

5-15-2015

Uncertainty-Integrated Surrogate Modeling for Complex System Optimization

Ali Mehmani
Syracuse University

Follow this and additional works at: <http://surface.syr.edu/etd>

 Part of the [Engineering Commons](#)

Recommended Citation

Mehmani, Ali, "Uncertainty-Integrated Surrogate Modeling for Complex System Optimization" (2015). *Dissertations - ALL*. Paper 227.

This Dissertation is brought to you for free and open access by the SURFACE at SURFACE. It has been accepted for inclusion in Dissertations - ALL by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

ABSTRACT

Approximation models such as surrogate models provide a tractable substitute to expensive physical simulations and an effective solution to the potential lack of quantitative models of system behavior. These capabilities not only enable the efficient design of complex systems, but is also essential for the effective analysis of physical phenomena/characteristics in the different domains of Engineering, Material Science, Biomedical Science, and various other disciplines. Since these models provide an abstraction of the real system behavior (often a low-fidelity representative) it is important to quantify the accuracy and the reliability of such approximation models without investing additional expensive system evaluations (simulations or physical experiments). Standard error measures, such as the mean squared error, the cross-validation error, and the Akaike's information criterion however provide limited (often inadequate) information regarding the accuracy of the final surrogate model while other more effective dedicated error measures are tailored towards only one class of surrogate models. This lack of accuracy information and the ability to compare and test diverse surrogate models reduce the confidence in model application, restricts appropriate model selection, and undermines the effectiveness of surrogate-based optimization.

A key contribution of this dissertation is the development of a new model-independent approach to quantify the fidelity of a trained surrogate model in a given region of the design domain. This method is called the Predictive Estimation of Model Fidelity (PEMF). The PEMF method is derived from the hypothesis that *“the accuracy of an approximation model is related to the amount of data resources leveraged to train the model”*. In PEMF, intermediate surrogate models are iteratively constructed over heuristic subsets of sample points. The median and the maximum errors estimated over the remaining points are used to determine the respective error distributions at each iteration. The estimated modes of the error distributions are represented as functions of the density of intermediate training points through nonlinear regression, assuming a smooth decreasing trend of errors with increasing sample density. These regression functions are then used to predict the expected median and maximum errors in the final surrogate models. It is observed that the model fidelities estimated by PEMF are up to two orders of magnitude more accurate and statistically more stable compared to those based on the popularly-used leave-one-out cross-validation method, when applied to a variety of benchmark problems.

By leveraging this new paradigm in quantifying the fidelity of surrogate models, a novel automated surrogate model selection framework is also developed. This PEMF-based model selection framework is called the Concurrent Surrogate Model Selection (COSMOS). COSMOS, unlike existing model selection methods, coherently operates at all the three levels necessary to facilitate optimal selection, i.e., (1) selecting the model type, (2) selecting the kernel function type, and (3) determining the optimal values of the typically user-prescribed parameters. The selection criteria that guide optimal model selection are determined by PEMF and the search process is performed using a MINLP solver. The effectiveness of COSMOS is demonstrated by successfully applying it to different benchmark and practical engineering problems, where it offers a first-of-its-kind globally competitive model selection.

In this dissertation, the knowledge about the accuracy of a surrogate estimated using PEMF is applied to also develop a novel model management approach for engineering optimization. This approach adaptively selects computational models (both physics-based models and surrogate models) of differing levels of fidelity and computational cost, to be used during optimization, with the overall objective to yield optimal designs with high-fidelity function estimates at a reasonable computational expense. In this technique, a new adaptive model switching (AMS) metric defined to guide the switching of model from one to the next higher fidelity model during the optimization process. The switching criterion is based on whether the uncertainty associated with the current model output dominates the latest improvement of the relative fitness function, where both the model output uncertainty and the function improvement (across the population) are expressed as probability distributions. This adaptive model switching technique is applied to two practical problems through Particle Swarm Optimization to successfully illustrate: (i) the computational advantage of this method over purely high-fidelity model-based optimization, and (ii) the accuracy advantage of this method over purely low-fidelity model-based optimization.

Motivated by the unique capabilities of the model switching concept, a new model refinement approach is also developed in this dissertation. The model refinement approach can be perceived as an adaptive sequential sampling approach applied in surrogate-based optimization. Decisions regarding when to perform additional system evaluations to refine the model is guided by the same model-uncertainty principles as in the adaptive model switching technique. The effectiveness of this new model refinement technique is illustrated through application to practical surrogate-based optimization in the area of energy sustainability.

**UNCERTAINTY-INTEGRATED SURROGATE MODELING
FOR COMPLEX SYSTEM OPTIMIZATION**

By

Ali Mehmani

M.S., K.N.Toosi University of Technology, 2010

Dissertation

Submitted in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mechanical Engineering

Syracuse University
Syracuse, New York, USA

May 2015

© Copyright Ali Mehmani 2015

All Rights Reserved

DEDICATION

I dedicate this thesis to my wife, Gelare, for her continuous support and unwavering love through the best and the worst of times.

To my parents, without whose nurturing and encouragement, this work would have never been possible. I cannot imagine successfully finishing the mammoth task of doctoral work without their teachings and kindest love.

To my brother, Hamed, who has supported me all the way since the beginning of my studies.

To my mother and father-in-law for their love and moral support.

To my sister and brother-in-law, Tina and Amir. You both continue to support me through words and actions.

To my aunt for constant support and encouragement.

ACKNOWLEDGMENT

I would like to sincerely thank my advisor, Professor Achille Messac, without whose support I would not have been able to reach the position that I have. I appreciate all his contributions of time and ideas to make my PhD experience productive and stimulating. Much of what he taught will stay with me for the rest of my life. Prof. Messac gave me the confidence to push beyond what I thought I could do, which was pivotal in carrying out my research and completing my PhD. I am very proud to be his student and I will never forget this part of my life.

I would like to thank my co-adviser, Dr. Souma Chowdhury, for his continuous professional support and for the technical insights regarding this dissertation.

I would also like to thank my doctoral committee members, Prof. Uptal Roy, Prof. Jianshun Zhang, Prof. Mark Glauser, Prof. Jeongmin Ahn, and Prof. Dawit Negussey for their valuable advice, and for their willingness to serve on my committee. Special thanks to Prof. Uptal Roy, for his help and support in other areas of my academic life at the MAE department in SU.

I wish to extend my warmest thanks to those who have helped me with my work in the Multidisciplinary Design and Optimization Laboratory at Syracuse University, particularly my former colleague, Dr. Jie Zhang, and my dear friend and colleague, Weiyang Tong.

Sponsorship of this work by the National Science Foundation awards CMMI-1100948 and CMMI-1437746 is also gratefully acknowledged.

Special thanks to my friends, Habibollah Fakhraei, Shahriar Salehi, Shahed Shojaeipour, Hossein Jarrahi, Abbas Javaheri, Ashkan Noushabadi and Kaveh Yazdani for engaging me in enriching social and scientific discussions, and for creating a home away from home. I will always cherish their love and friendship.

These acknowledgements would not be complete without thanking the staff at Syracuse University, who have helped me in various aspects. I would like to thank Kathleen J Datthyn-Madigan, Kristin Shapiro, Kimberly Drumm-Underwood, and Deborah Brown at SU MAE department for all their efforts. A lot of friends made life at Syracuse University fun, I thank them all.

CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	ix
NOMENCLATURE	x

I Technical Preliminaries **xiv**

1. Background, Motivation, Objectives and Impact	1
1.1 Surrogate Modeling	1
1.1.1 Surrogate Model Construction	3
1.1.2 Testing The Accuracy of The Surrogate Model	4
1.1.3 Variable Fidelity Optimization	4
1.1.4 Model Refinement	5
1.2 Motivation, Objectives and Impact of the Research in this Dissertation	5
1.2.1 Research Motivation	5
1.2.2 Research Question	7
1.2.3 Research Objective	7
1.2.4 Research Impact	8
1.3 Dissertation Outline and Structure	9
2. Literature Survey	13
2.1 Surrogate Modeling	13
2.2 Model Uncertainty: Quantification and Impact	19
2.2.1 Uncertainty in Physics-based Modeling	20
2.2.2 Uncertainty in Surrogate Modeling	22
2.2.2.1 Global Error Measurement	23
2.2.2.2 Local Error Measurement	26
2.2.3 Observations and Research Needs	27
2.3 Model Selection	28
2.3.1 Existing Model Selection Method	28
2.3.2 Observations and Research Needs	31
2.4 Variable Fidelity Optimization	33

2.4.1	Model Switching & Model Management	34
2.4.2	Model Refinement & Surrogate-based Optimization	37
2.4.3	Observations and Research Needs	41
II Surrogate Model Uncertainty Quantification		43
3.	Predictive Estimation of Model Fidelity (PEMF)	44
3.1	Chapter Introduction	45
3.2	Predictive Estimation of Model Fidelity (PEMF)	45
3.2.1	Generating sample data	46
3.2.2	Defining the region of interest	47
3.2.3	Quantifying the variation of the error distribution with sample density	47
3.2.4	Predicting the fidelity of the final surrogate model	49
3.2.4.1	Constructing and Testing Functions to Represent the Variation of Error with Sample Density (VESD)	49
3.3	Investigating the Stability of PEMF error measures	53
3.4	Application of the PEMF Method	55
3.4.1	Surrogate models	55
3.4.2	Numerical Experiments	57
3.4.3	Performance Criteria	59
3.5	Results and Discussion	60
3.5.1	Investigating the Performance of PEMF	60
3.5.1.1	Comparing PEMF with <i>cross-validation</i>	64
3.5.2	Application of PEMF to higher dimensional benchmark problems . .	65
3.6	Chapter Summary	69
4.	Practical Implementation & Parametric Analysis of Predictive Estimation of Model Fidelity (PEMF)	71
4.1	Chapter Introduction	71
4.2	Statistical test on results of PEMF and <i>leave-one-out</i> cross-validation method	73
4.3	Parametric Analysis on PEMF's Parameters	80
4.4	Application of the PEMF Method on Wind Farm Power Generation Problem	85
4.5	Chapter Summary	91
III Model Selection & Variable Fidelity Optimization		93

5.	Concurrent Surrogate Model Selection (COSMOS) based on Predictive Estimation of Model Fidelity	94
5.1	Chapter Introduction	95
5.2	Development of COSMOS	96
5.2.1	3-Level automated model selection	96
5.2.1.1	Cascaded technique	98
5.2.1.2	One-Step technique	101
5.3	Predictive Estimation of Model Fidelity (PEMF) in COSMOS framework . .	107
5.4	Numerical Experiments with COSMOS	110
5.5	Results and Discussion	113
5.6	Chapter Summary	123
6.	Variable-Fidelity Optimization with Adaptive Model Switching (AMS)	127
6.1	Chapter Introduction	128
6.2	Variable Fidelity Optimization with Adaptive Model Switching (AMS) . . .	129
6.2.1	Major steps in Optimization with AMS	129
6.2.2	The Adaptive Model Switching (AMS) Metric	131
6.2.3	Quantifying Model Uncertainties	135
6.2.4	Optimization Algorithm: Particle Swarm Optimization	138
6.3	Numerical Experiments with AMS	139
6.3.0.1	Shape Optimization of a Cantilever Composite Beam	139
6.3.0.2	Structural models with different levels of fidelity	141
6.4	Results and Discussion	142
6.5	Chapter Summary	146
7.	Variable-Fidelity Optimization with Adaptive Model Refinement (AMR)	148
7.1	Chapter Introduction	149
7.2	Surrogate-based optimization with Adaptive Model Refinement	150
7.2.1	Major steps in Surrogate-based Optimization with AMR	150
7.2.2	The Adaptive Model Refinement (AMR)	153
7.3	Numerical Experiments with AMR	155
7.4	Results and Discussion	157
7.5	Chapter Summary	162

**IV Practical Application:
Error Quantification, Model Selection, Variable Fidelity Opti-
mization** **165**

8. Application of PEMF-derived Methods to CFD-based Problems	166
8.1 Chapter Introduction	166
8.2 Numerical Case Studies with COSMOS	167
8.2.1 Airfoil Design Problem:	167
8.2.2 Three-Pane Window Heat Transfer Problem:	168
8.2.3 Results and Discussion	169
8.3 Numerical Case Study with Adaptive Model Switching	175
8.3.1 Aerodynamic shape optimization of 2D airfoil	175
8.3.2 Aerodynamic models with different level of fidelity	176
8.3.3 Results and Discussion	179
8.4 Chapter Summary	183
9. Application of PEMF-derived Methods to a Complex Practical Problem	186
9.1 Chapter Introduction	186
9.2 A Very Large Utility-scale Wind Farm	187
9.3 Mapping of the Layout for a Large Scale Wind Farm	188
9.4 Energy Production Model	190
9.5 Surrogate Model Selection using COSMOS	196
9.5.1 Results and Discussion	198
9.6 Variable Fidelity Optimization using AMR	199
9.6.1 Results and Discussion	201
9.7 Chapter Summary	204

V Concluding Remarks and Future Work 206

10. Concluding Remarks and Future Work	207
10.1 Summary of Contributions in the Dissertation	207
10.2 Limitations and Future Work	210
10.2.1 Predictive Estimation of Model Fidelity (PEMF)	211
10.2.2 Concurrent Surrogate Model Selection (COSMOS)	211
10.2.3 Variable Fidelity Optimization	212
10.2.4 Conservative surrogate model	214
10.2.5 Hybrid surrogate model	214
10.2.6 Large-scale applications	214

VI REFERENCES	216
----------------------	------------

VII APPENDICES	235
-----------------------	------------

APPENDICES

A. Gambit and Fluent Journal Files: 2D airfoil design problem	236
B. The List of Probability Distribution Functions	247
B.1 Weibull Distribution	247
B.2 Gamma Distribution	248
B.3 Normal Distribution	248
B.4 Lognormal Distribution	249
B.5 Rayleigh Distribution	249

VIII CURRICULUM VITAE	251
------------------------------	------------

LIST OF TABLES

2.1	Techniques for surrogate modeling	14
2.2	Basis or Kernel functions and their hyper-parameters in the Surrogate Models .	20
2.3	Techniques for surrogate model construction and validation	23
2.4	Optimization algorithms	41
3.1	Numerical setup for the benchmark functions	57
3.2	Median Error Estimated using <i>PEMF</i> and a large pool of additional test points (E_{Mo-med}^{Actual})	60
3.3	Maximum Error Estimated using <i>PEMF</i> and a large pool of additional test points (E_{Mo-max}^{Actual})	64
3.4	Relative differences of the regional errors evaluated using <i>PEMF</i> and <i>CV</i> from the actual errors in Branin-Hoo Function (case <i>I</i>)	65
3.5	Relative differences of the global errors evaluated using <i>PEMF</i> and <i>CV</i> from the actual errors in Branin-Hoo Function (case <i>II</i>)	66
3.6	Relative differences of the global errors evaluated using <i>PEMF</i> and <i>CV</i> from the actual errors in Perm Function	69
3.7	Relative differences of the global errors evaluated using <i>PEMF</i> and <i>CV</i> from the actual errors in Dixon & Price Function	69
4.1	90% bootstrapped CIs for residuals of E_{med}^{PEMF} and E_{mean}^{CV} in different surrogates	78
4.2	90% bootstrapped CIs for residuals of E_{max}^{PEMF} and E_{max}^{CV} in different surrogates	78
4.3	Functions linst for Parametric Analysis	82
5.1	Candidate model-kernel combinations and their integer-codes	103
5.2	Range of hyper-parameters	105
5.3	Basis or Kernel functions and their hyper-parameters in the candidate surrogate models	111
5.4	Problem properties and optimization settings	113
5.5	The set of Pareto Surrogate Models given by (i) Cascaded technique, (ii) One- Step technique, and (iii) Model selection based on actual error	116

