>
<td>-3.32200</td>
<td></td>
<td></td>
<td>0.2173</td>
<td>0.1489</td>
<td>0.4797</td>
<td>0.6578</td>
<td>-3.3186</td>
</tr>
<tr>
<td>H6</td>
<td>-50.0000</td>
<td></td>
<td></td>
<td>5.9955</td>
<td>9.9955</td>
<td>11.9877</td>
<td>11.9781</td>
<td>-49.9997</td>
</tr>
<tr>
<td>H6</td>
<td>25.8750</td>
<td></td>
<td></td>
<td>-0.4506</td>
<td>-0.4972</td>
<td>-0.4403</td>
<td>-0.5052</td>
<td>26.1754</td>
</tr>
</tbody>
</table>

Note: F. Val=Objective Function Value; #F. Eval=Number of Objective Function Evaluations; $\alpha_1^*$, $\alpha_2^*$ and $\alpha_3^*$ are QPF, RBF and KRG Weight Coefficients Respectively

Table 5-2. Optimization Test Results using Mixed Surrogates, Radial Basis Function and Kriging.

<table>
<thead>
<tr>
<th>Test Problem</th>
<th>Analytical F. Val.</th>
<th>Mixed RBF and KRG</th>
<th>Calculated Optimum Point</th>
<th>F. Val.</th>
<th>#F. Evl.</th>
<th>$\alpha_2^*$</th>
<th>$\alpha_3^*$</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC</td>
<td>-1.03160</td>
<td></td>
<td></td>
<td>-0.0869</td>
<td>0.7099</td>
<td>-1.03150</td>
<td>148</td>
<td>0.43</td>
</tr>
<tr>
<td>Shubert</td>
<td>-186.731</td>
<td></td>
<td></td>
<td>-1.4252</td>
<td>5.4829</td>
<td>-186.7309</td>
<td>247</td>
<td>0.33</td>
</tr>
<tr>
<td>Levy</td>
<td>0.00000</td>
<td></td>
<td></td>
<td>1.0109</td>
<td>0.9529</td>
<td>0.9624</td>
<td>0.9407</td>
<td>0.0021</td>
</tr>
<tr>
<td>H6</td>
<td>-3.32200</td>
<td></td>
<td></td>
<td>0.1974</td>
<td>0.1527</td>
<td>0.4798</td>
<td>0.6543</td>
<td>-3.3213</td>
</tr>
<tr>
<td>Trid</td>
<td>-50.0000</td>
<td></td>
<td></td>
<td>6.0342</td>
<td>10.0825</td>
<td>12.1349</td>
<td>12.0746</td>
<td>-49.9903</td>
</tr>
<tr>
<td>H16</td>
<td>25.8750</td>
<td></td>
<td></td>
<td>-0.5473</td>
<td>-0.5325</td>
<td>-0.5194</td>
<td>-0.6131</td>
<td>26.1594</td>
</tr>
</tbody>
</table>
### Table 5-3. Optimization Test Results using Mixed Surrogates, Quadratic Response Function and Radial Basis Function.

<table>
<thead>
<tr>
<th>Test Problem</th>
<th>Analytical F. Val.</th>
<th>Mixed QPF and RPF</th>
<th>Calculated Optimum Point</th>
<th>F. Val.</th>
<th>F. Evl.</th>
<th>$\alpha_1^*$</th>
<th>$\alpha_2^*$</th>
<th>$\alpha_3^*$</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC</td>
<td>-1.03160</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0544</td>
<td>-0.6753</td>
<td>-1.0172</td>
<td></td>
<td></td>
<td>0.69</td>
<td>0.31</td>
<td>0.0433</td>
<td></td>
</tr>
<tr>
<td>Shubert</td>
<td>-186.731</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.4252</td>
<td>5.4829</td>
<td>-186.7309</td>
<td></td>
<td></td>
<td>0.54</td>
<td>0.46</td>
<td>6.0e-06</td>
<td></td>
</tr>
<tr>
<td>Levy</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0130</td>
<td>0.9714</td>
<td>0.9741</td>
<td></td>
<td></td>
<td>0.0004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H6</td>
<td>-3.32200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1848</td>
<td>0.4760</td>
<td>0.2810</td>
<td></td>
<td></td>
<td>-3.3155</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trid</td>
<td>-50.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.0006</td>
<td>9.9769</td>
<td>11.9817</td>
<td></td>
<td></td>
<td>-49.9996</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H16</td>
<td>25.8750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.5101</td>
<td>-0.5073</td>
<td>-0.5066</td>
<td></td>
<td></td>
<td>25.9565</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 5-4. Optimization Test Results using Mixed Surrogates Quadratic Response Functions, Radial Basis Function and Kriging.

<table>
<thead>
<tr>
<th>Test Problem</th>
<th>Analyt. F. Val.</th>
<th>Mixed QPF, RBF and KRG</th>
<th>Weight Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Calculated Optimum Point</td>
<td>F. Val.</td>
</tr>
<tr>
<td>SC</td>
<td>-1.03160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shubert</td>
<td>-186.731</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Levy</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H6</td>
<td>-3.32200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trid</td>
<td>-50.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H16</td>
<td>25.8750</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


5.5 Results Discussion

Results obtained from the presented tests showed that the newly introduced approach (MSSE) in which mixed surrogates is introduced into the search space has performed comparably well. The new algorithm is capable of handling and solving the challenging benchmark problems in global optimization; locating the optimum with comparable accuracy and obtaining the results with reduced number of function evaluations and less computation cost compared to other methods.

To show the advantage of introducing mixed surrogates that is introducing more than one approximate model to the promising regions, results from four different scenarios were introduced. These scenarios include mixing QPF and Kriging, RBF and Kriging, QPF and RBF, and finally mixing QPF, RBF and Kriging. The obtained results are shown in Table 5-1 to Table 5-4 for a variety of benchmark test problems in terms of their dimensionality, surface texture of the function, and possibility of having many global optima.

5.5.1 Convergence Capability and Accuracy

The new approach can easily converge to global solutions with comparable number of function evaluations and computation cost. The convergence accuracy is really impressive. The convergence speed is acceptable as can be seen in Figure 5-2 to Figure 5-9. Figure 5-2 shows that fitting QPF, RBF, and Kriging on SC function produced an accurate solution and converged to the global optimum in a comparable number of function evaluations. Also fitting QPF and Kriging produced almost same results. It is noticeable that introducing three surrogates make the search starts from a point that is
close to the global optimum as can be seen in most of the figures presented in this section. For H6 function the three mixed surrogates outperform the other scenarios and produced very accurate solutions and shows quick convergence (Figure 5-3). An outstanding performance with Levy functions when three surrogates were used as can be seen in Figure 5-5. Again, good accuracy and speed of convergence were noticeable. In some cases the proposed algorithm needed more number of function evaluations to converge, but accurate global optima were obtained as can be seen in Figure 5-4. The same applies to H16 test problem; again the global optimum was obtained accurately (Figure 5-7). We also noticed that the combination of surrogates starts the search somewhere away from the global optima but when mixing three surrogates used, the search starts from a starting point that is close to the global optimum.

![Figure 5-2. Convergence Trends of Mixed Surrogates for SC Test Problem](image-url)
Figure 5-3. Convergence Trends of Mixed Surrogates for H6 Test Problem

Figure 5-4. Convergence Trends of Mixed Surrogates for Shubert Test Problem
Figure 5-5. Convergence Trends of Mixed Surrogates for Levy Test Problem

Figure 5-6. Convergence Trends of Mixed Surrogates for Trid Test Problem
Figure 5-7. Convergence Trends of Mixed Surrogates for H16 Test Problem

Figure 6-8 and Figure 6-9 shows different optimization runs on both Hartmann H6 and Trid benchmark test problems when QPF, RBF, and Kriging are used to fit candidate design points in the reduced design space. In all five runs the algorithm converges to the global optimum with comparable accuracy and convergence speed. The algorithm sample randomly the design space which means a different solution for every run is possible but Figure 5-8 and Figure 5-9 show that the proposed method can comparably converge to the same global optima or nearby. In summary, the convergence accuracy is considered to be acceptable based on the results obtained from testing different benchmark problems.
Figure 5-8. Optimization Runs on H6 Test Problem using Mixed Surrogates, Quadratic Polynomial Function (QPF), Radial Basis Function (RBF) and Kriging.

Figure 5-9. Optimization Runs on Trid Test Problem using Mixed Surrogates, Quadratic Polynomial Function (QPF), Radial Basis Function (RBF) and Kriging.
5.5.2 Performance Comparison

The results were compared with other optimization algorithms and found to be excellent and acceptable. Table 5-6 shows the comparison results of the newly introduced algorithm (MSSE) with both Mode Pursuing Sampling (MPS) and Dividing Rectangular (DIRECT) global optimization algorithms. The new approach performs better than MPS and DIRECT and locates the global minimum with less number of function evaluations. Benchmark test problems such as Six-hump Camel-back (SC), Levy and Hartmann (H6) functions were selected, tested and compared. The new approach can locate the global minimum in most of the test cases, even for functions with irregular shape and structure.

All computations were carried out in MATLAB environment. The advanced global optimization programs are either downloaded from the referenced sites or based on the search codes written by the author of this dissertation. The calculations were done using a regular PC workstation. Number of objective function evaluation is used to measure the relative performance of the search methods to avoid the dependence of computing platform and to address the concerns of complex engineering design applications with computationally intensive black-box objective functions.

In summary, it is hard to judge what the combination that gives the best surrogate models is. It might depend on the nature of the objective function and the constraint functions. However, from the test problems used in this dissertation, the combination of RBF and QPF did well. But it got even better when the three surrogates are used. The algorithm starts its searching process from places that are close to the global optimum and converges accurately to the global optimum.
Using region elimination strategy is very effective because the search is being carried out in the identified promising regions not in the original design space. These promising regions can be efficiently explored by introducing mixed approximate models (surrogates). It is found that when mixed approximate models are introduced to the design space, accurate solutions can be obtained with less computational cost.

Table 5-5. Summary of Benchmark Test Results of the Proposed Algorithm

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SC</td>
<td>-1.0316</td>
<td>-1.0316</td>
<td>110</td>
<td>-1.0315</td>
<td>148</td>
<td>-1.0172</td>
<td>104</td>
<td>-1.0316</td>
<td>110</td>
</tr>
<tr>
<td>Shubert</td>
<td>-186.7309</td>
<td>-186.7309</td>
<td>247</td>
<td>-186.7309</td>
<td>247</td>
<td>-186.7309</td>
<td>247</td>
<td>-186.7308</td>
<td>386</td>
</tr>
<tr>
<td>Levy</td>
<td>0.0000</td>
<td>0.0005</td>
<td>206</td>
<td>0.0021</td>
<td>432</td>
<td>0.0004</td>
<td>153</td>
<td>0.0007</td>
<td>150</td>
</tr>
<tr>
<td>Trid</td>
<td>-50</td>
<td>-49.9997</td>
<td>250</td>
<td>-49.9903</td>
<td>250</td>
<td>-49.9996</td>
<td>219</td>
<td>-49.9997</td>
<td>369</td>
</tr>
</tbody>
</table>

Figure 5-10. The Proposed Algorithm Performance Comparison for Different Surrogates
Table 5-6. Performance Comparison on Selected Benchmark Test Problems

<table>
<thead>
<tr>
<th>Benchmark Test Problem</th>
<th>Optimization Method</th>
<th>Analytical minimum</th>
<th>Global Optimum</th>
<th># Fun. Evl.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Global minimum x*</td>
<td>Optimum f*</td>
<td></td>
</tr>
<tr>
<td>Six-hump Camel-back Function (SC) n=2</td>
<td><strong>MSSE</strong></td>
<td>-1.0316</td>
<td>(0.0863, -0.7031)</td>
<td>-1.0316</td>
</tr>
<tr>
<td></td>
<td><strong>MPS</strong></td>
<td>-1.0316</td>
<td>(0.088, -0.7120)</td>
<td>-1.0316</td>
</tr>
<tr>
<td></td>
<td><strong>DIRECT</strong></td>
<td>-1.0316</td>
<td>(0.0903, -0.7157)</td>
<td>-1.0136</td>
</tr>
<tr>
<td>Levy Function n=4</td>
<td><strong>MSSE</strong></td>
<td>0.0000</td>
<td>(1.0130, 1.0174, 0.9714, 0.9741)</td>
<td>0.0004</td>
</tr>
<tr>
<td></td>
<td><strong>MPS</strong></td>
<td>0.0000</td>
<td>(0.9910, 1.0276, 0.9407, 0.9954)</td>
<td>0.0008</td>
</tr>
<tr>
<td></td>
<td><strong>DIRECT</strong></td>
<td>0.0000</td>
<td>(1.0000, 0.9999, 0.9999, 1.0000)</td>
<td>0.0000</td>
</tr>
<tr>
<td>Hartmann (H6) Function n=6</td>
<td><strong>MSSE</strong></td>
<td>-3.322</td>
<td>(0.2035, 0.1483, 0.4772, 0.2740, 0.3123, 0.6595)</td>
<td>-3.3221</td>
</tr>
<tr>
<td></td>
<td><strong>MPS</strong></td>
<td>-3.322</td>
<td>(0.2018, 0.1538, 0.4783, 0.2762, 0.3119, 0.6577)</td>
<td>-3.3221</td>
</tr>
<tr>
<td></td>
<td><strong>DIRECT</strong></td>
<td>-3.322</td>
<td>(0.4053, 0.8827, 0.8704, 0.5741, 0.1049, 0.0391)</td>
<td>-3.1999</td>
</tr>
</tbody>
</table>

Figure 5-11. Performance Comparison with Other Optimization Algorithms
5.5.3 The Proposed Algorithm’s Limitations

I am working on expanding the proposed algorithm to handle multi-objective optimization problems. Also practice implementations of the proposed algorithm in real life will be carried out. Because of the way that this approach is using that is dividing the design space into sub-design spaces. Handling high dimensional problems with high number of design variables might be a true challenge. Overlapping between these subspaces is a possibility but it will not affect the robustness of the method. In fact, it will ensure that all design space was searched.

5.6 Conclusions

In this chapter, a new global optimization algorithm, namely Mixed Surrogate and Space Exploration (MSSE) method, is introduced. The approach identifies three representative surrogate models based upon their performance under different types of optimization problems. The three surrogate models are used jointly with equal initial weights in this method. Change of their weighting factors are made based upon the accessed search efficiency of each of these models. The method avoids the need of selecting appreciate surrogates for a design optimization problem and applies an adaptive scheme to identify the most appropriate surrogate model in metamodel based design optimization. Dynamic adjustment of the weighting coefficients can be done automatically in implementation.

In addition, the approach divides the field of interest into several unimodal regions using design experiment data; identifies and ranks the regions that likely contain the global minimum; fits a mixed surrogates over each promising region with additional design experiments data points using Latin Hypercube Designs; identifies its minimum
and removes the processed region; and then moves to the next most promising region until all promising regions are processed and the global optimum is identified. By avoiding redundant searches, the approach identifies the global optimum with reduced number of objective function evaluations and computation efforts. The new algorithm was tested using a variety of benchmark test problems for global optimization and compared with several widely used region elimination and space exploration global optimization algorithms. The experiment results showed robust performance, comparable search accuracy and good computation efficiency, making the new method an excellent tool for computation intensive, computer analysis/simulation and black-box function based global design optimization problems.

6.1 Introduction

Product design and development requires that engineers consider trade-offs between product attributes in the areas of cost, weight, manufacturability, quality, and performance. The “optimum” design is in fact usually one in which compromises are acceptable, but understanding the impact of design decisions on all relevant attributes is tremendously difficult. Engineers are faced with the difficult challenge of determining how to arrive at the best overall design, making the right compromises, and not sacrificing critical attributes, like safety.

The use of approximation models in multi-objective optimization problems that involve expensive analysis and simulation processes, such as finite element analysis (FEA) and computational fluid dynamics (CFD), has became more popular and more attractive. Approximation models are found to be a promising tool for multi-objective optimization problems because of their efficiency in imitating the real model besides their compatibility for intensive computation problems. Many optimization applications
involve fitness evaluations that are expensive to perform. This becomes even worse when multiple objective black-box functions needs to be evaluated. In this chapter, a new adaptive multi-objective optimization approach based meta-modeling techniques is introduced.

This algorithm adaptively selects the right approximate model from three offered approximate models based on a selection criterion which will be explained in this chapter. The algorithm adaptively fits the right metamodel to identify the Pareto frontier for the multi-objective problem. Three metamodels are used in this work, QPF, RBF and Kriging. Classical sampling techniques suffer from the curse of dimensionality that is as the number of design variables increases the sampling points needed to explore the design space increases exponentially. For this reason, Latin Hypercube Designs (LHD) is used in this chapter. LHD explores the design space away better than classical sampling designs. It covers the design space equally and samples randomly. These contribute largely towards reducing the computation cost and increasing the chances of identifying the Pareto frontier for MOO problems. The proposed approach can easily identify the Pareto front for multi-objective optimization problems with high accuracy. The computation cost associated with identifying the Pareto front for expensive black-box functions is reduced. The proposed approach was tested using benchmark test problems and practical examples. The obtained results are found to be promising and encouraging.

6.2 Statement of the Multi-objective Optimization Problem

Multi-objective optimization can be defined as a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of the performance
criteria which are usually in conflict with each other. Hence, the term “optimize” means finding such a solution which would give the values of all objective functions acceptable to the designer. Formally, we can state that as follows

Find the vector

\[ \bar{x}^* = [x_1^*, x_2^*, ..., x_n^*]^T \quad (6-1) \]

which will satisfy the \( m \) inequality constraints

\[ g(\bar{x}) \geq 0, \quad i = 1, 2, ..., m \quad (6-2) \]

and the \( p \) equality constraints

\[ h(\bar{x}) = 0, \quad i = 1, 2, ..., p \quad (6-3) \]

to optimize the vector function

\[ \bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), ..., f_k(\bar{x})]^T \quad (6-4) \]

where \( x = [x_1, x_2, ..., x_n]^T \) is the vector of decision variables. In other words, we wish to determine from among the set \( f \) of all numbers which satisfy Equations (6-2) and (6-3) the particular set \( x_1^*, x_2^*, ..., x_k^* \) which yields the optimum values of all the objective functions.

### 6.3 Pareto Frontier and Pareto Efficiency

The Pareto frontier or Pareto set, named after the economist Vilfredo Pareto, is the set of all Pareto efficient outcomes. A vector of \( x^* \) is Pareto optimal if there exists no feasible vector \( x \) that would decrease some objective function without causing a
simultaneous increase in at least one objective function. Mathematically, the Pareto optimality can be defined as follows

A vector of \( x^* \) is a Pareto optimum if and only if, for any \( x \) and \( i \),

\[
f_j(x) \leq f_j(x^*), \quad j = 1, \ldots, m; \quad j \neq i, \quad \Rightarrow f_i(x) \geq f_i(x^*)
\] (6-5)

In multi-objective optimization (MOO), the main task is to identify Pareto set points so that a Pareto frontier can be identified. It is hard to claim that these points are in the Pareto set or Pareto frontier unless there is a convincing way to judge whether these points are in the Pareto set or not. That could be easily decided if a fitness function is adapted.

### 6.4 Fitness Function

The fitness function used in this work is defined in Equation (6-6). This function firstly appeared in [143]. The fitness function is defined as:

\[
G_i = \left[ 1 - \max \left( \min \left( f_{s_{i1}}^i - f_{s_{i1}}^j, f_{s_{i1}}^i - f_{s_{i1}}^j, \ldots, f_{s_{im}}^i - f_{s_{im}}^j \right) \right) \right]^l
\] (6-6)

where, \( G_i \) denotes the fitness value of the \( i \)th design; \( f_{sk}^i \) is scaled to \( k \)th objective function value of the \( i \)th design, \( k = 1, \ldots, m \); and \( l \) is called the frontier exponent. In this work, \( l \) is taken as a constant 1. The max in Equation (6-6) is over all other designs \( j \neq i \) in the set and the min is over all the objectives. The objectives \( f_{s_{i1}}, f_{s_{i2}}, \ldots, f_{s_{im}} \) in Equation (6-7) are scaled to a range \([0, 1]\). For example for \( f_{s_{i1}} \),

\[
f_{s_{i1}} = \frac{rawf_{i1} - rawf_{i1,min}}{rawf_{i1,max} - rawf_{i1,min}}
\] (6-7)
where, $rawf_{1,i}$ denotes un-scaled value of the first objective for the $i$th design; $rawf_{1,\text{min}}$ denotes the maximum $f_{s1}$ un-scaled value of the first objective among all designs; and $rawf_{1,\text{min}}$ denotes the minimum un-scaled value of the first objective among all designs. In case that an objective function is a constant, the scaled objective function value is taken as 1 in this work.

6.5 The Proposed Approach

The main objective of the proposed algorithm is to efficiently identify the Pareto frontier with less computation cost for costly black-box functions. This goal can be met if effective and efficient tools are employed. LHD sampling technique is used in this work. Every time we sample or generate new sample points, the Pareto frontier should be approached time after. The question is how we know if we are approaching the Pareto frontier. A sampling guide function is needed to judge whether these sample points are close or on the Pareto frontier or not. In the next section, a sampling guidance function is introduced.

6.6 Sampling Guidance Function

There are many statistical methods available nowadays that can be used to generate sample points based on a given probability density function (PDF). Wang et al. [16] were inspired by Fu and Wang [144] and developed a sampling method called mode pursuing sampling method (MPS) in which they start by constructing an approximation model from a few sample points, and then generate a large number of points using the approximate model, sorts the points, and constructs a cumulative function analogous to cumulative density function (CDF) proposed in [144] by adding up all the function values
before the current point in the sorted point set. A sample is then drawn from the point set according to this cumulative function. This will ensure that more new sample points are generated around the current minimum in the region of interest and less in other regions in the design space. Then, the variation of the objective function in MPS is considered as a sampling guidance function. This sampling guidance function is adapted and used in this work.

6.7 Adaptive Approximation Models Fitting

After generating few sample points in the space of interest using latin hypercube designs, three metamodels are constructed. These are QPF, RBF and Kriging. The root mean square error (RSME) of these metamodels is measured. The metamodel with less RSME value is selected to be constructed in the next steps. It is known that Kriging always gives less RSME than QPF and RBF. In this approach, a rule was set to guide the metamodel constructing process that is to check RMSE for QPF and RBF first. If RSME is greater than δ, set equal to 5 in this work, then construct Kriging. If the opposite is true then construct either QPF or RBF based on RSME.

6.8 Termination Criteria

In this work, the maximum number of iterations is set to be one of the termination criteria besides measuring the difference between frontier points after two consecutive iterations. If the difference is sufficiently small, then the algorithms should terminate.

6.9 Steps of the Proposed Approach

The proposed approach steps are summarized and explained in the following steps:

1) Sampling initial random design points.
Generating few sample points that are randomly distributed in the design space. LHD is used carry out the sampling process.

2) The generated sample points are evaluated using the black-box function (expensive function) to identify the current frontier points.

3) Calculating the root mean square error (RMSE) of the constructed metamodels.

In this step, three different approximate models (metamodels) are introduced, quadratic polynomial function (QPF), radial basis function (RBF) and Kriging (KRG) metamodel. These metamodels are fitted to the previously generated sample points and the RMSE is calculated. Based on the RSME value, the algorithm adaptively selects the metamodel that have less RMSE value compared to the other two values of RMSE. This metamodel should be used in the next steps. It is known that Kriging most of the times gives low RMSE values. In this case a comparison between the RMSE values of QPF and RBF functions are carried out and the one with better accuracy will be used. If the RSME values of both functions QPF and RBF were a bit high (greater than 5), the algorithm selects Kriging as a metamodel that should be used in the next steps.

4) Generate a large number of cheap points. After the approximation model is constructed, they are used to evaluate the large number of sample points which were generated using LHD.

5) Combining the sample points. In this step all generated sample points (expensive and cheap points) are combined together to identify or move towards the preliminary approximated Pareto frontier.
6) Finding the best points among all the existing points (points obtained in the previous step). These points have high probability of becoming Pareto set points.

7) Function evaluation for new sample points. These points were obtained from step 5 and were evaluated using the expensive black-box functions.

8) Combining new sample points with expensive frontier points to identify of the new frontier set.

9) Checking convergence criteria, if convergence criteria are met, the procedure terminates, otherwise go back to step 4.

The flow diagram of the proposed algorithm is shown in Figure 6-1.

6.10 Numerical Examples and Results

The proposed method is tested with a number of well-known multi-objective optimization benchmark test problems found in the literature. The results of four test problems which were tested are shown in this section. Also a practical example is optimized and the obtained results are also presented in this section.

6.10.1 Test Problem 1: Kita’s Function

The problem was proposed by Kita et al. [145]. It is a two-objective function problem with two decision variables.

Maximize  \( F = (f_1(x_1,x_2), f_2(x_1,x_2)) \)

Where

\[
 f_1(x_1,x_2) = -x_1^2 + x_2; \quad f_2(x_1,x_2) = \frac{1}{2}x_1 + x_2 + 1;
\]

Subject to

\[
 0 \geq \frac{1}{6}x_1 + x_2 - \frac{13}{2}; \quad 0 \geq \frac{1}{2}x_1 + x_2 - \frac{15}{2}; \quad 0 \geq 5x_1 + x_2 - 30; \quad 5 \geq x_1, x_2 \geq -5
\]
Figure 6-1. Flowchart of the Proposed Algorithm
6.10.2 Test Problem 2: Veldhuizen and Lamont’s Function

Veldhuizen and Lamont’s MOP3 [146] is a three-objective function of two decision variables.

\[ \text{Minimize } F = (f_1(x_1, x_2), f_2(x_1, x_2), f_3(x_1, x_2)) \]

where

\[ f_1(x_1, x_2) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2); \]

\[ f_2(x_1, x_2) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15; \]

\[ f_3(x_1, x_2) = \frac{1}{(x_1^2 + x_2^2 + 1)} - 1.1 \exp(-x_1^2 - x_2^2); \]

subject to:

\[ -3 \leq x_1, x_2 \leq 3 \]

This test function has a disconnected Pareto optimal set, and the Pareto front is a curve following a convoluted path through objective space.