5.6	Comparison of the performance of the COSMOS with the current paradigm in benchmark problems	121
5.7	The set of Pareto Surrogate Models given by COSMOS	123
6.1	Design variables for the cantilever composite beam design problem	140
6.2	Specified fixed design parameters for the beam design problem	140
6.3	Models with different levels of fidelity used in the cantilever beam optimization problem (the high fidelity model is assumed to be a true representation of the system behavior)	143
6.4	Cantilever beam design: Optimization results using different approaches	145
7.1	Optimization results using different optimization approaches	160
8.1	Three-pane window geometry	169
8.2	Problem properties and optimization settings	169
8.3	The set of Pareto Surrogate Models given by (i) Cascaded technique, and (ii) One-Step technique	170
8.4	Comparison of the performance of the COSMOS with the current paradigm in engineering problems	173
8.5	The set of Pareto Surrogate Models given by COSMOS	174
8.6	Design variables in airfoil optimization problem	176
8.7	Models with different levels of fidelity used in the airfoil optimization problem (the high fidelity model is assumed to be a true representation of the system behavior)	180
8.8	2D Airfoil Design: Optimization results using different optimization approaches	183
9.1	Upper and lower bounds of mapping design factors	190
9.2	Features of the <i>GE-1.5MW-XLE</i> turbine	194
9.3	Details of the NDAWN station at Baker, ND [118]	195
9.4	The set of Pareto Surrogate Models given by COSMOS for the wind farm problem	198
9.5	Upper and lower bounds of design variables	200
9.6	Optimization results using different optimization approaches	204

LIST OF FIGURES

1.1	Applications of Surrogate Modeling	2
1.2	Graphical description of dissertation outlines and structure	12
2.1	Chapter 2 orientation	13
2.2	A generic topology of neural networks [2]	18
2.3	ϵ -insensitive loss function	20
2.4	The impact of shape hyper parameter (σ) in a RBF surrogate with Multiquadric basis function constructed for a 1-D problem; $f(x) = (6x - 2)^2 \sin[2(6x - 2)]$, $x \in [-1, 1]$	30
2.5	Elements of 3-level Surrogate Model Selection	32
2.6	Variable Fidelity Models	35
2.7	Sampling strategies	38
3.1	Chapter 3 orientation	44
3.2	The Predictive Estimation of Model Fidelity (PEMF) algorithm	46
3.3	Variation of Estimated Modal values of Median Errors with Sample Density using PEMF (Branin-Hoo Function)	54
3.4	Variation of typical Mean Error with Sample Density, as given by standard k -fold cross-validation (Branin-Hoo Function)	54
3.5	VESD function to predict the regional median error ($E_{\text{med}}^{\text{PEMF}}$) in Branin-Hoo Function (case <i>I</i>)	61
3.6	Distribution determination based on the χ^2 criterion on median regional error frequency diagrams in Kirigng model constructed for Branin-Hoo Function (case <i>I</i>)	61
3.7	VESD function to predict the global median error ($E_{\text{med}}^{\text{PEMF}}$) in Branin-Hoo Function (case <i>II</i>)	62
3.8	Distribution determination based on the χ^2 criterion on median global error frequency diagrams in Branin-Hoo Function (case <i>II</i>)	62
3.9	VESD function to predict the regional maximum error ($E_{\text{max}}^{\text{PEMF}}$) in Branin-Hoo Function (case <i>I</i>)	63

3.10	VESD function to predict the global maximum error (E_{\max}^{PEMF}) in Branin-Hoo Function (case <i>II</i>)	64
3.11	Comparison of the <i>PEMF</i> and <i>CV</i> error measures with the Actual regional error in Branin-Hoo Function (case <i>I</i>)	65
3.12	Comparison of the <i>PEMF</i> and <i>CV</i> error measures with the Actual global error in Branin-Hoo Function (case <i>II</i>)	66
3.13	VESD function to predict the global median error ($E_{\text{med}}^{\text{PEMF}}$) in Perm Function	67
3.14	VESD function to predict the global maximum error (E_{\max}^{PEMF}) in Perm Function	67
3.15	VESD function to predict the global median error ($E_{\text{med}}^{\text{PEMF}}$) in Dixon & Price Function	68
3.16	VESD function to predict the global maximum error (E_{\max}^{PEMF}) in Dixon & Price Function	68
3.17	Comparison of the <i>PEMF</i> and <i>CV</i> error measures with the Actual global error in Perm Function	68
3.18	Comparison of the <i>PEMF</i> and <i>CV</i> error measures with the Actual global error in Dixon & Price Function	69
4.1	Chapter 4 orientation	71
4.2	Distribution of the Residual of median error estimated using PEMF ($E_{\text{med}}^{\text{PEMF}}$) for Branin-Hoo Function	74
4.3	Distribution of the Residual of mean error estimated using <i>leave-one-out cross-validation</i> ($E_{\text{mean}}^{\text{CV}}$) for Branin-Hoo Function	75
4.4	Distribution of the Residual of maximum error estimated using PEMF (E_{\max}^{PEMF}) for Branin-Hoo Function	76
4.5	Distribution of the Residual of maximum error estimated using <i>leave-one-out cross-validation</i> E_{\max}^{CV} for Branin-Hoo Function	77
4.6	Interquartile Range box plot of the residuals of the central tendency of surrogate model errors (median in PEMF; and mean in <i>leave-one-out cross-validation</i>) for Branin-Hoo Function	78
4.7	Interquartile Range box plot of the residuals of the maximum surrogate error for Branin-Hoo Function	79
4.8	Function 1 (6D) - the normalized bias of the PEMF method in different combinations of β and δ in Case I and Case II	83
4.9	Function 2 (10D) - the normalized bias of the PEMF method in different combinations of β and δ in Case I and Case II	84

4.10	Function 3 (10D) - the normalized bias of the PEMF method in different combinations of β and δ in Case I and Case II	85
4.11	Function 4 (10D) - the normalized bias of the PEMF method in different combinations of β and δ in Case I and Case II	86
4.12	Function 5 (16D) - the normalized bias of the PEMF method in different combinations of β and δ in Case I and Case II	87
4.13	Function 1 (6D) : Comparison of the PEMF and CV-derived error measures .	87
4.14	Function 2 (10D) : Comparison of the PEMF and CV-derived error measures	88
4.15	Function 3 (10D) : Comparison of the PEMF and CV-derived error measures	88
4.16	Function 4 (10D) : Comparison of the PEMF and CV-derived error measures	89
4.17	Function 5 (16D) : Comparison of the PEMF and CV-derived error measures	89
4.18	Wind farm array schematic	90
4.19	Comparison of the performance of the PEMF with Actual error in the Wind Farm Power Generation problem	91
4.20	Comparison of the <i>PEMF</i> and <i>CV</i> error measures with the Actual global error in Wind Farm Power Generation problem	92
5.1	Chapter 5 orientation	94
5.2	The inputs and outputs of the PEMF method	98
5.3	The 3-level Cascaded model selection technique based on PEMF	100
5.4	Optimal solutions given by the multiobjective hyper-parameter optimization with the SVR-Linear model for the Branin-Hoo test function	101
5.5	The 3-level <i>One-Step</i> model selection technique(M: Surrogate model type; K: Kernel function type; HP: Hyper-parameter; u : vector of Hyper-parameter values)	104
5.6	Optimal solutions given by the simultaneous model selection using One-Step technique for the Branin-Hoo test function	106
5.7	Application of COSMOS in surrogate constructed for Branin-Hoo Function . .	115
5.8	Trade-offs between modal values of median and maximum error - Branin-Hoo test function (2-variable) : Pareto models and final population of models from all Φ classes	118
5.9	Trade-offs between modal values of median and maximum error - Hartmann test function (3-variable) : Pareto models and final population of models from all Φ classes	119