6.10.3 Test Problem 3: Comet Function

This function is known as the Comet function [147].

\[ \text{Minimize } F = (f_1(x_1, x_2, x_3), f_2(x_1, x_2, x_3), f_3(x_1, x_3)) \]

where

\[ f_1(x_1, x_2, x_3) = (1 + x_3)(x_1^3x_2^3 - 10x_1 - 4x_2); \]

\[ f_2(x_1, x_2, x_3) = (1 + x_3)(x_1^3x_2^3 - 10x_1 - 4x_2); \]

\[ f_3(x_1, x_3) = 3(1 + x_3)x_1^2; \]

subject to:
1 ≤ x₁ ≤ 3.5;  -2 ≤ x₂ ≤ 2;  x₃ ≥ 0

6.10.4 Test Problem 4: The Concave Pareto Front Function

This function is a two-objective function of two decision variables [78].

\[ \text{Maximize } F = (f₁(x₁, x₂), f₂(x₁, x₂)) \]

Where

\[ f₁(x₁, x₂) = (x₁ - 2)^2 + (x₂ - 1)^2; \]
\[ f₂(x₁, x₂) = x₁^2 + (x₂ - 6)^2; \]

Subject to

\[ x₁ - 1.6 ≤ 0; \quad 0.4 - x₁ ≤ 0; \quad x₂ - 5 ≤ 0; \quad 2 - x₂ ≤ 0 \]

The problem has a concave Pareto front.

6.10.5 Test Problem 5: Practical Example

The practical application is the fuel cell optimization problem proposed by Shan and Wang in [83]. It is a multiple function panel for fuel cell component design. The panel has three functions: (1) To function as a distributed spring system to compensate the enormous amount of hydro and thermal expansions during the fuel cell operation; (2) to function as a radiator to get rid of the heat generated during the operation; and (3) to function as a conductor to collect electrons for a stack of many fuel cells. The objective is to find the optimum design parameters which optimize the panel component dimensions height, width, and thickness (h, w, and t). The objective functions and the constraints are as follows

\[ \text{Minimize } f₁(h, w, t) = h - \sqrt\left( \left( \frac{h}{\sinθ} - u \right)^2 - \frac{1}{4}w^2 - 0.034 \right) / 0.034; \]
Maximize \( f_2(h, w, t) = \text{findht}(h, w, t); \)

subject to

\[ f_2(h, w, t) \geq 340; \quad 4 \leq h \leq 9; \quad 4 \leq w \leq 9; \quad 0.06 \leq t \leq 6 \]

where \( = 2h/w, u = Pwh/2Et \sin^2 \theta \), \( f_1 \) denotes the ratio of the difference between the panel’s real deformation and the ideal deformation over the ideal deformation under a given pressure \( P \). \( E \) is the Young’s modulus of copper. \( f_2 \) is the exit air temperature from the panel, which is to be maximized for the radiation purpose. \( \text{findht} \) is a program that finds the exit air temperature, which can be considered as a black-box function.

### 6.11 Results and Discussions

The proposed approach is tested with a number of well-known MOO problems and in real engineering design problem. Promising results were obtained using the proposed approach. Many optimization runs were carried out (more than 10 runs). From Table 6-1 to Table 6-5, samples of the results obtained from selected 5 runs are presented. For test problem 1 as shown in Table 6-1, the number of iterations variation range is (63-74), the median number of iterations is 68. The variation range of the number of the function evaluations is (69-80) and the median is 69. KRG metamodel is the winner in most of the test runs. The median number of converged Pareto points is 40. Good CPU time is achieved when RBF is constructed. These results show the reduction in computational cost of the proposed method.
### Table 6-1. Test Results of the Proposed Approach on Test Problem 1

<table>
<thead>
<tr>
<th>Run#</th>
<th># iterations</th>
<th># Evaluations</th>
<th># Pareto points</th>
<th>CPU time (Sec.)</th>
<th>Metamodel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run#1</td>
<td>71</td>
<td>75</td>
<td>39</td>
<td>258.98</td>
<td>KRG</td>
</tr>
<tr>
<td>Run#2</td>
<td>74</td>
<td>80</td>
<td>41</td>
<td>262.79</td>
<td>KRG</td>
</tr>
<tr>
<td>Run#3</td>
<td>65</td>
<td>71</td>
<td>38</td>
<td>258.16</td>
<td>KRG</td>
</tr>
<tr>
<td>Run#4</td>
<td>66</td>
<td>77</td>
<td>38</td>
<td>226.37</td>
<td>RBF</td>
</tr>
<tr>
<td>Run#5</td>
<td>63</td>
<td>69</td>
<td>41</td>
<td>354.01</td>
<td>KRG</td>
</tr>
<tr>
<td>Run#6</td>
<td>71</td>
<td>77</td>
<td>42</td>
<td>343.37</td>
<td>KRG</td>
</tr>
</tbody>
</table>

### Table 6-2. Test Results of the Proposed Approach on Test Problem 2

<table>
<thead>
<tr>
<th>Run#</th>
<th># iterations</th>
<th># Evaluations</th>
<th># Pareto points</th>
<th>CPU time (Sec.)</th>
<th>Metamodel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run#1</td>
<td>71</td>
<td>85</td>
<td>33</td>
<td>291.63</td>
<td>RBF</td>
</tr>
<tr>
<td>Run#2</td>
<td>62</td>
<td>88</td>
<td>22</td>
<td>178.91</td>
<td>RBF</td>
</tr>
<tr>
<td>Run#3</td>
<td>44</td>
<td>53</td>
<td>24</td>
<td>115.23</td>
<td>QPF</td>
</tr>
<tr>
<td>Run#4</td>
<td>52</td>
<td>65</td>
<td>41</td>
<td>168.16</td>
<td>KRG</td>
</tr>
<tr>
<td>Run#5</td>
<td>43</td>
<td>57</td>
<td>29</td>
<td>168.96</td>
<td>KRG</td>
</tr>
<tr>
<td>Run#6</td>
<td>58</td>
<td>74</td>
<td>34</td>
<td>235.36</td>
<td>KRG</td>
</tr>
</tbody>
</table>

### Table 6-3. Test Results of the Proposed Approach on Test Problem 3

<table>
<thead>
<tr>
<th>Run#</th>
<th># iterations</th>
<th># Evaluations</th>
<th># Pareto points</th>
<th>CPU time (Sec.)</th>
<th>Metamodel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run#1</td>
<td>99</td>
<td>126</td>
<td>37</td>
<td>304.31</td>
<td>RBF</td>
</tr>
<tr>
<td>Run#2</td>
<td>77</td>
<td>93</td>
<td>43</td>
<td>206.65</td>
<td>KRG</td>
</tr>
<tr>
<td>Run#3</td>
<td>50</td>
<td>71</td>
<td>36</td>
<td>120.22</td>
<td>KRG</td>
</tr>
<tr>
<td>Run#4</td>
<td>64</td>
<td>84</td>
<td>41</td>
<td>204.15</td>
<td>KRG</td>
</tr>
<tr>
<td>Run#5</td>
<td>53</td>
<td>68</td>
<td>36</td>
<td>130.28</td>
<td>KRG</td>
</tr>
</tbody>
</table>

### Table 6-4. Test Results of the Proposed Approach on Test Problem 4

<table>
<thead>
<tr>
<th>Run#</th>
<th># iterations</th>
<th># Evaluations</th>
<th># Pareto points</th>
<th>CPU time (Sec.)</th>
<th>Metamodel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run#1</td>
<td>19</td>
<td>79</td>
<td>71</td>
<td>9.10</td>
<td>RBF</td>
</tr>
<tr>
<td>Run#2</td>
<td>16</td>
<td>74</td>
<td>63</td>
<td>7.86</td>
<td>RBF</td>
</tr>
<tr>
<td>Run#3</td>
<td>15</td>
<td>74</td>
<td>63</td>
<td>6.93</td>
<td>RBF</td>
</tr>
<tr>
<td>Run#4</td>
<td>17</td>
<td>89</td>
<td>78</td>
<td>7.93</td>
<td>QPF</td>
</tr>
<tr>
<td>Run#5</td>
<td>23</td>
<td>85</td>
<td>72</td>
<td>10.79</td>
<td>RBF</td>
</tr>
</tbody>
</table>
Table 6-5. Test Results of the Proposed Approach on Test Problem 5

<table>
<thead>
<tr>
<th>Run#</th>
<th># iterations</th>
<th># Evaluations</th>
<th>#Pareto points</th>
<th>CPU time (Sec.)</th>
<th>Metamodel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run#1</td>
<td>226</td>
<td>285</td>
<td>35</td>
<td>402.54</td>
<td>QPF</td>
</tr>
<tr>
<td>Run#2</td>
<td>203</td>
<td>263</td>
<td>38</td>
<td>479.89</td>
<td>QPF</td>
</tr>
<tr>
<td>Run#3</td>
<td>181</td>
<td>212</td>
<td>32</td>
<td>362.04</td>
<td>RBF</td>
</tr>
<tr>
<td>Run#4</td>
<td>129</td>
<td>164</td>
<td>37</td>
<td>389.32</td>
<td>QPF</td>
</tr>
<tr>
<td>Run#5</td>
<td>62</td>
<td>71</td>
<td>27</td>
<td>285.52</td>
<td>KRG</td>
</tr>
<tr>
<td>Run#6</td>
<td>90</td>
<td>102</td>
<td>33</td>
<td>806.14</td>
<td>KRG</td>
</tr>
<tr>
<td>Run#7</td>
<td>94</td>
<td>108</td>
<td>32</td>
<td>831.87</td>
<td>KRG</td>
</tr>
</tbody>
</table>

Figure 6-2 shows the performance space and evaluated points for test problem 1, how they are distributed and how these points are converging to the optimal Pareto frontier. Figure 6-3 shows how all evaluated points have converged to form the optimal Pareto frontier. Figure 6-4 and Figure 6-10 show that in all selected 5 runs the algorithm always identifies the same Pareto frontier. These also can be seen for all tested benchmark problems in Appendix C and specifically in Figures C1- C5.

![Figure 6-2. Performance Space and Evaluated Points for Test Problem 1](image-url)
Figure 6-3. Pareto Set Points for Test Problem 1

Figure 6-4. Pareto Frontier of Five Runs for Test Problem 1
Figure 6-5. Performance Space and Evaluated Points for Test Problem 2

Figure 6-6. Pareto Set Points for Test Problem 2
Figure 6-7. Performance Space and Evaluated Points for Test Problem 3

Figure 6-8. Performance Space and Evaluated Points for Test Problem 4
Figure 6-9. Pareto Frontier of Five Runs for Test Problem 5

Figure 6-10. Performance Space and Evaluated Point of Five Runs for Test Problem 5
### 6.12 Summary of Test Results

Table 6-6 shows a summary of the best results obtained from more than 10 runs. Table 6-7 shows the number of iterations, the number of objective function evaluations, the number of Pareto points in the optimal Pareto frontier, the CPU time took the algorithm to converge to the Optimal Pareto frontier and the metamodel used in that run to achieve that solution for all test problems. It is clear that RBF which is loyal to the sample points and Kriging which interpolates the sample points and produces good accuracy did a good job in identifying the efficient Pareto set with less computation cost and fairly number of Pareto set points. In Table 6-7, the variation range and the median of the number of iterations, number of function evaluations and Pareto set points are shown.

Since LHD randomly generates the sample points in the region of interest, more than 5 runs have been carried out for each test problem. These results represent the best result obtained from those different runs. The number of iterations, total number of evaluated points, number of converged frontier points, CPU time needed to identify the Pareto frontier and the metamodel used were recorded for 5 different runs.

**Table 6-6. Summary of the Best Test Results of the Proposed Approach**

<table>
<thead>
<tr>
<th>Test Problem</th>
<th>Number of Iterations</th>
<th>Number of Evaluations</th>
<th>Number of Pareto Set Points</th>
<th>CPU time (sec.)</th>
<th>Metamodel Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Problem 1</td>
<td>66</td>
<td>77</td>
<td>38</td>
<td>226.37</td>
<td>RBF</td>
</tr>
<tr>
<td>Test Problem 2</td>
<td>52</td>
<td>65</td>
<td>41</td>
<td>168.16</td>
<td>KRG</td>
</tr>
<tr>
<td>Test Problem 3</td>
<td>50</td>
<td>71</td>
<td>36</td>
<td>120.22</td>
<td>KRG</td>
</tr>
<tr>
<td>Test Problem 4</td>
<td>15</td>
<td>85</td>
<td>73</td>
<td>007.39</td>
<td>RBF</td>
</tr>
<tr>
<td>Test Problem 5</td>
<td>62</td>
<td>71</td>
<td>27</td>
<td>285.52</td>
<td>KRG</td>
</tr>
</tbody>
</table>
Table 6-7. Summary of Test Results of the Proposed Approach

<table>
<thead>
<tr>
<th>Test Problem</th>
<th>Number of Iterations</th>
<th>Number of Evaluations</th>
<th>Number of Pareto Set Points</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Range</td>
<td>Median</td>
<td>Range</td>
</tr>
<tr>
<td>Test Problem 1</td>
<td>[63-74] 68.5</td>
<td>[69-80] 76.0</td>
<td>[38-42] 40</td>
</tr>
<tr>
<td>Test Problem 2</td>
<td>[44-71] 55.0</td>
<td>[57-88] 69.5</td>
<td>[22-41] 31</td>
</tr>
<tr>
<td>Test Problem 3</td>
<td>[50-99] 64.0</td>
<td>[71-126] 84.0</td>
<td>[36-43] 37</td>
</tr>
<tr>
<td>Test Problem 4</td>
<td>[15-23] 17.0</td>
<td>[74-89] 79</td>
<td>[63-78] 71</td>
</tr>
</tbody>
</table>

It is noticeable that the total number of expensive points is small for all test problems. This is also reflected in the CPU time needed to find or form to the Pareto frontier. The CPU time was reduced. Comparably small number of iterations is required to perform the task. To better observe the accuracy of converged frontier points, all of the evaluated points are plotted together with the feasible performance space, as shown in Figure 6-2. It is clear from the plots that the converged frontier points are very close to the real Pareto frontier.

6.13 Conclusions

A new approach for identifying the efficient Pareto set for multi-objective optimization is introduced. The proposed approach adaptively chooses and fits the right metamodel to identify the optimal Pareto frontier for black-box functions. The proposed approach offers decision makers with a set of Pareto set for choices without prior knowledge of the objective function. Even when the frontier surface is highly nonlinear or discontinuous, the proposed approach obtains solutions that can reflect the entire Pareto optimal frontier. The proposed approach can identify the optimal Pareto frontier for most of the benchmark test problems with few numbers of expensive points’ evaluations and
iterations. Different multi-objective benchmark test functions and practical engineering multi-objective application were used to test the proposed approach and reveal its pros and cons. The obtained results showed that the proposed approach is efficient, robust, and accurate in identifying the Pareto front. Comparably promising results were obtained from the newly proposed approach.
Chapter 7. Automotive Magnetorheological Brake (MRB) System Design Optimization

7.1 Introduction

In this chapter, the superior performance of a novel space exploration and unimodal region elimination global optimization algorithm, SEUMRE, is demonstrated through comparisons with other well known global optimization techniques, including genetic algorithm (GA), simulated annealing (SA), and a highly nonlinear design problem – the optimal design of automotive magnetorheological brake (MRB). Unlike the conventional brakes, an MRB employs the interaction between a magnetorheological fluid and an applied magnetic field to generate the retarding braking torque. The SEUMRE design optimization algorithm was used to maximize the braking torque and minimize the weight of the brake structure. The computation time and optimized design parameters illustrated SEUMRE’s capability to converge to an accurate result faster than the conventional global optimization methods.

Many optimization algorithms have been introduced and almost every day there are new ideas and new algorithms evolve to improve the performance of such algorithms. Generally, optimization algorithms can be divided into two main categories, gradient
based and non gradient based [148, 149]. Engineering design problems are highly complex in nature. Therefore, deriving the gradient is very expensive; requires intensive computational efforts and resources. For that reason, non-gradient optimization algorithms, stochastic and heuristic optimization algorithms [137, 150], became attractive. These algorithms have become popular and have found a lot of interest because of their superior performance and their capability in extracting information on the search space without experiencing any computational complexities.

Despite the fact that many optimization algorithms available today to handle complex engineering problems for different applications, the need for efficient, effective and robust algorithms maintains pace in algorithm evolution and presents a true challenge. Thus, space exploration and region elimination optimization algorithms are introduced. These optimization algorithms start their search by spreading sampling points to explore the design space and based on the preliminary information obtained, they decide whether the global solution may exist and where the search must be focused. This made such optimization algorithms suitable for highly nonlinear and complex design optimization problems involving expensive analysis and simulation processes such as finite element analysis (FEA) and computational fluid dynamics (CFD).

SEUMRE was introduced recently by Younis and Dong [18]. In this chapter, a highly nonlinear and complex real-life engineering design problem is selected to be optimized using SEUMRE as well as other well known stochastic optimization algorithms, i.e. GA and SA. The main objective of this work is to compare the performances and the optimum results obtained by these global optimization algorithms.
A basic configuration of MRB was introduced by Park et al. [151]. As shown in Figure 7-1, in this configuration, a rotating disk (3) is enclosed by a static casing (5), and the gap (7) between the disk and casing is filled with the MR fluid (a typical MR fluid is a type of functional fluid which has a suspension of magnetic particles in inert carrier liquids). A coil winding (6) is embedded on the perimeter of the casing and when electrical current is applied to it, magnetic fields are generated, and the MR fluid in the gap becomes solid-like instantaneously. The shear friction between the rotating disk and the solidified MR fluid provides the controllable retarding torque and the magnitude of the shear friction is directly related to the applied magnetic flux density on the fluid.

![Figure 7-1. Basic MRB Design (Park et. al. [159])](image)

### 7.2 MRB Model

The MRB design optimization is based on the model for calculating the braking torque resulting from different MRB configurations. The idealized characteristics of the MR
fluid can be described effectively by using the Bingham plastic model [89, 90, 152, 153]. According to this model, the total shear stress $\tau$ is:

$$\tau = \tau_H \text{sgn}(\dot{\gamma}) + \mu_p \dot{\gamma}$$  \hspace{1cm} (7-1)

where $\tau_H$ is the yield stress due to the applied magnetic field, $\mu_p$ is the no-field plastic viscosity of the fluid and $\dot{\gamma}$ is the shear rate. The braking torque for the geometry shown in Figure 7-2 can be defined as follows:

$$T_b = \int_A \tau r dA = 2\pi N \int_{j}^{z} \left( \tau_H \text{sgn}(\dot{\gamma}) + \mu_p \dot{\gamma} \right) r^2 dr$$ \hspace{1cm} (7-2)

where $A$ is the working surface area (the domain where the fluid is activated by applied magnetic field intensity), $z$ and $j$ are the outer and inner radii of the disk, $N$ is the number of disks used in the enclosure and $r$ is the radial distance from the centre of the disk.

Assuming the MR fluid gap in Figure 7-2 to be very small (e.g., ~1 mm), the shear rate can be obtained by:

$$\dot{\gamma} = \frac{rw}{h}$$ \hspace{1cm} (7-3)

with the additional assumption of linear fluid velocity distribution across the gap and no slip conditions. In Equation (7-3), $w$ is the angular velocity of the disk and $h$ is the thickness of the MR fluid gap. In addition, the yield stress, $\tau_H$ can be approximated in terms of the magnetic field intensity applied specifically onto the MR fluid, $H_{MRF}$, and the MR fluid dependent constant parameters, $k$ and $\beta$, i.e.

$$\tau_H = kH_{MRF}^\beta$$ \hspace{1cm} (7-4)
By substituting Equations (7-3) and (7-4), the braking torque equation in Equation (7-2) can be rewritten as:

\[ T_b = 2\pi N \int_j^z \left( k H_{MRF}^b \text{sgn}(\gamma) + \mu_p \frac{rW}{h} \right) r^2 \, dr \]  \hspace{1cm} (7-5)

Then, Equation (7-5) can be divided into the following two parts after the integration:

\[ T_H = \frac{2\pi}{3} N k H_{MRF}^b (r_2^3 - r_j^3) \]  \hspace{1cm} (7-6)

\[ T_\mu = \frac{\pi}{2h} N \mu_p (r_2^4 - r_j^4)w \]  \hspace{1cm} (7-7)

where \( T_H \) is the torque generated due to the applied magnetic field and \( T_\mu \) is the torque generated due to the viscosity of the fluid. Finally, the total braking torque is \( T_b = T_\mu + T_H \). From a design point of view, the parameters that can be varied to increase the braking torque generation capacity are: the number of disks (i.e. \( N \)), the dimensions and configuration of the magnetic circuit (i.e. \( r_z, r_j \)) and other structural design parameters shown in Figure 7-3, and \( H_{MRF} \) that is directly related to the applied current density in the electromagnet and materials used in the magnetic circuit.

In order to solve for the braking torque, the magnetic field distribution over the MRB domain has to be calculated. Therefore, a nonlinear finite element model (FEM) for the MRB is created and, using this model, the braking torque generation for different configurations can be calculated. For the current comparison study of the global optimization algorithms, a one disk MRB configuration, shown in Figure 7-3, is adapted as the benchmark design.
7.3 Formulation of the Optimization Problem

The cross-section of the selected one disk configuration of the MRB is shown in Figure 7-2. As a next step, the chosen design configuration was optimized for higher braking torque and lower weight. In setting up such an optimization problem for the MRB, a cost function was defined by including the braking torque and weight as functions of the dimensional parameters of the magnetic circuit shown in Figure 7-3.

![Figure 7-2. Chosen MRB Based on the Design Criteria](image)
Figure 7-3. Dimensional Parameters Related to Magnetic Circuit Design

The objective function of the MRB optimization problem is defined as

\[
\text{Minimize } f(d) = k_w \frac{W}{W_{\text{ref}}} - k_t \frac{T_H}{T_{\text{ref}}}
\]  

(7-8)

where:

\[k_w + k_t = 1\]  

(7-9)

Subject to:

\[W < 150 \, N, \text{ and } d_{\text{brake}} < 240 \, mm\]  

(7-10)

\[W_{\text{ref}}\] was obtained considering the overall system weight of the CHB [30]. Moreover, since the braking torque generated by the proposed MRB configuration is comparably
less than that of the CHB, $T_{ref}$ was selected to be 20 Nm. This reference torque value was selected by checking a number of random MRB designs which satisfied the constraints.

As the constraints for the optimization problem, the weight of the actuator was set to be smaller than the weight of the CHB, i.e. $W < 150$ N. In addition to this, since the brake should fit into a wheel, the diameter of the MRB is set to be smaller than the inner diameter of the wheel. In this study, the brake is optimized for the standard 13” wheel whose inner diameter is 240 mm, i.e. $d_{brake} < 240$ mm.

Due to the multidisciplinary nature of the design problem, the Multidisciplinary Design Optimization (MDO) procedure is used in the calculation of the objective function for each design as illustrated in Figure 7-4. Detailed analysis for forming the MDO problem can be found in [154]. The optimization problem is then solved using the SEUMRE algorithm. To assess the performance of the algorithm, two other commonly used global optimization search methods, GA and SA, are also used as comparison tests. The results are presented and discussed in detail in the following section.

**7.4 Optimization Results and Discussions**

For a given MRB design configuration, COMSOL Multiphysics™, a commercial FE software package, was used to calculate the magnetic flux density within MRB. MATLAB was then used to solve for the weight and the braking torque using the given brake configuration and the FEA result. Each of the three global optimization programs has been run multiple times and the best optimum result is accepted and used as the measure for search efficiency, as given in Table 8-1. A PC with dual core Intel Xeon (E5140) processor and 8 GB memory is used during the design optimization.
Figure 7-4. Process of Computing the Cost Function for a Random Design

Table 7-1. The Calculated Optimum Parameters and Results Comparison

<table>
<thead>
<tr>
<th>Global Optimization Algorithm</th>
<th>Obtained Optimum Function Value</th>
<th>Obtained Optimum Design Parameters (cm)</th>
<th>Number of Iterations</th>
<th>CPU time (hrs)</th>
<th>Relative CPU time (hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulated Annealing (SA)</td>
<td>-1.3623</td>
<td>2.28, 3.99, 3.75, 0.27, 0.50, 2.35, 1.10, 1.97, 0.10, 1.27</td>
<td>2500</td>
<td>4.0348</td>
<td>1.344</td>
</tr>
<tr>
<td>Genetic Algorithms (GA)</td>
<td>-1.0998</td>
<td>3.84, 2.47, 5.01, 0.30, 0.76, 1.85, 1.15, 2.00, 0.10, 1.23</td>
<td>1100</td>
<td>3.002</td>
<td>1.000</td>
</tr>
<tr>
<td>Space Exploration (SEUMRE)</td>
<td>-1.1630</td>
<td>1.04, 2.92, 4.71, 0.15, 0.73, 2.15, 1.13, 1.82, 0.11, 1.34</td>
<td>115</td>
<td>1.145</td>
<td>0.381</td>
</tr>
</tbody>
</table>
The obtained optimum function values are -1.3623, -1.0998 and -1.1630 from SA, GA and SEUMRE, respectively. It appears that SA outperformed the rest in terms of the solution accuracy. However it took SA approximately 4.035 hours to reach the optimum solution with high accuracy. If we take the computation cost into consideration, SEUMRE performed well in comparison to SA and GA. Because GA is well known and widely used algorithm it was considered as a reference to measure the performance of all. The last column in Table 7-1 represents the relative CPU time needed to converge to global optimum solutions. SEUMRE and SA required 1/3 and 1.3 times of the CPU time, comparing with GA. Figure 7-5 illustrated the needed computation time and number of iterations by these three algorithms to converge to the global optimum. SEUMRE used less computation time and required least number of iterations (115 iterations) to converge to the design optimum. SA and GA required 2500 and 1100 iterations, respectively. SA is the worst in terms of both the number of iterations and CPU time although it does lead to a slightly better minimum in the global optimization.

![Figure 7-5. Computation Cost Comparison](image-url)
The optimal design parameters from the global optimization are given in Table 7-2. One of the very important considerations of MRB system with many rotating parts is the weight. The optimal design using SEUMRE leads to a minimum weight of 13.7 kg, compared to 14.67 and 14.49 Kg from SA and GA, respectively. In this application, the SA algorithm with a deterministic local optimization core search scheme led to more accurate optimum. While the SEUMRE algorithm uses the Kriging search scheme and GA uses the stochastic search method, excellent for high order, large dimensional problems, came slightly short on the convergence accuracy. The resulting small accuracy difference may be quite different for another problem.