5.10	Computational time of the two different 3-level model selection techniques . . .	119
5.11	Trade-offs between modal values of median and maximum error - Perm test function (10-variable) : Pareto models and final population of models from all Φ classes	120
5.12	Trade-offs between modal values of median and maximum error - Dixon & Price test function (50-variable) : Pareto models and final population of models from all Φ classes	120
5.13	Branin-Hoo test function (2-variable) : Pareto models and final population of models from all Φ classes under Test II and Test III	124
5.14	Hartmann test function (3-variable) : Pareto models and final population of models from all Φ classes under Test II and Test III	124
5.15	Perm test function (10-variable) : Pareto models and final population of models from all Φ classes under Test II and Test III	125
5.16	Dixon & Price test function (50-variable) : Pareto models and final population of models from all Φ classes under Test II and Test III	125
6.1	Chapter 6 orientation	127
6.2	Adaptive Model Switching in population-based optimization	131
6.3	The illustration of the AMS Metric	133
6.4	The illustration of the AMS hypothesis test (comparing the model error distribution (\mathbb{P}_i) and the distribution of fitness function improvement (Θ))	134
6.5	Cantilever composite beam subjected to a parabolic distributed load	139
6.6	Distributions of the model errors for the cantilever beam design problem	142
6.7	Distribution of the fitness function improvements in different iterations of the beam optimization with PSO-AMS (also showing the model error distributions)	144
6.8	Optimization history of the cantilever beam optimization with PSO-AMS	145
6.9	Percentage-contribution of each model in the cantilever beam optimization problem using PSO-AMS	146
7.1	Chapter 7 orientation	148
7.2	The inputs and outputs of the PEMF method	151
7.3	Adaptive Model Refinement in Surrogate-based Optimization	152
7.4	The improvement of the surrogate model fidelity through sequential model refinement process with AMR	158

7.5	Distribution of the fitness function improvements in different iterations of the global optimization problem with AMR-RBF (also showing the model error distributions). Rosenbrock function	159
7.6	Rosenbrock function (5-variable): Optimization history of the global optimization problem	161
7.7	Hartmann function (6-variable): Optimization history of the global optimization problem	162
7.8	Levy function (10-variable): Optimization history of the global optimization problem	163
8.1	Chapter 8 orientation	166
8.2	Design variables governing the shape geometry of the Airfoil	168
8.3	Schematic of the three-pane window [132]	169
8.4	Trade-offs between modal values of median and maximum error - Airfoil Design: Pareto models and final population of models from all Φ classes	172
8.5	Trade-offs between modal values of median and maximum error - Window model: Pareto models and final population of models from all Φ classes	172
8.6	Airfoil Design: Pareto models and final population of models from all Φ classes	174
8.7	Window model: Pareto models and final population of models from all Φ classes	175
8.8	Fine and coarse mesh for CFD of airfoil [131]	177
8.9	Pressure field around the airfoil at a baseline design	177
8.10	Distribution of the model errors: aerodynamic C_L/C_D ratio of the 2D airfoil	179
8.11	Distribution of the fitness function improvements in different iterations of the airfoil optimization with PSO-AMS (also showing the model error distributions)	181
8.12	Optimization history of the airfoil design problem	182
8.13	Percentage-contribution of each model in the airfoil optimization problem performed by PSO-AMS	184
9.1	Chapter 9 orientation	186
9.2	Input and output structure of the Wind Farm Layout Mapping	190
9.3	Input and output structure of the average annual energy production model	194
9.4	Baker station setup [118]	195
9.5	Wind rose diagram for the site at Baker	196

9.6	Model selection for the average annual energy production of a large-scale wind farm	197
9.7	Pareto models and final population of models from all Φ classes	199
9.8	Wind farm problem: Each tile shows a contour of the wind farm average annual energy production versus two of the six mapping factors (variables), with the remaining four variables held at the baseline value	201
9.9	The improvement of the surrogate model fidelity through sequential model refinement process with AMR	203
9.10	Convergence history of the surrogate-based optimization using AMR	203
9.11	Illustration of the optimized wind farm layout	205
10.1	Chapter 10 orientation	207

NOMENCLATURE

$\{X\}$	Set of sample points
$\{X_{\text{TR}}\}$	Training data set
$\{X_{\text{TE}}\}$	Test data set
N^{itr}	Number of iteration in PEMF
β	Fraction fold in the first iteration of PEMF
δ	which controls the Incremental value (step size) in PEMF
$N(\cdot)$	Size of data set $\{\cdot\}$
E_{med}	Median error of surrogate
E_{max}	Maximum error of surrogate
$E_{\text{med}}^{\sigma^2}$	Variance of the median error
$E_{\text{max}}^{\sigma^2}$	Variance of the maximum error
Mo^{med}	Mode of the median error
Mo^{max}	Mode of the maximum error
δ	Bias of error
m	Surrogate model type in COSMOS
k	Kernel type in COSMOS
σ	RBF shape parameter
θ	Kriging shape parameter
C	SVR penalty parameter
γ	SVR kernel parameter
ρ	Indicator of Monotonicity in PEMF

Q	Quantile of the error distribution
P_{cr}	Critical probability on AMS
Γ^{Infill}	Batch size of infill points in AMR
E	Young's modulus
C_L	Lift coefficient
C_D	Drag coefficient
P_{farm}	Power generated by a wind farm (MW)
C_p	Power coefficient of a wind turbine
a	Induction factor of a wind turbine
P_0	Rated power of a wind turbine (MW)
C_{in}	Installation cost per kilowatt installed ($\$/kW$)
C_t	Total annual cost per kilowatt installed ($\$/kW$)
D	Rotor diameter of a wind turbine (m)
H	Hub height of a wind turbine (m)
$D_{wake,j}$	The diameter of the expanding wake at a distance s behind Turbine j (m)
U	Wind speed (m/s)
z_0	The average roughness length in the farm region (m)

Subscripts and Superscripts

(\sim)	indicates approximate quantity
i, j, k	dummy indices
l	lower limit of a variable
u	upper limit of a variable

$()^+$ indicates matrix transpose

Acronyms

AEP	Annual Energy Production
AIC	Akaik's Information
AMR	Adaptive Model Refinement
AMS	Adaptive Model Switching
ANN	Artificial Neural Networks
COE	Cost of Energy
COSMOS	Concurrent Surrogate Model Selection
CFD	Computational Fluid Dynamics
CV	Cross Validation
DOE	Design of Experiments
E-RBF	Extended Radial Basis Functions
EA	Evolutionary Algorithms
FEA	Finite Element Analysis
IoC	Indicator of Conservativeness
KDE	Kernel Density Estimation
KKT	Karush Kuhn Tucker
LHS	Latin Hypercube Sampling
LRM	Linear Reference Model
MAE	Maximum Absolute Error
MDO	Multidisciplinary Design Optimization

MDPSO	Mixed-Discrete PSO
MINLP	Mixed Integer Nonlinear Programing
MTC	Monotonic Trend Criterion
NDAWN	North Dakota Agricultural Weather Network
N-RBF	Non-Radial Basis Functions
PEMF	Predictive Estimation of Model Fidelity
PLF	Physics-based Low Fidelity model
PRESS	Prediction Sum of Squares
PRSM	Polynomial Response Surface Methodology
PSO	Particle Swarm Optimization
QRSM	Quadratic Response Surface Methodology
RAE	Relative Accuracy Error
RBF	Radial Basis Functions
RMSE	Root Mean Squared Error
SBO	Surrogate-Based Optimization
SM	Surrogate Model
SVM	Support Vector Machine
SVR	Support Vector Regression
TLF	Tuned Low Fidelity Model
UWFLO	Unrestricted Wind farm Layout Optimization
VESD	Variation of Error with Sample Density
VFO	Variable Fidelity Optimization

PART I

Technical Preliminaries

CHAPTER 1

Background, Motivation, Objectives and Impact

In this chapter, an overview of the research performed in this dissertation is presented. We begin by providing the technical context and background of the research. Research objectives, motivation, and impact are then presented, followed by the outline of the dissertation.

1.1 Surrogate Modeling

Complex systems such as an aircraft or a wind farm, often involve a large number of design parameters, and the analysis of the performance of such systems generally demands computationally expensive simulations and/or expensive physical experiments. The direct application of simulations/experiments in a design process can be anywhere between challenging to prohibitive from a time/cost perspective. As a result, designers often choose to (i) simplify the model for the sake of computational efficiency by making assumptions, or (ii) limit the design space exploration. Surrogate modeling is a more pragmatic, yet highly effective solution to this challenge in complex system design. Surrogate models are commonly used to provide a tractable and inexpensive approximation of the actual system behavior in many routine engineering analysis and design activities, e.g., domain exploration, sensitivity analysis, development of empirical models, and optimization. In the design literature, surrogate models, which are purely mathematical models and are not directly derived from the physics of the system being modeled, are referred to by several names: response surfaces, approximation models, or metamodels. Figure 1.1 illustrates the different areas of surrogate modeling application. The process of surrogate modeling construction can be defined as

follows: “Given a set of input data points $x^i \in R^m, i = 1, \dots, n_p$, and the corresponding output/function values, $f(x^i)$, obtain a global approximation function, $\tilde{f}(x)$, that represents the original function over a given design domain”.

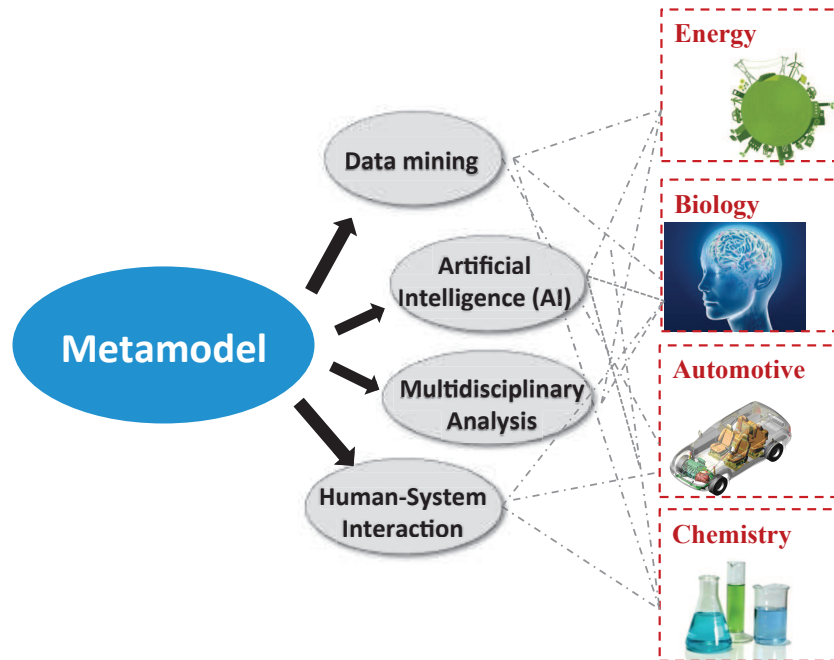


Figure 1.1: Applications of Surrogate Modeling

The process of surrogate modeling generally involves the following steps:

1. Performing a design of experiments (DOE);
2. Executing the sample experiments (e.g., high fidelity simulations or physical experiments) to generate the training data;
3. Selecting and training the appropriate surrogate model using the training data; and
4. Testing the accuracy of the surrogate models (prior to model implementation).

The selection of points in the design variable space where the response should be evaluated is commonly referred to as a Design of Experiments (DOE). DOE techniques are

originally developed to study the behavior of systems through physical experiments (e.g., in experimental chemistry [1]). *DOE can be defined as the design of a controlled information gathering exercise for a system or phenomenon where variation is known to be present* [2]. Although traditionally, controlled experiments referred only to physical experiments, in modern times, they would include both physical experiments and computational experiments (i.e., simulation-based experiments such as Finite Element Analysis). The planning of a DOE (i.e., the choice of the set of points in the design variable space) can have a considerable impact on the accuracy and the efficiency of the surrogate model. The typical DOE strategy is to generate a distribution of sample points throughout the design space in a uniform fashion. There are many techniques to determine how to distribute the sample points, including Factorial and Central Composite designs [3], Latin Hypercube design [4], Orthogonal arrays [5], and Sobol sequence [6]. In practice, the number of sample points is, in general, limited (due to computational expense or budget).

1.1.1 Surrogate Model Construction

The sample points used to construct surrogate models are generally called *training points*. Once an appropriate DOE has been performed and the training points have been defined, the next step is to choose an approximating model and the training method. Both parametric and nonparametric methods are available for surrogate model construction. Major surrogate modeling methods include Polynomial Response Surfaces [7], Kriging [8, 9], Radial Basis Functions (RBF) [10], Neural Networks [11], Support Vector Regression [12], and hybrid surrogates [13, 14]. In Chapter 2, we present a review of the state-of-the-art surrogate modeling techniques.

1.1.2 Testing The Accuracy of The Surrogate Model

The final step in the surrogate modeling process is to evaluate the expected accuracy or inversely the estimation error of the surrogate model. Two popular error metrics are (i) the *root mean squared error* (RMSE), which is a global error measure and provides an understanding of the model accuracy over the entire design domain; and (ii) the *maximum absolute error* (MAE), which provides an understanding of the maximum local deviation of the model estimation from the actual output. Popular approaches for assessing these error metrics include [13]: (i) split sample, (ii) cross-validation, (iii) boot-strapping, and (iv) Akaike’s information criterion (AIC).

The knowledge of the regional and global accuracy of a surrogate is crucial for (i) model selection, (ii) further improvement of the surrogate model (model refinement), e.g., using adaptive sampling [15] or active learning [16], (iii) iterative surrogate-based optimization [17, 18], and (iv) quantifying the uncertainty associated with the surrogate model. Other applications of error metrics include the construction of hybrid models and conservative surrogate models. A detailed review and discussion on state-of-the-art measures of surrogate model fidelity and their application is provided in Chapter 2.