Table 7-2. Optimized MRP Prototype Specifications

<table>
<thead>
<tr>
<th>Main Specifications of MRP</th>
<th>Optimization Methods Used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SA</td>
</tr>
<tr>
<td>Diameter (mm)</td>
<td>240.00</td>
</tr>
<tr>
<td>Coil Wire Size</td>
<td>AWD 21 (dia. 0.77mm)</td>
</tr>
<tr>
<td>MR Fluid Used</td>
<td>MRF-132DG</td>
</tr>
<tr>
<td>Maximum Current Applied (A)</td>
<td>1.80</td>
</tr>
<tr>
<td>Maximum Current Density Applied (A/m²)</td>
<td>2.54E+6</td>
</tr>
<tr>
<td>Magnetic Materials Used</td>
<td>Steel 1018</td>
</tr>
<tr>
<td>Weight (kg)</td>
<td>14.8708</td>
</tr>
<tr>
<td>Amount of MR Fluid Used (m³)</td>
<td>6.725E-05</td>
</tr>
<tr>
<td>Maximum Braking Torque(N.m)</td>
<td>33.9896</td>
</tr>
</tbody>
</table>

The other important consideration is the maximum torque generated during braking. Figure 7-6 shows the relation between the current applied to the coil in Amps and the generated braking torque in Nm obtained using the three global optimization algorithms. The optimization result from SA gave better results than SEUMRE and GA. The amount
of the generated braking torque obtained by SA is 33.47 Nm compared to 30.04 and 27.92 Nm from SEUMRE and GA, respectively (see Table 7-2).

The overall performance of an optimization algorithm for computation intensive design optimization problems, such as the MRB design, is judged using the following three criteria.

Figure 7-6. MR Brake System Optimization Results

- **Computational efficiency**

  The number of iterations and computation time needed by space exploration and region elimination global optimization algorithms is always less than that for GA and SA. It is well reflected by the results from the design optimizations using SEUMRE, GA and SA.

- **Performance or convergence accuracy**

  Space exploration and region elimination optimization algorithms can locate the global optimum quickly and reach comparable accuracy for intensive computation optimization applications. In this work, SA converged to the global optimum with
higher accuracy due to its capable, deterministic local optimization search scheme. However, much greater computation efforts (high number of iterations and CPU time) are needed to reach this accuracy.

7.5 Conclusions

In this chapter, the proposed space exploration and region elimination based SEUMRE, and the conventional GA and SA global optimization algorithms were used for the design optimization of an automotive MRB system. The optimal design parameters of the MRB system were identified with maximum braking torque generated at the rotating disks and minimum brake system weight. The results obtained from the three optimization algorithms were compared against each other in terms of computational efficiency, robustness, and accuracy. The advantages of SEUMRE as a new promising global optimization algorithm for complex engineering applications requiring intensive numerical computation are demonstrated.
Chapter 8. Application of the New SEUMRE Global Optimization Tool in High Efficiency EV/PHEV/EREV Electric Mode Operations

8.1 Introduction

Electric vehicles (EV) technologies are environmentally very efficient but cannot succeed on the market because of their limited ability to satisfy customer’s requirements. The overall efficiency and the travelling range which should be improved are some of today’s customers needs that have to be satisfied. Improving the EV energy efficiency between the two electric motors in the model used is the main objective of this work. This cannot be achieved unless the control system, which takes care of the torque distribution between these two electric motors, is optimized. The optimal control system can be obtained by examining the energy efficiency of the vehicle under different loads (speed and torque). In this work, global optimization techniques are used to optimize the EV model presented. Well known and recently introduced non-derivative global optimization techniques are employed to carry out the optimization task. GA, PSO and SEUMRE optimization techniques are particularly used. The proposed model of the EV
is introduced and tested. The obtained results are relatively compared to reveal the advantages and the drawbacks of every one of them. The obtained results are found to be promising. The optimization results are presented and discussed.

Electric Vehicles (EV), Hybrid Electric Vehicles (HEV), Plug-in Hybrid Electric Vehicles (PHEV) and Extend Range Electric Vehicles (EREV) draw mechanical power or regenerate electric power using multiple electric motors and generators (M/Gs). Conventionally, heuristics and experience based control rules are used to guide the determination of powertrain component operation parameters to obtain good efficiency. To achieve optimal vehicles electrical/mechanical energy conversion efficiency and to prolong the pure electric range of these vehicles, the energy conversion efficiency is to be maximized against powertrain component operation parameters using high fidelity model and simulation. However, the energy conversion efficiency model using vehicle powertrain component model and simulation is complex, multimodal, and computationally intensive. An efficient global optimization tool is needed to produce the optimal efficiency look-up surface for real-time control system implementation, or to search for the optimal operation parameters in real time. In this work, the electrical/mechanical energy conversion efficiency of EV and PHEV/EREV in EV mode is modeled using MATLAB Simulink based powertrain component models. In particular, a new 2 mode-plus EREV design is used as a design example. The optimal vehicle electrical/mechanical energy conversion efficiency under various powertrain component operation parameters are obtained using three alternative global optimization tools, GA, PSO and SEUMRE. The conventional GA and PSO tools, with less efficient search efficiency and requiring long search time, are used for benchmark comparisons. The new
SEUMRE global optimization tool is used obtain equally accurate results much efficiently. A rough look-up surface is created to demonstrate the difference in computational efficiency. The application of the SEUMRE global optimization tool allows refined and more accurate vehicle electrical/mechanical energy conversion efficiency map being created for the optimal operation of the EV/PHEV/EREV. Optimal vehicle control schemes can then be generated to determine the speed and torque of the M/Gs of the vehicle without violating their physical constraints and achieving the overall maximum efficiency of the hybrid powertrain system. Results of the design optimization are presented and compared. New design guidelines are provided.

In this work, an investigation was made to apply optimization based approaches to the determination of the optimal control parameters, in order to maximize the energy efficiency. Most existing global optimization methods such as GA and PSO cannot fully satisfy the design need because of the problem complexity. SEUMRE is used and the obtained results are compared with other optimization methods. Since the studied vehicle has a number of states where each state uses a different combination of power plant, the problem has to be defined separately for each state. In this work, the optimization problem primarily solves the electric mode state of the vehicle where two electric motors provide vehicle propulsion. Different optimization algorithms will be employed to solve the problem and the result will be compared.

Various computer simulation software tools, such as ADVISOR [155], PSAT, MATLAB Simulink and Dymola are available for the simulation and analysis of HV operation. The vehicle is modeled in Modelica-based Dymola, which is a forward-looking, transient-capable software. This software has built-in optimization tools,
although the algorithms are not capable of being used for the complex optimization
problem of this study. The author’s research group has a large collection of optimization
algorithms based on the MATLAB programming language [156] that are sufficiently
robust for the task.

8.2 The Vehicle Model

8.2.1 Vehicle Description

The vehicle platform under investigation in this chapter is a 2-mode plus AWD
vehicle. It is technically defined as an extended range electric vehicle (E-REV) which
falls in the bigger plug-in hybrid vehicle category. The original vehicle chassis is based
on a 2009 Saturn VUE which is a compact size SUV. The stock vehicle powertrain on the
front wheels including the ICE and the transmission was replaced by a 2-mode hybrid
system and an electric drive was installed on the rear axle. The major powertrain
components are listed in Table 8-1. A GM Ecotec engine runs on either gasoline or E85.
A GM 2-mode hybrid transmission is a strong hybrid transmission [157, 158] with two
planetary gear sets and two 55 kW permanent magnet electric motors. The engine and the
2-Mode system are placed in the front engine compartment and are connected to the front
wheels. The UQM Powerphase 125 is a permanent magnet synchronized motor with a
peak power output of 125 kW. It is connected to the rear wheels through a rear
differential. The 21 kW-hr A123 battery pack provides power to both the 2-Mode system
and the UQM motor. A schematic layout of the studied vehicle is shown in Table 8-1.
8.2.2 Configuration

In a typical daily operating cycle of less than 60 miles, the majority of the trip will be completed in the EV mode which utilizes only the electric power from the batteries. In the EV mode, two electric motors that drive the vehicle are the rear traction motor and one of the two electric motors from the 2-mode transmission. To most efficiently use the two motors to power the vehicle, optimizations are performed to find the most efficient power split ratio between the two motors at different vehicle speed and load conditions.

**Table 8-1. Vehicle Specifications**

<table>
<thead>
<tr>
<th>Engine</th>
<th>GM 2.4 L Ecotech engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rear wheel Traction Motor</td>
<td>UQM Powerphase 125 kW electric drive</td>
</tr>
<tr>
<td>Battery</td>
<td>21 kW-hr- A123 Power Systems</td>
</tr>
<tr>
<td>Transmission</td>
<td>GM 2-Mode FWD</td>
</tr>
</tbody>
</table>
The driver commanded torque is interpreted from the accelerator pedal position and vehicle speed. In the EV, the driver’s torque request is commanded to the rear traction motor and one motor in the 2-mode transmission. The control parameters are listed in Table 8-2.

Table 8-2. Control Parameter List

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_{RTM}</td>
<td>Nm</td>
<td>Traction torque from the rear traction motor</td>
</tr>
<tr>
<td>T_{2-mode}</td>
<td>Nm</td>
<td>Traction torque from the 2-mode motor</td>
</tr>
<tr>
<td>T_{request}</td>
<td>Nm</td>
<td>Torque request from the vehicle driver</td>
</tr>
</tbody>
</table>

8.3 The Optimization Problem Formulation

The goal of the optimization is to find optimal control commands for motor generator B (MGB) that can maximize the energy efficiency (electrical/mechanical energy conversion efficiency) of the vehicle at different loading conditions (speed and torque).

The objective function can be expressed as:

$$\eta = f(v, T, P_{MGB})$$

where $\eta$ is the electrical/mechanical energy conversion efficiency in the electric drives. When the motors are propelling the vehicle, $\eta$ is a ratio of the mechanical power output from the motor divided by the electrical power withdrawn from the batteries. When the motor is generating electricity during vehicle braking, $\eta$ is a ratio of the generated electrical power divided by the mechanical power. The power flow direction when batteries are discharged and the motors are propelling is defined as positive. Likewise, the power flow when batteries are being charged and the motors are regenerating is defined as negative. Therefore, the energy efficiency during either propelling or braking.
is in the range of 0-100%. \( \nu \) is the vehicle speed in km/hr. \( T \) is the overall vehicle torque demand on wheels in Nm and \( P_{MGB} \) is the commanded power output at MGB in KW.

To derive optimal control rules for controlling the MGB and the rear wheel traction motor (RTM) under varying vehicle load conditions, a grid is generated to represent different driving conditions as combinations of \( \nu \) and \( T \). The problem then becomes solving \( P_{MGB} \) at each grid point \( \eta_{\nu,T} \). A negative sign is applied before \( \eta \) to convert the maximization problem into a minimization problem. Then the modified objective function is

\[
-\eta_{\nu,T} = f(P_{MGB})
\]  

(8-2)

Subject to the following constraints:

\[
0.8 \times P_{Battery\_charging} \leq \nu \times T \leq -0.8 \times P_{Battery\_Discharging} ;
\]  

(8-3)

\[
-P_{regen}(n_{MGB}) \leq P_{MGB} \leq P_{prop}(n_{MGB}) ;
\]  

(8-4)

\[
-P_{regen}(n_{RTM}) \leq P_{RTM} \leq P_{prop}(n_{RTM}) ;
\]  

(8-5)

\[
0 \leq \nu \leq 60
\]  

(8-6)

The vehicle speed is between 0-16 km/hr range because of the 2-Mode transmission limitation at EV mode as previously explained. The vehicle power demand defined as \( \nu \times T \) has to be within 80% of the battery power capacity where 80% is an empirical safety factor for battery power management. Power output at MGB and RTM has to be within the power limits based on the characteristics of each motor.
8.4 Optimization Results

Three optimization methods were selected to perform the optimization process and search for global solutions of the objective function (maximizing the electrical/mechanical energy conversion efficiency of HEV). These optimization algorithms are different in the way they search for global solution.

The obtained results are presented in Table 8-3 and Table 8-4. These are look-up tables which were generated using the three optimization algorithms. These look-up tables can be used to get the optimum electrical/mechanical energy conversion efficiency at any vehicle speed and power. The results presented in Table 8-3 are the ones obtained from SEUMRE. These results are obtained at every grid point (combination of speed and torque). A detailed discussion of the results is also included in this section.

Table 8-3. Optimum Electrical/Mechanical Energy Efficiency Results Obtained from (SEUMRE)

<table>
<thead>
<tr>
<th>Speed(km/hr)</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power(KW)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-60</td>
<td>79.8546</td>
<td>81.8703</td>
<td>83.9417</td>
<td>86.4968</td>
<td>89.5518</td>
<td>90.7136</td>
<td>92.1070</td>
</tr>
<tr>
<td>-50</td>
<td>79.8435</td>
<td>82.0245</td>
<td>84.3929</td>
<td>88.2284</td>
<td>90.5610</td>
<td>90.8687</td>
<td>91.6932</td>
</tr>
<tr>
<td>-40</td>
<td>79.8496</td>
<td>82.2553</td>
<td>86.0245</td>
<td>89.0512</td>
<td>90.3157</td>
<td>90.7044</td>
<td>92.3663</td>
</tr>
<tr>
<td>-30</td>
<td>79.8497</td>
<td>82.9387</td>
<td>87.0611</td>
<td>89.0955</td>
<td>91.0749</td>
<td>90.2089</td>
<td>92.9315</td>
</tr>
<tr>
<td>-20</td>
<td>79.7511</td>
<td>83.8005</td>
<td>88.0740</td>
<td>89.7249</td>
<td>90.1364</td>
<td>90.3694</td>
<td>91.8777</td>
</tr>
<tr>
<td>-10</td>
<td>79.8467</td>
<td>85.1718</td>
<td>87.6299</td>
<td>87.6757</td>
<td>87.2052</td>
<td>87.4184</td>
<td>87.4881</td>
</tr>
<tr>
<td>10</td>
<td>79.8334</td>
<td>85.2197</td>
<td>87.6515</td>
<td>87.6767</td>
<td>87.3099</td>
<td>87.4186</td>
<td>87.5873</td>
</tr>
<tr>
<td>20</td>
<td>79.8484</td>
<td>83.9529</td>
<td>88.2357</td>
<td>89.7577</td>
<td>90.0853</td>
<td>90.3669</td>
<td>91.8932</td>
</tr>
<tr>
<td>30</td>
<td>79.8499</td>
<td>83.3645</td>
<td>87.2085</td>
<td>89.8565</td>
<td>91.7278</td>
<td>91.5711</td>
<td>92.9862</td>
</tr>
<tr>
<td>40</td>
<td>79.8567</td>
<td>82.4203</td>
<td>86.6157</td>
<td>89.0197</td>
<td>90.9605</td>
<td>91.0795</td>
<td>92.3730</td>
</tr>
<tr>
<td>50</td>
<td>79.8554</td>
<td>82.1528</td>
<td>85.9635</td>
<td>88.7494</td>
<td>90.9398</td>
<td>91.1775</td>
<td>91.6691</td>
</tr>
<tr>
<td>60</td>
<td>79.8544</td>
<td>81.9758</td>
<td>84.6526</td>
<td>88.2703</td>
<td>90.4644</td>
<td>90.9807</td>
<td>92.0998</td>
</tr>
<tr>
<td>70</td>
<td>79.8568</td>
<td>81.8495</td>
<td>84.1189</td>
<td>87.2325</td>
<td>90.1986</td>
<td>90.8292</td>
<td>92.2250</td>
</tr>
<tr>
<td>80</td>
<td>79.8550</td>
<td>81.7538</td>
<td>83.8346</td>
<td>86.0727</td>
<td>89.2418</td>
<td>90.3291</td>
<td>91.6669</td>
</tr>
</tbody>
</table>
The power split ratio between electrical and mechanical motors are shown in Table 8-4.

A summary of the results obtained from the three global optimization algorithms used in this work are presented in Table 8-5. Average optimum fuel efficiency obtained from each algorithm is also presented in Table 8-5. CPU time, in hours, needed to obtain global solutions, maximum electrical/mechanical energy conversion efficiency, as well as relative CPU time where GA is being considered as a reference since it is one of the well known global optimization algorithm. SEUMRE requires less than half the computation time than what GA requires and PSO requires more than 3 times than what GA requires. So this clearly tells how well SEUMRE did.

Table 8-4. Optimum Power Split Ratio Results Obtained from SEUMRE

<table>
<thead>
<tr>
<th>Speed(km/hr)</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>-60</td>
<td>-0.0079</td>
<td>-0.9997</td>
<td>-0.9999</td>
<td>-0.9997</td>
<td>-1.0000</td>
<td>-1.0000</td>
<td>-0.9999</td>
</tr>
<tr>
<td>-50</td>
<td>-0.0215</td>
<td>-1.0000</td>
<td>-0.9999</td>
<td>-0.9997</td>
<td>-1.0000</td>
<td>-0.9999</td>
<td>-0.8598</td>
</tr>
<tr>
<td>-40</td>
<td>-0.0003</td>
<td>-0.9999</td>
<td>-0.9999</td>
<td>-0.9814</td>
<td>-0.9998</td>
<td>-0.9999</td>
<td>-0.0065</td>
</tr>
<tr>
<td>-30</td>
<td>-0.0186</td>
<td>-0.9997</td>
<td>-1.0000</td>
<td>-0.9999</td>
<td>-1.0000</td>
<td>-0.0055</td>
<td>-0.0041</td>
</tr>
<tr>
<td>-20</td>
<td>-0.0166</td>
<td>-0.8066</td>
<td>-0.9999</td>
<td>-0.7703</td>
<td>-0.6973</td>
<td>-0.0033</td>
<td>-0.0055</td>
</tr>
<tr>
<td>-10</td>
<td>-0.0025</td>
<td>-0.7706</td>
<td>-0.5137</td>
<td>-0.3868</td>
<td>-0.3489</td>
<td>-0.0018</td>
<td>-0.0008</td>
</tr>
<tr>
<td>10</td>
<td>0.1998</td>
<td>0.3880</td>
<td>0.0008</td>
<td>0.0003</td>
<td>0.2050</td>
<td>-0.0001</td>
<td>-0.0008</td>
</tr>
<tr>
<td>20</td>
<td>0.0027</td>
<td>0.6000</td>
<td>0.5280</td>
<td>0.4111</td>
<td>0.4136</td>
<td>0.0014</td>
<td>-0.0006</td>
</tr>
<tr>
<td>30</td>
<td>0.0039</td>
<td>0.6053</td>
<td>0.6021</td>
<td>0.6122</td>
<td>0.6198</td>
<td>0.6393</td>
<td>0.0000</td>
</tr>
<tr>
<td>40</td>
<td>0.0062</td>
<td>0.6019</td>
<td>0.6638</td>
<td>0.5206</td>
<td>0.8261</td>
<td>0.5976</td>
<td>0.0028</td>
</tr>
<tr>
<td>50</td>
<td>0.0003</td>
<td>0.6022</td>
<td>0.8064</td>
<td>0.6325</td>
<td>0.6547</td>
<td>0.6875</td>
<td>0.5225</td>
</tr>
<tr>
<td>60</td>
<td>0.0029</td>
<td>0.6039</td>
<td>0.9462</td>
<td>0.8322</td>
<td>0.7381</td>
<td>0.8036</td>
<td>0.7389</td>
</tr>
<tr>
<td>70</td>
<td>0.0002</td>
<td>0.6023</td>
<td>0.6434</td>
<td>0.9606</td>
<td>0.9199</td>
<td>0.8043</td>
<td>0.8046</td>
</tr>
<tr>
<td>80</td>
<td>0.0039</td>
<td>0.6022</td>
<td>0.8030</td>
<td>0.8486</td>
<td>0.997</td>
<td>0.8944</td>
<td>0.8874</td>
</tr>
</tbody>
</table>
Table 8-5. Summary of Optimization Algorithms Results

<table>
<thead>
<tr>
<th>Global Optimization Method</th>
<th>Average Optimum Fuel Efficiency Obtained</th>
<th>CPU time (hrs) required to converge to the optimum</th>
<th>Relative CPU time (hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEUMRE</td>
<td>86.9963</td>
<td>0.7673</td>
<td>0.47</td>
</tr>
<tr>
<td>PSO</td>
<td>86.9494</td>
<td>5.5512</td>
<td>3.38</td>
</tr>
<tr>
<td>GA</td>
<td>86.9945</td>
<td>1.6419</td>
<td>1.00</td>
</tr>
</tbody>
</table>

The optimum energy obtained using SEUMRE is plotted as a surface plot (Figure 8-2) to show the efficiency obtained when SEUMRE is used and how SEUMRE contributes largely towards improving the fuel efficiency of HV. This surface plot could be to directly obtain the vehicle efficiency at any speed and power.

![Figure 8-2. The Optimum Energy Efficiency Obtained Using SEUMRE](image)

8.5 Discussion of The Results

The optimization results obtained from the three optimization methods are introduced and discussed in this section. Three important features - performance or convergence
accuracy, robustness and computational efficiency - of the optimization methods used in this work are discussed and compared.

The results from the tests carried out in this work have demonstrated that space exploration and region elimination based meta-modeling optimization techniques has many advantages. The three used optimization algorithms were capable of solving the challenging hybrid vehicle problem. SEUMRE located the optimum with comparable accuracy and obtained the results with much reduced computation time.

The computation efficiency advantage of SEUMRE will become more significant with more complex engineering applications. The comparison studies showed this trend. From the results presented in Table 8-3, it can be observed that in most tested scenarios the accuracy of GA and PSO is better than SEUMRE. However, these two global optimization methods need too much time to produce accurate solutions.

All global optimization methods used in this work were able to converge to global optimum solutions. SEUMRE has yield comparably good results with less computational cost compared to GA and PSO. Also, SEUMRE converged to an acceptable efficiency which is the maximum energy efficiency for each combination of vehicle speed and power. In all these scenarios for speed ranging from 0 m/s to 15m/s and power ranging from -60 kW till 80 kW, SEUMRE was able to converge to global optimum solutions with less computation cost. The CPU time needed by the SEUMRE method is only 50.00% of GA and 12.9% of PSO in average. Figure 8-3 shows the required computation time for all algorithms used.
The maximum (best) electrical/mechanical energy conversion efficiency obtained is about 93%, obtained at vehicle speed equal to 14 m/s and vehicle power equal to 30 kW. These results were obtained by means of GA and PSO, while the results at the same grid point (14 m/s and 30 kW) obtained using SEUMRE is about 92.4%, slightly lower, but with much less computation. This is why the space exploration method is a good alternative for high computation applications.

As shown in Figure 8-3, SEUMRE has contributed largely in improving the electrical/mechanical energy conversion efficiency fuel efficiency of HEV. SEUMRE performs well compared to the other methods when it comes to comparable accuracy and computation cost as well as robustness. Figure 8-4 shows a comparison between these optimization methods at different vehicle power values (-60, -30, 40 and 80 kW). It also shows the efficiency obtained using three optimization methods at different speed and

![Figure 8-3. Computational Time Required by Optimization Algorithms to Converge to Global Solutions.](image)
power values. These figures clearly show that the energy efficiency obtained using these optimization algorithms is high and specifically from SEUMRE which performs well compared to the other two well established global optimization algorithms. What makes SEUMRE even better is the amount of time needed to approach the global solutions. SEUMRE approaches global solutions with much less computation time compared to GA and PSO.

Figure 8-4. Optimization Algorithms Performance Comparison at Different Vehicle Power Values (a) -60Kw; (b) -30 Kw; (c) 40 Kw; and (d) 80Kw

8.6 Conclusions

The work carried out in this chapter was to investigate and maximize the energy efficiency of plug-in hybrid electric powertrain. The energy conversion efficiency is maximized against powertrain component operation parameters using high fidelity model
and simulation. An efficient global optimization tool SEUMRE was used to produce the optimal efficiency look-up surface for real-time control system implementation, and to search for the optimal operation parameters in real time. The electrical/mechanical energy conversion efficiency of EV and PHEV/EREV in EV mode was modeled. The optimal vehicle electrical/mechanical energy conversion efficiency under various powertrain component operation parameters were obtained using three alternative global optimization tools, GA, PSO and SEUMRE.

A rough look-up surface was created to demonstrate the difference in computational efficiency. Application of the SEUMRE global optimization tool allowed refined and more accurate vehicle electrical/mechanical energy conversion efficiency map being created for the optimal operation of the EV/PHEV/EREV. Optimal vehicle control schemes were generated which can be used to determine the speed and torque of the M/Gs of the vehicle without violating their physical constraints and achieving the overall maximum efficiency of the hybrid powertrain system. Results of the design optimization were presented and compared. New design guidelines were provided.
Chapter 9. Conclusions and Recommendations for Future Research

9.1 Conclusions

New global optimization algorithms were developed and tested with benchmark test problems and real life applications. An adaptive multi-objective optimization algorithm to identify the efficient Pareto frontier for back-box functions were developed and introduced. The proposed algorithms have shown to provide global solutions with comparable accuracy and with less computation cost. The research objectives set in Section 1.4 have been completed with the results summarized in the following subsections.

9.1.1 New Region Elimination Optimization Algorithm

A new sequential, global optimization algorithm based on design experiments, region elimination and response surface model, namely Approximated Unimodal Region Elimination (AUMRE) method, is introduced. The method divides the design space into many sub-areas to make that the optimization in the sub-area becomes a unimodal function so that a second order polynomial response surface model can be used to replace
the black-box objective function of numerical analysis and simulation, and conventional optimization algorithms can be used to locate its minimum. The search starts from the most promising sub-area, and identify the global optimum of the design problem by comparing the optima of all sub-areas processed. In this work, the new algorithm and its implementation are first explained. Tests on the robustness and efficiency of the new algorithm are carried out using a number of benchmark problems, in comparison of several well-known global optimization methods.

To illustrate the advantages of this newly introduced algorithm, a number of benchmark global optimization problems were solved using the newly proposed global optimization algorithm and other reviewed global optimization algorithms. AUMRE obtained better results in most cases, and consistently outperformed the stochastic global optimization techniques in computation time, making it an ideal tool for identifying the optimal design using complex, black-box computer analysis and simulation tools in multidisciplinary design optimization. The algorithm performs well largely for low dimensional, unconstrained problems, leading to the introduction of more complex search algorithms in the following sections.

9.1.2 New Space Exploration Optimization Algorithm

A new metamodelling, space exploration and region reduction search algorithm is introduced. This algorithm, namely Space Exploration and Unimodal Region Elimination (SEUMRE), divides the design space into key unimodal regions using design experiment data; identifies the regions that most likely contain the global minimum; fits Kriging models with additional design experiments using Latin Hypercube designs over these regions; identifies their local minima, and then the global optimum. By identifying
promising unimodal regions of the objective and reducing searching space, the method can find the global optimum effectively and efficiently, particularly suited for optimization problems that require extensive computation through engineering analyses and simulations. Comparisons with existing space exploration and region elimination/reduction methods using benchmark test problems have been carried out to demonstrate the advantages of the new method.