1.1.3 Variable Fidelity Optimization

The major pitfall in using low-fidelity models (e.g., surrogate models) in optimization is that they can often mislead the search process due to underestimation or overestimation of a criteria function, leading to suboptimal or infeasible solutions. To address this issue and provide optimum designs with high-fidelity system evaluations, “Variable Fidelity Optimization” methods can be applied. In these methods, model switching strategies integrate models of different fidelity into the optimization process, with the overall objective to identify opti-

mal designs that have high fidelity function estimates at a reasonable computational expense. A detailed review of Variable Fidelity Optimization approaches is given in Chapter 2.

1.1.4 Model Refinement

In spite of the rapid growth of the surrogate modeling paradigm, the low fidelity of these models often limits their use in practical engineering design optimization. When surrogate models are used to represent the behavior of a complex system, it is often challenging to simultaneously obtain high accuracy over the entire design space. When such surrogates are used for optimization, it becomes challenging to find the optimum/optima with desired certainty. “Model Refinement” approaches offer a powerful solution to this challenge. In these approaches, the accuracy and robustness of the surrogate model is improved by adding points where additional evaluations of the high fidelity model / experiment are desired to be performed. ‘Model Refinement’ methods used to improve the accuracy of the surrogate models during the optimization process, it is known as “Surrogate-based Optimization (SBO)”. In typical SBO, the optimization process is repeated multiple times, where each time the accuracy of the surrogate model is improved by the addition of new sample points. A detailed review of Model Refinement approaches is given in Chapter 2.

1.2 Motivation, Objectives and Impact of the Research in this Dissertation

1.2.1 Research Motivation

In the era of complex system design, surrogate models are being considered integral components of the design process, especially as an alternative or complement to computationally expensive simulations and physical experiments. Surrogate models have been widely

incorporated in the leading commercially analysis and optimization packages (e.g., Genesis, modeFRONTIER, and iSIGHT).

However the construction and use of surrogate models is almost always plagued with one or both of the following realities: (i) scarcity of data, since high-fidelity data is expensive, and (ii) lack of knowledge of the underlying functional form of the system behavior being modeled. As a result, surrogate models are generally expected to introduce a degree of uncertainty in characterizing the real system behavior. In surrogate-based analysis and optimization, the quantitative knowledge of the uncertainty arising from model inaccuracies is critical to estimating the reliability of the final feasible or optimal designs. As such, developing a reliable and model-independent measure of surrogate uncertainty, which does not require additional system evaluations, is critical to the success of these models in practical complex system design.

Owing to the availability of multiple surrogate modeling techniques with diverse functional characteristics, the selection of the most suitable surrogate model for a given experimental or simulation data set is far from intuitive. This selection is also critical to facilitate effective and reliable usage of surrogate models in any application. A systematic surrogate model selection framework that coherently operates at three levels (i.e., choice of model type, kernel type, and hyper-parameter value) is thus needed.

Population-based heuristic optimization algorithms (e.g., evolutionary algorithms and swarm optimization algorithms) have been proven to be very effective in solving complex design optimization problems, especially those involving highly nonlinear functions. The computational cost of the high-fidelity simulation models (e.g., CFD, FEA models) and the large number of function evaluations (often demanded by heuristic algorithms) limit their applicability to practical complex system design. One approach to address this issue

is Variable Fidelity Optimization, which effectively integrates the information from models with different levels of fidelity into the heuristic optimization process is needed.

1.2.2 Research Question

How to understand, analyze, and quantify the regional and global fidelity of surrogate models, without having to exploit its functional characteristics (i.e., model-independence), and while remaining minimally sensitive to undesirable outliers? In successfully addressing this question, we can lay the foundation to address the following corollary questions: (i) How to construct a rigorous framework to perform globally competitive surrogate model selection; and (ii) How to formulate a model-based optimization process that integrates models of varying fidelity to offer high computational efficiency while not compromising the reliability of the optimal design decisions.

1.2.3 Research Objective

The primary goal of this dissertation is to develop a ‘*model-independent*’ approach to quantifying the fidelity of the ‘*actual surrogate*’ without requiring additional system evaluations, and use this model fidelity measure to formulate novel strategies for global model selection, adaptive model switching, and model refinement (in variable fidelity optimization). In this context, the actual surrogate refers to the surrogate model constructed using all available training points (and not any subset of training points, as is the case with cross-validation). Specific research objectives are as follows:

(I) Predictive Estimation of Model Fidelity (PEMF)

To formulate a predictive approach to estimating surrogate model fidelity and explore the effectiveness of the ensuing error measures by applying them to different

benchmark problems of varying dimensionality.

(II) **Automated surrogate model selection**

To develop a new automated surrogate model selection framework which allows the simultaneous selection of the most suitable surrogate type, kernel type, and hyperparameter values from a comprehensive candidate pool of surrogates through a synergy of PEMF and Mixed Integer Nonlinear Programming (MINLP).

(III) **Model switching metric for variable fidelity optimization**

To develop a new model switching approach to integrate models with different levels of fidelity (including physics-based analytical models, surrogate models, and computational models) into a population-based optimization process, such as to obtain high-fidelity design decision at a reasonable computational expense.

(IV) **Model refinement approach for surrogate-based optimization**

To develop a new model-independent approach to refining surrogate models during optimization, through adaptive sampling that preserves a desired level of fidelity and robustness.

(V) To apply the new error estimation (PEMF), model selection, and variable fidelity optimization methods to practical engineering design problems in order to explore the effectiveness of these methods.

1.2.4 Research Impact

The new surrogate model error estimation paradigm developed here, can provide uniquely useful information about the accuracy of the surrogate model in regions of in-

terest or in the entire design domain. This method can extract more pertinent information regarding the model (compared to most of the existing methods), which exploiting limited resources already available to construct surrogate models. This error estimation in turn significantly improves the usability of the surrogate model in real life complex system design applications.

The proposed model selection framework to select the globally-best surrogate model, will allow for radically improved design space exploration capability. This framework can encourage the modeling and analysis community to develop more advanced analysis tools that can better capture physical behaviors of a system.

The proposed approach in variable fidelity and surrogate-based design optimization will allow more effective and powerful MDO methodologies to be developed, which will ultimately lead to the strengthening of the design process.

It should be noted that, application of surrogate modeling is not restricted to the engineering design. The frameworks and approaches developed in this dissertation have the potential to significantly impact other non-engineering fields that involve expensive experiments (simulation-based or physical experiments), such as chemistry, material science, finance (for cost modeling), and biomedical science.

1.3 Dissertation Outline and Structure

This section provides a basic outline of the remainder of this dissertation, which is also illustrated in Fig. 1.2. As shown in the figure, the dissertation is divided into four major parts, which are listed below:

- **Part I Technical Preliminaries**

This part presents the necessary technical preliminaries pertaining to the content of the

dissertation. Chapter 2 reviews the literature in the areas associated with the research objectives of this dissertation and discussion the gaps in the current paradigm.

- **Part II Surrogate Model Uncertainty Quantification**

This part describes the formulation of the error estimation method. Chapter 3 provides the mathematical formulations of the method called the Predictive Estimation of Model Fidelity (PEMF), followed by the validation of the method through a set of standard numerical examples. Further statistical analysis and parametric analysis of PEMF are presented in Chapter 4

- **Part III Model Selection & Variable Fidelity Optimization**

This part focuses on the implementation of PEMF (from Part II) in the development of model selection and variable fidelity optimization methods. Chapter 5 develops the Concurrent Surrogate Model Selection (COSMOS) framework based on the Predictive Estimation of Model Fidelity (PEMF), which is a novel approach to identifying a globally-competitive surrogate models. Chapter 6 develops a new model management approach for engineering analysis. Chapter 7 develops a new adaptive surrogate-based optimization (SBO) approach.

- **Part IV Practical Application: Error Quantification, Model Selection, Variable Fidelity Optimization**

Chapters 8 and 9 of this dissertation present the application of the PEMF-based

methodologies (from Chapters 5, 6, and 7) to practical engineering design problems, including CFD-based problems and design problems in the area of energy sustainability.

- **Part V Concluding Remarks and Future Work**

Chapter 10 of this dissertation provides concluding remarks for this dissertation and outlines potential directions for further research in the concerned areas of interest.

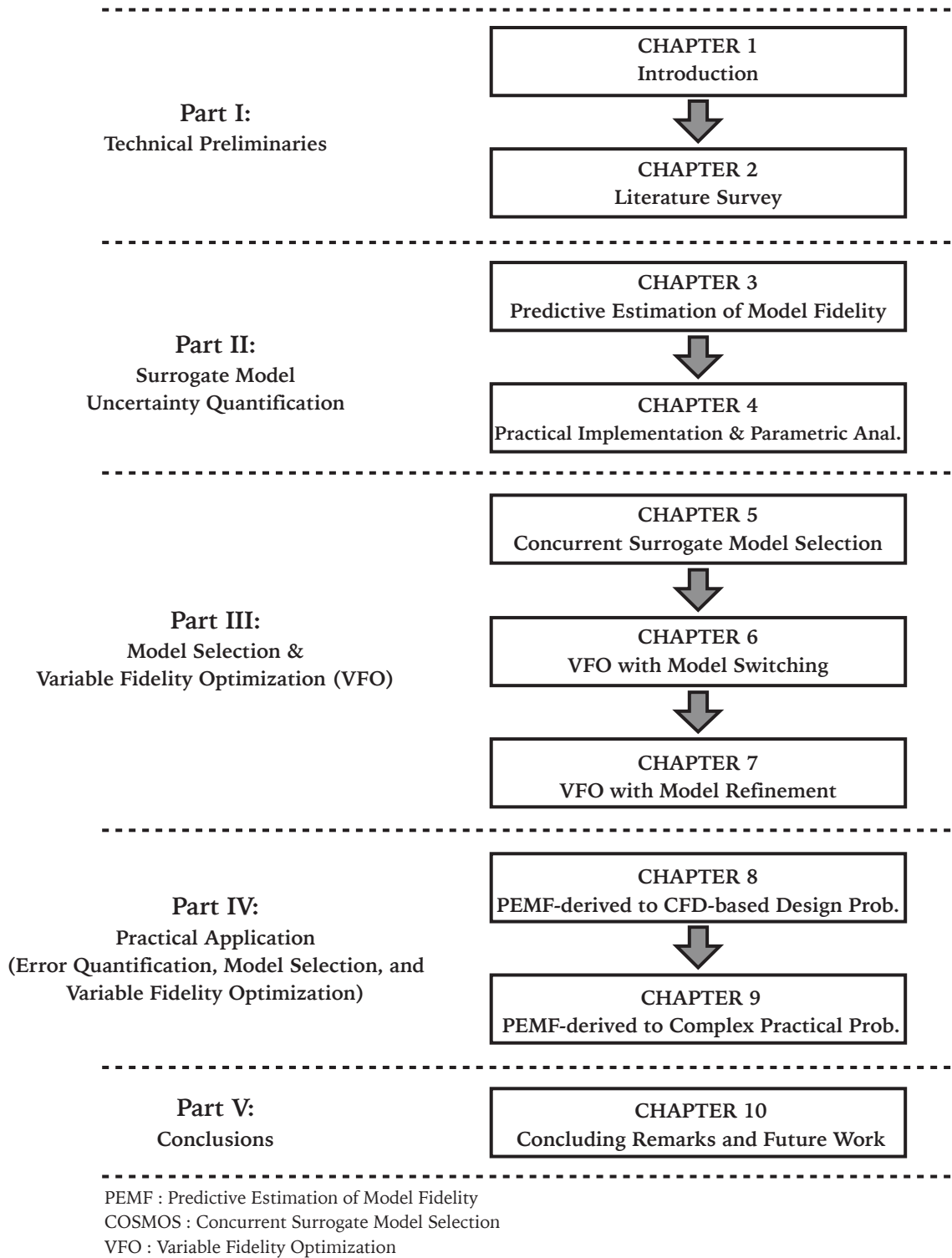


Figure 1.2: Graphical description of dissertation outlines and structure

CHAPTER 2

Literature Survey

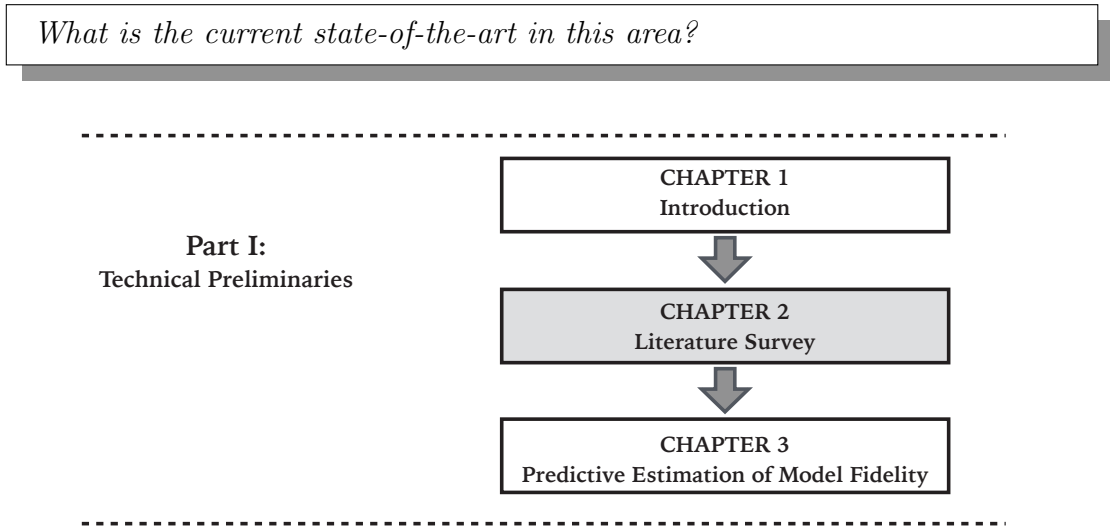


Figure 2.1: Chapter 2 orientation

This chapter reviews the literature in the following key areas of focus: (i) surrogate modeling techniques; (ii) error measures; (iii) model selection; (iv) surrogate-based optimization; and (v) variable fidelity modeling. Key observations from the literature review will be also discussed, which form the foundation of the proposed research.

2.1 Surrogate Modeling

Several reviews of the state-of-the-art surrogate modeling and approximation strategies have been reported in the literature ([19], [20], and [21]). Surrogate modeling is concerned with the construction of purely mathematical models to define relationships between specific system inputs and outputs. Over the past couple of decades, surrogate models are being extensively used in the analysis and optimization of complex systems or in the solution of complex problems. Major surrogate modeling methods include (i) Polynomial Response

Surface Methodology [22], (ii) Kriging [23, 24], (iii) Radial Basis Functions (RBF) [10, 25, 26, 27], (iv) Extended Radial Basis Functions (E-RBF) [28], (v) Artificial Neural Networks (ANN) [11, 29], and (vi) Support Vector Regression (SVR) [29, 30, 31, 32]. Table 2.1 provides a list of standard sampling techniques, and surrogate modeling methods, which extends the preexisting work by Simpson et al. [33].