9.1.3 New Mixed Surrogate Models Optimization Algorithm

A new global optimization algorithm, namely Mixed Surrogate Models and Design Space Elimination Search (MSSE), is introduced. The approach identifies the most appreciate surrogate for the metamodel through the search process to obtain consistent and superior search efficiency. It also divides the field of interest into several unimodal regions; identify and rank the regions that likely contain the global minimum; fits a Radial Basis function and Quadratic Response Surface model over each promising region with additional design experiments data points using Latin Hypercube designs; identifies its minimum and removes the processed region; and moves to the next most promising region until all regions are processed and the global optimum is identified. The new algorithm was tested using several benchmark problems for global optimization and compared with several widely used region elimination and space exploration global optimization algorithms, showing reduced computation efforts, robust performance and comparable search accuracy, making the new method an excellent tool for computation intensive, computer analysis/simulation based global design optimization problems.
9.1.4 New Multi-objective Optimization Algorithm

Design of complex multidisciplinary system requires computation intensive analysis and simulation tools, such as finite element analysis (FEA) and computational fluid dynamics (CFD), and involves multiple, competing design objectives. In this work, an adaptive metamodeling technique for multi-objective optimization (MOO) is introduced. The new method extends the adaptive metamodeling mechanism for single objective global optimization to the multi-objective design optimization problems. The approach can identify the Pareto front in multi-objective optimization more efficiently and with high accuracy. The newly proposed algorithm was tested using various benchmark test problems as well as a practical design example with promising results. Direct tests of MOO are to be carried out in future research.

9.1.5 Real Life Engineering Design Optimization Problems

The newly proposed algorithms were used to optimize two engineering applications. The first is the MRP system in which optimum design parameters are required to minimize the weight of the rotating disk and maximize the generated torque. The other application is the optimization of electric vehicles (EV) control strategy. In this work, the electrical/mechanical energy conversion efficiency of EV and PHEV/EREV in EV mode is modeled using MATLAB Simulink based powertrain component models. In particular, a new 2 mode-plus EREV design is used as a design example. The optimal vehicle electrical/mechanical energy conversion efficiency under various powertrain component operation parameters are obtained using three alternative global optimization tools, Genetic algorithm (GA), Particle Swarm Optimization (PSO) and Space Elimination and Region Reduction (SEUMRE). The conventional GA and PSO tools,
with less efficient search efficiency and requiring long search time, are used for benchmark comparisons. The new SEUMRE global optimization tool is used obtain equally accurate results much efficiently. A rough look-up surface is created to demonstrate the difference in computational efficiency.

9.2 Research Contributions

This work has introduced new robust and efficient global optimization (GO) algorithms and improves the performance accuracy of these algorithms through space exploration, region reduction and metamodeling techniques. GO algorithms with improved performance capabilities and robustness efficiency can perform well when used in optimizing complex engineering applications. Many single objective and multi-objective global optimization algorithms based on space exploration and region elimination for black-box function in engineering design are introduced and tested using benchmark test problems and real life engineering applications. The dissertation contributions can be summarized as follows:

1. Carried out extensive literature review and benchmark testing on state of the art metamodel-based global optimization techniques and multidisciplinary optimization search algorithms.

2. Introduced several new region elimination and metamodeling-based global optimization algorithms with improved search efficiency and robust performance, for black-box, computation intensive, engineering analysis/simulation based global optimization problems.
3. Introduced new multi-objective global optimization algorithms based space exploration and metamodeling techniques for complex multi-objective optimization problems.

4. Tested the new optimization algorithms using various benchmark test examples and typical industrial design problems.

5. Applied the newly developed, metamodel-based global optimization techniques in the design optimization and control strategy development of two-mode hybrid electric vehicles and other complex mechanical systems.

9.3 Future Work

Region elimination, space exploration, and multi-objective optimization algorithms are continuously developing fields of Engineering Optimization. This work contributes to both fields, but more challenges remain to be overcome. Several areas for future research that needs to be carried out to extend the results of this work are listed in the following subsections.

9.3.1 Dividing and Partitioning the Design Space

This work introduced new search optimization algorithms that depend largely on the way in which design space is divided, or partitioned. Efficient techniques in dividing and partitioning the design space are seriously needed. Good techniques of how every algorithm deal with partitioning and dividing the design space could highly impact the speed of the algorithm in converging to global optimum solutions.
9.3.2 Introducing Efficient and Robust Global Optimization Algorithms

More efficient and robust global optimization algorithms that can handle optimization problems with hundreds of design variables and converge to optimum solutions with good accuracy are seriously needed. The proposed algorithms introduced in this work should be improved to handle black-box functions with hundreds of design variables and highly nonlinear objective functions and constraints. Introducing new efficient and robust optimization algorithms for black-box functions will definitely reduce and minimize the need for expensive resources such as very powerful PCs to carry out computational tasks. The new algorithms should converge to global optimum solutions with high accuracy and less computation cost.

9.3.3 Enhancing Computer Experiments Sampling Techniques

Selecting the right sampling technique has an important influence on the efficiency of exploring the design space. It is very crucial to sample and cover the whole design space in an efficient and economic way to explore it and find global solutions. It is clear that in some sampling techniques the number of sample points depends on the number of design variables. As the number of design variable increases the number of sample points increases exponentially. This type of problem should be addressed and new efficient sampling techniques should be developed and introduced. Introducing new sampling techniques will defiantly affect and improve the speed of the algorithm and yield good solutions with acceptable accuracy.

9.3.4 Testing of MOO Design Applications

Use of the newly introduced MOO algorithm to directly search for MOO results is to be carried out.
References


Appendix A. Tested Benchmark Problems

The following are the mathematical formula and the analytical global optimum solution of the benchmark test problems used in this dissertation.

A.1 Unconstrained Test Problems

1) **Banana Function, n=2:**

   \( f(x) = 100(x_2 - x_1^2) + (1 - x_1)^2 \)

   Search Space: \(-2 \leq x_i \leq 2, \ i = 1, 2.\)

   Global minimum: \( x^* = (1, ..., 1), \ f(x^*) = 0 \)

2) **Beak Function, n=2:**

   \[ f(x_1, x_2) = 3(1 - x_1)^2e^{-(x_1^2-(x_2+1)^2)} - 10 \left( \frac{x_1^4}{5} - x_1^3 - x_2^5 \right) e^{-(x_1^2-x_2^2)} \]

   \[ - \frac{1}{3} e^{-(x_1+1)^2-x_2^2} \]

   Search Space: \(-3 \leq x_1 \leq 3, \ -4 \leq x_2 \leq 4.\)

   Global minimum: \( x^* = (0.2283, -1.6255), \ f(x^*) = -6.5511 \)

3) **Branin Function, n=2:**

   \[ f(x) = \left( x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10 \]

   Search Space: \(-5 \leq x_1 \leq 0, \ 10 \leq x_2 \leq 15.\)

   Global minimum: \( x^* = (-\pi, 12.275), \ x^* = (\pi, 12.275), \ f(x^*) = 0.3978 \)
4) Generalize Polynomial Function, $n=2$:

\[
f(x_1, x_2) = (1.5 - x_1 (1 - x_2))^2 + (2.25 - x_1 (1 - x_2^2))^2 + (2.625 - x_1 (1 - x_2^2))^2
\]

Search Space: $-2 \leq x_1, x_2 \leq 2$

Global minimum: $x^* = (2.0000, 0.1700), \ f(x^*) = 0.5233$

5) Goldstein and Price Function (GP), $n=2$:

\[
f(x_1, x_2) = \left(1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right)
\times \left(30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right)
\]

Search Space: $-2 \leq x_1, x_2 \leq 2$

Global minimum: $x^* = (0, -1), \ f(x^*) = 3$

6) Griewank Function, $n=2$:

\[
f(x_1, x_2) = \frac{x_1^2 + x_2^2}{200} - \cos x_1 \cos \left(\frac{x_2}{\sqrt{2}}\right) + 1
\]

Search Space: $-100 \leq x_1, x_2 \leq 100$

Global minimum: $x^* = (0,0), \ f(x^*) = 0$

7) Hartman Function (H6), $n=6$:

\[
f_{HN}(x) = -\sum_{i=1}^{4} C_i \exp \left( -\sum_{j=1}^{n} a_{ij} (x_j - p_{ij})^2 \right)
\]

\[
[a_{ij}]_{j=1,...,6} = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}
\]

\[
c = [1 \ 1.2 \ 3 \ 3.2]
\]
\[
[p_{ij}]_{j=1,\ldots,6} = \begin{bmatrix}
0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\
0.2329 & 0.4139 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\
0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\
0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381
\end{bmatrix}
\]

Search space: \(0 \leq x_i \leq 1, \ i = 1, \ldots, n\)

Global minimum: \(x^* = (0.02169, 0.15001, 0.47687, 0.27533, 0.31165, 0.6573)\);

\(f(x^*) = -3.3223\)

8) Hartmann Function with 16 Design Variables (H16), \(n=16\):

\(f(x) = \sum_{i=1}^{16} \sum_{j=1}^{16} a_{ij} (x_i^2 + x_i + 1)(x_j^2 + x_j + 1), \ i, j = 1, 2, \ldots, 16\)

\([a_{ij}]_{row1-8} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}\)

\([a_{ij}]_{row9-16} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}\)

Search Space: \(-1 \leq x_i \leq 1, \ i = 1, \ldots, 16\)

Global minimum: \(f(x^*) = 25.875\)
9) Levy Function, n=4:

\[ f(x) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1} \left( (y_i - 1)^2 (1 + \sin^2(\pi y_i + 1)) \right) + (y_n - 1)^2 (1 + 10 \sin^2(2\pi y_n)) , \ y_i = 1 + \frac{x_i - 1}{4}, \ i = 1, \ldots, n \]

Search Space: \(-10 \leq x_i \leq 10, \ i = 1, \ldots, n.\)

Global minimum: \(x^* = (1, \ldots, 1), \ f(x^*) = 0\)

10) Shekel Function, n=4:

\[ f(x) = -\sum_{i=1}^{4} \alpha_i \exp \left(-\sum_{j=1}^{6} B_{ij} (x_j - Q_{ij})^2 \right), \]

\[ \alpha = [1 \ 1.2 \ 3 \ 3.2]^T, \]

\[ B = \begin{bmatrix} 10 & 3 & 17 & 3.05 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}, \]

\[ Q = 10^{-4} \times \begin{bmatrix} 1312 & 1696 & 15569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 03736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix} \]

Search Space: \(0 \leq x_i \leq 1, \ i = 1, \ldots, 6.\)

Global minimum: \(x^* = (4, 4, 4, 4), \ f(x^*) = -10.1532\)

11) Shubert Function, n=2:

\[ f(x_1, x_2) = \left( \sum_{i=1}^{5} i \cos ((i + 1)x_1 + i) \right) \left( \sum_{i=1}^{5} i \cos ((i + 1)x_2 + i) \right), \]

Search Space: \(-10 \leq x_i \leq 10, \ i = 1, 2\)
Global minimum: \( x^* = (-1.4252, 5.4829) \), \( f(x^*) = -186.7309 \)

12) **Sphere Function, n=10:**

\[
f(x) = \sum_{i=1}^{n} x_i^2,
\]

Search Space: \(-2.56 \leq x_i \leq 5.12, \quad i = 1, \ldots, n\)

Global minimum: \( x^* = (0, \ldots, 0) \), \( f(x^*) = 0 \)

13) **Six-hump Camel-back Function (SC), n=2:**

\[
f(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4
\]

Search Space: \(-2 \leq x_1, x_2 \leq 2, \)

Global minimum: \( x^* = (0.089842, -0.712656), x^* = (-0.089842, 0.712656) \)

\( f(x^*) = -1.031628 \)

14) **Trid Function, n=10:**

\[
f(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_ix_{i-1},
\]

Search Space: \(-n^2 \leq x_i \leq n^2, \quad i = 1, \ldots, n\)

Global minimum: \( x_i^* = i(7 - i), \quad i = 1, \ldots, n, \quad f(x^*) = -50 \)

A.2 **Constrained Test Problems**

15) **Problem C1:**

\[
\min f(x_1, x_2) = (x_1 - 10)^3 + (x_2 - 20)^3
\]

subject to:

\[
h_1(x) = (x_1 - 5)^2 + (x_2 - 5)^2 + 100 \leq 0,
\]
\[ h_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2 - 82.81 \leq 0. \]

Search Space: \( 100 \leq x_1 \leq 13; \ 100 \leq x_2 \leq 0. \)

Global minimum: \( x^* = (14.095, 0.84296), \ f(x^*) = -6961.81388 \)

16) Problem C2:

\[
\min f(x_1, x_2, x_3, x_4, x_5) = e^{x_1^2x_2x_3x_4x_5}
\]

Subject to:

\[
h_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0,
\]

\[
h_2(x) = x_2x_3 - 5x_4x_5 = 0,
\]

\[
h_3(x) = x_1^3 + x_2^3 + 1 = 0.
\]

Search Space: \( 2.3 \leq x_1, x_2 \leq -2.3; \ 3.2 \leq x_3, x_4, x_5 \leq -3.2 \)

Global minimum: \( x^* = (-1.71714, 1.59571, 1.82724, -0.76364, -0.76364 ); \)

\[ f(x^*) = 0.0539498 \]

17) Problem C3:

\[
\min f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141
\]

Subject to:

\[
h_1(x) = u(x) - 92 \leq 0,
\]

\[
h_2(x) = -u(x) \leq 0,
\]

\[
h_3(x) = v(x) - 110 \leq 0,
\]

\[
h_4(x) = -v(x) + 90 \leq 0,
\]

\[
h_5(x) = -w(x) + 20 \leq 0,
\]

where

\[ u(x) = 85.334407 + 0.0056858x_2x_5 + 0.000626x_1x_4 - 0.0022053x_3x_5, \]

\[ v(x) = 80.51249 + 0.007131x_2x_5 + 0.0029955x_1x_2 + 0.002183x_3^2, \]
Global minimum: $x^* = (78, 33, 29.9952, 45, 36.77581); f(x^*) = -30.665.539$

18) Tension compression string problem

The problem of minimizing the weight of a tension-compression string (Coello and Montez 2002) can be expressed as the following optimization problem with three design variables $x = (x_1, x_2, x_3)$:

$$\min f(x_1, x_2, x_3) = x_1^2 x_2(x_3 + 2),$$

subject to:

$$h_1(x) = 1 - \frac{x_2^3 x_3}{71,785 x_1^4} \leq 0,$$

$$h_2(x) = \frac{4x_2^2 - x_1 x_2}{12,566 x_1^2 (x_2 - x_1)} + \frac{1}{5,108 x_1^2} - 1 \leq 0,$$

$$h_3(x) = 1 - \frac{140.45 x_1}{x_3 x_2^2} \leq 0,$$

$$h_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0.$$ 

Global minimum: $x^* = (0.0517425, 0.3580047, 11.2139073); f(x^*) = 0.012665$
Appendix B. Sample of SEUMRE Optimization Results on the Tested Benchmark Problems

Table B-1. Optimization Results on Six-hump Camel-back (SC) Function

<table>
<thead>
<tr>
<th>Run No.</th>
<th>Global Optimum $x^*$</th>
<th>Optimum Value</th>
<th># F. Evl.</th>
<th>CPU (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0.0900, -0.7128)</td>
<td>-1.0316</td>
<td>47</td>
<td>3.4</td>
</tr>
<tr>
<td>2</td>
<td>(0.0896, -0.7123)</td>
<td>-1.0316</td>
<td>33</td>
<td>2.0</td>
</tr>
<tr>
<td>3</td>
<td>(0.0811, -0.7159)</td>
<td>-1.0312</td>
<td>30</td>
<td>1.7</td>
</tr>
<tr>
<td>4</td>
<td>(-0.0923, 0.7104)</td>
<td>-1.0316</td>
<td>28</td>
<td>1.5</td>
</tr>
<tr>
<td>5</td>
<td>(-0.0925, 0.7110)</td>
<td>-1.0316</td>
<td>35</td>
<td>2.3</td>
</tr>
<tr>
<td>6</td>
<td>(0.0912, -0.7131)</td>
<td>-1.0316</td>
<td>37</td>
<td>2.4</td>
</tr>
<tr>
<td>7</td>
<td>(0.0890, -0.7116)</td>
<td>-1.0316</td>
<td>35</td>
<td>2.2</td>
</tr>
<tr>
<td>8</td>
<td>(-0.0893, 0.7145)</td>
<td>-1.0316</td>
<td>37</td>
<td>2.4</td>
</tr>
<tr>
<td>9</td>
<td>(0.0901, -0.7127)</td>
<td>-1.0316</td>
<td>40</td>
<td>2.7</td>
</tr>
<tr>
<td>10</td>
<td>(-0.0905, 0.7121)</td>
<td>-1.0316</td>
<td>40</td>
<td>2.7</td>
</tr>
</tbody>
</table>

Table B-2. Optimization Results on Levy Function

<table>
<thead>
<tr>
<th>Run No.</th>
<th>Global Optimum $x^*$</th>
<th>Optimum Value</th>
<th># F. Evl.</th>
<th>CPU (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0.9980, 0.9851, 0.9897, 0.9383)</td>
<td>4.0780e-004</td>
<td>127</td>
<td>19.1</td>
</tr>
<tr>
<td>2</td>
<td>(0.9867, 1.0181, 1.0177, 0.9383)</td>
<td>7.6680e-004</td>
<td>137</td>
<td>22.3</td>
</tr>
<tr>
<td>3</td>
<td>(0.9913, 0.9950, 0.9837, 1.0988)</td>
<td>8.5546e-004</td>
<td>113</td>
<td>15.8</td>
</tr>
<tr>
<td>4</td>
<td>(0.9969, 0.9804, 0.9110, 0.9399)</td>
<td>0.0041</td>
<td>127</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>(0.9945, 0.9999, 1.0037, 1.0795)</td>
<td>4.4158e-004</td>
<td>120</td>
<td>17.4</td>
</tr>
<tr>
<td>6</td>
<td>(1.0569, 1.0177, 0.9915, 1.0325)</td>
<td>0.0040</td>
<td>162</td>
<td>33.2</td>
</tr>
<tr>
<td>7</td>
<td>(0.9939, 0.9907, 0.9985, 0.9834)</td>
<td>1.0230e-004</td>
<td>120</td>
<td>17.9</td>
</tr>
<tr>
<td>8</td>
<td>(1.0029, 0.9991, 0.9857, 1.0785)</td>
<td>5.0207e-004</td>
<td>134</td>
<td>22.1</td>
</tr>
<tr>
<td>9</td>
<td>(0.9872, 1.0246, 0.9772, 0.9658)</td>
<td>8.2650e-004</td>
<td>113</td>
<td>15.9</td>
</tr>
<tr>
<td>10</td>
<td>(1.0013, 0.9906, 0.9177, 1.0236)</td>
<td>0.0032</td>
<td>127</td>
<td>19.7</td>
</tr>
</tbody>
</table>
Table B-3. Optimization Results on Hartmann (H6) Function

<table>
<thead>
<tr>
<th>Run No.</th>
<th>Global Optimum $x^*$</th>
<th>Optimum Value</th>
<th># F. Evl.</th>
<th>CPU (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0.2001, 0.1528, 0.4834, 0.2689, 0.3128, 0.6573)</td>
<td>-3.3207</td>
<td>86</td>
<td>16.7</td>
</tr>
<tr>
<td>2</td>
<td>(0.2002, 0.1496, 0.4786, 0.2570, 0.3125, 0.6601)</td>
<td>-3.3129</td>
<td>86</td>
<td>16.7</td>
</tr>
<tr>
<td>3</td>
<td>(0.2318, 0.1582, 0.4773, 0.2695, 0.3094, 0.6560)</td>
<td>-3.3090</td>
<td>86</td>
<td>16.7</td>
</tr>
<tr>
<td>4</td>
<td>(0.2021, 0.1491, 0.4761, 0.2754, 0.3121, 0.6574)</td>
<td>-3.3223</td>
<td>79</td>
<td>14.5</td>
</tr>
<tr>
<td>5</td>
<td>(0.2000, 0.1509, 0.4781, 0.2685, 0.3130, 0.6572)</td>
<td>-3.3209</td>
<td>86</td>
<td>16.7</td>
</tr>
<tr>
<td>6</td>
<td>(0.2008, 0.1516, 0.4780, 0.2650, 0.3144, 0.6577)</td>
<td>-3.188</td>
<td>86</td>
<td>16.8</td>
</tr>
<tr>
<td>7</td>
<td>(0.2119, 0.1474, 0.4748, 0.2751, 0.3115, 0.6562)</td>
<td>-3.3209</td>
<td>93</td>
<td>19.5</td>
</tr>
<tr>
<td>8</td>
<td>(0.2158, 0.1493, 0.4740, 0.2745, 0.3112, 0.6579)</td>
<td>-3.3199</td>
<td>79</td>
<td>14.1</td>
</tr>
<tr>
<td>9</td>
<td>(0.1938, 0.1499, 0.4754, 0.2766, 0.3127, 0.6581)</td>
<td>-3.3214</td>
<td>86</td>
<td>16.7</td>
</tr>
<tr>
<td>10</td>
<td>(0.2113, 0.1503, 0.4783, 0.2766, 0.3107, 0.6575)</td>
<td>-3.3211</td>
<td>79</td>
<td>14.1</td>
</tr>
</tbody>
</table>

Table B-4. Optimization Results on Six-hump Camel-back (SC) Function

<table>
<thead>
<tr>
<th># Iteration</th>
<th>Global Optimum $x^*$</th>
<th>Function Value</th>
<th># F. Evl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(-0.3153, -0.4559)</td>
<td>-0.0399</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>(0.0552, -0.5785)</td>
<td>-0.9104</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>(0.0875, -0.6046)</td>
<td>-0.9501</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>(0.1196, -0.6952)</td>
<td>-1.0252</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>(0.1196, -0.6952)</td>
<td>-1.0252</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>(0.0846, -0.7096)</td>
<td>-1.0315</td>
<td>26</td>
</tr>
<tr>
<td>7</td>
<td>(0.0846, -0.7096)</td>
<td>-1.0315</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>(0.0846, -0.7096)</td>
<td>-1.0315</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>(0.0896, -0.7123)</td>
<td>-1.0316</td>
<td>33</td>
</tr>
</tbody>
</table>
Table B-5. Optimization Results on Levy Function

<table>
<thead>
<tr>
<th># Iteration</th>
<th>Global Optimum $x^*$</th>
<th>Function Value</th>
<th># F. Evl.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(0.0000, 0.0000, 0.0000, 0.0000)</td>
<td>0.8975</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>(1.0525, 0.8465, -0.8464, 1.3558)</td>
<td>0.6400</td>
<td>33</td>
</tr>
<tr>
<td>12</td>
<td>(0.7183, -0.6420, 1.4062, 1.6220)</td>
<td>0.5313</td>
<td>50</td>
</tr>
<tr>
<td>20</td>
<td>(1.3519, 0.7595, 0.9411, 0.8561)</td>
<td>0.1788</td>
<td>68</td>
</tr>
<tr>
<td>30</td>
<td>(0.9447, 0.2660, 0.8530, 1.2262)</td>
<td>0.1069</td>
<td>92</td>
</tr>
<tr>
<td>35</td>
<td>(1.0531, 1.0121, 1.2148, 0.4418)</td>
<td>0.0616</td>
<td>103</td>
</tr>
<tr>
<td>40</td>
<td>(1.0053, 0.9858, 0.9484, 0.9663)</td>
<td>0.0015</td>
<td>115</td>
</tr>
<tr>
<td>43</td>
<td>(1.0072, 1.0130, 0.9625, 0.9180)</td>
<td>0.0013</td>
<td>117</td>
</tr>
<tr>
<td>45</td>
<td>(0.9939, 0.9907, 0.9985, 0.9834)</td>
<td>1.0230e-004</td>
<td>120</td>
</tr>
</tbody>
</table>

Table B-6. Optimization Results on Hartmann (H6) Function

<table>
<thead>
<tr>
<th># Iteration</th>
<th>Global Optimum $x^*$</th>
<th>Function Value</th>
<th># F. Evl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0.2137, 0.1804, 0.4452, 0.2697, 0.3259, 0.5777)</td>
<td>-3.1282</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>(0.1680, 0.1739, 0.4950, 0.3014, 0.3242, 0.6477)</td>
<td>-3.2685</td>
<td>33</td>
</tr>
<tr>
<td>8</td>
<td>(0.2103, 0.1697, 0.5126, 0.2748, 0.3215, 0.6425)</td>
<td>-3.2944</td>
<td>42</td>
</tr>
<tr>
<td>12</td>
<td>(0.1989, 0.1735, 0.4978, 0.2720, 0.3096, 0.6640)</td>
<td>-3.3107</td>
<td>52</td>
</tr>
<tr>
<td>16</td>
<td>(0.2051, 0.1595, 0.4781, 0.2731, 0.3057, 0.6564)</td>
<td>-3.3192</td>
<td>61</td>
</tr>
<tr>
<td>20</td>
<td>(0.2024, 0.1563, 0.4773, 0.2758, 0.3121, 0.6596)</td>
<td>-3.3217</td>
<td>70</td>
</tr>
<tr>
<td>24</td>
<td>(0.2021, 0.1491, 0.4761, 0.2754, 0.3121, 0.6574)</td>
<td>-3.3223</td>
<td>79</td>
</tr>
</tbody>
</table>
Figure B-1. Convergence Process of SEUMRE for Levy Function

Figure B-2. Convergence Process of SEUMRE for Hartmann (H6) Function
Figure B-3. Convergence Process of SEUMRE for C3 Function
Appendix C. Tested Multi-objective Optimization

Benchmark Problems Results’ Plots

Figure C-1. Performance Space and Evaluated Points of Five Runs for Test Problem 2

Figure C-2. Performance Space and Evaluated Points of Five Runs for Test Problem 3
Figure C-3. Pareto Frontier of Five Runs for Test Problem 3

Figure C-4. Performance Space and Evaluated Points of Five Runs for Test Problem 4
Figure C-5. Pareto Frontier of Five Runs for Test Problem 4

Figure C-6. Pareto Frontier of Five Runs for Test Problem 5