Table 2.1: Techniques for surrogate modeling

Design of Experiments	Surrogate Modeling
(Fractional) Factorial	Polynomial (linear, quadratic)
Central Composite	Splines (linear, cubic)
Latin Hypercube	Kriging
Hammersley Sequence	Radial Basis Functions (RBF)
Uniform Designs	Extended RBF
Sobol Sequence	Support Vector Regression (SVR)
Random Selection	Artificial Neural Network (ANN)
Box-Behnken	Hybrid Models
Plackett-Burman	
Orthogonal Arrays	

- **Quadratic Response Surfaces (QRS):**

Quadratic Response Surfaces (QRS) is one of the oldest and most popular type of surrogate model. The generic expression for a n -variable QRS is given by:

$$f(x) = C_0 + C_1 X + X^T C_2 X \quad (2.1)$$

where X is the n -dimensional input vector, and C_0 , C_1 , C_2 are respectively the scalar coefficient (constant term), the coefficient vector (of length n) for the linear terms, and the $n \times n$ coefficient matrix for the quadratic terms.

- **Radial Basis Functions (RBF)**

The idea of using Radial Basis Functions (RBF) as interpolating functions was introduced by Hardy [10]. The RBF approximation is a linear combination of basis functions

(Ψ) across all sample points, given by

$$\tilde{f}(x) = W^T \Psi = \sum_{i=1}^{n_p} w_i \psi(\|x - x^i\|) \quad (2.2)$$

where n_p denotes the number of selected sample points; w_i are the weights evaluated using the pseudo inverse method on sample points' responses; ψ is basis function expressed in terms of the Euclidean distance, $r = \|x - x^i\|$, of a point x from a given sample point, x^i . Table 2.2 listed the five different types of basis functions for RBF. In this table, σ represents the shape parameter of the basis function. The shape parameter, $\sigma = 0$, corresponds to a constant basis function [34].

- **Extended Radial Basis Functions (E-RBF)**

The Extended Radial Basis Functions (E-RBF), developed by Mullur and Messac [28], is an important advancement over the standard radial basis function model. The E-RBF surrogate model uses a linear combination of radial and non-radial basis functions. The Non-Radial Basis Functions (N-RBF) are not defined in terms of the Euclidean distance, r . Instead, N-RBFs are defined as functions of individual coordinates of the generic point, x , relative to a given data point, x_i , in each dimension separately. Further details of the E-RBF model, and of the essence of the two hyper-parameters (σ and λ) in the functional form of the E-RBF surrogate model, can be found in the paper by Mullur and Messac [28].

- **Kriging**

Kriging [23] is an approach to approximate irregular data. The kriging approximation function consists of two components: (i) a global trend function, and (ii) a deviation

function representing the departure from the trend function. The trend function is generally a polynomial (e.g., constant, linear, or quadratic). The general form of the kriging surrogate model is given by [24]:

$$\bar{F}(x) = \hat{f}(x, \varphi) + Z(x) \quad (2.3)$$

where $\bar{F}(x)$ is the unknown function of interest, $Z(x)$ is the realization of a stochastic process with the mean equal to zero, and a nonzero covariance, and \hat{f} is the known approximation function

$$\hat{f}(x, \varphi) = f(x)^T \varphi \quad (2.4)$$

where φ is the regression parameter matrix.

The i, j – th element of the covariance matrix, $Z(x)$, is given by

$$COV[Z(x^i), Z(x^j)] = \sigma_z^2 R_{ij} \quad (2.5)$$

where R_{ij} is the correlation function between the i^{th} and the j^{th} data points; and σ_z^2 is the process variance, which scales the spatial correlation function. The popular types of correlation functions are listed in Table 2.2. The correlation function controls the smoothness of the Kriging model estimation, based on the influence of other nearby points on the point of interest. In Kriging, the regression function coefficients, the process variance, and the correlation function parameters, $\{\varphi, \sigma_z^2, \theta\}$, each can be predefined or estimated using parameter estimation methods such as Maximum Likelihood Estimation (MLE). The regression function coefficients and the process variance

can be estimated using MLE, as given by

$$\begin{aligned}\varphi &= (F^t R^{-1} F)^{-1} F^T R^{-1} Y \\ \sigma_z^2 &= \frac{1}{n} (Y - F\tilde{\varphi})^T R^{-1} (Y - F\tilde{\varphi})\end{aligned}\tag{2.6}$$

where $Y = [y_1 \ y_2 \ \dots]$ represents the vector of the actual output at the training points; R is a correlation matrix; and F is a matrix of $f(x)$ evaluated by Kriging at each training point[35].

- **Artificial Neural Networks (ANN)**

A neural network generally contains an input layer, one or more hidden layers, and an output layer. Figure 2.2 shows a typical three layer feedforward neural network. An ANN is developed by defining the following three types of parameters:

1. The interconnection pattern between different layers of neurons;
2. The learning process for updating the weights of the interconnections; and
3. The activation function that converts a neuron’s weighted input to its output activation.

One of the drawbacks associated with neural networks for function approximation is the uncertainty in specifying several of the network parameters, such as the number of neurons, the number of layers, the type of activation function, and the optimization algorithm used to train the network. In addition, the training process generally needs to be supervised in order to avoid “over-fitting” [2, 36].

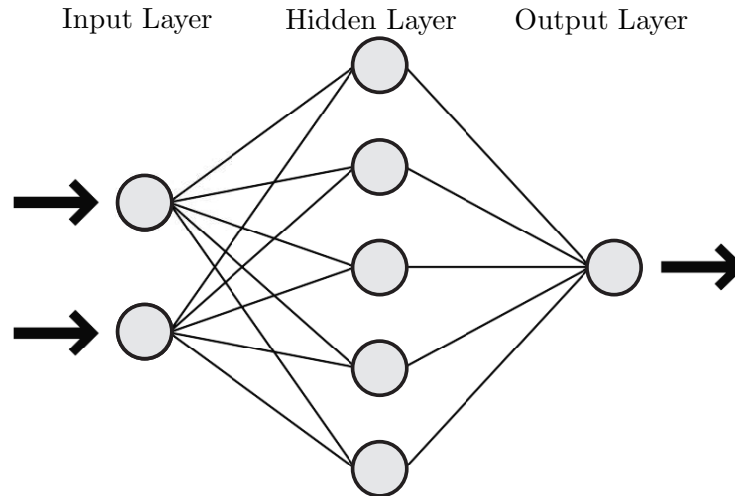


Figure 2.2: A generic topology of neural networks [2]

- **Support Vector Regression (SVR)**

Support Vector Regression (SVR) is a relatively newer regression type surrogate model. For a given training set of instance-label pairs (x_i, y_i) , $i = 1, \dots, n_p$, where $x_i \in R^n$ and $y \in 1, -1^m$, a linear SVR is defined by $f(x) = \langle w, x \rangle + b$, where b is a bias and $\langle \cdot, \cdot \rangle$ denotes the dot product. To train the SVR, the error, $|\xi| = |y - f(x)|$, is minimized by solving the following convex optimization problem:

$$\begin{aligned} & \text{Min } \left\{ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n_p} \xi_i + \tilde{\xi}_i \right\} \\ & \text{subject to} \\ & (w^T x_i + b) - y_i \leq \varepsilon + \xi_i \\ & y_i - (w^T x_i + b) \leq \varepsilon + \tilde{\xi}_i \\ & \xi_i, \tilde{\xi}_i \geq 0, \quad i = 1, 2, \dots, n_p \end{aligned} \tag{2.7}$$

In Eq.2.7, $\varepsilon \geq 0$ represents the difference between the actual and the predicted values;

ξ_i and $\tilde{\xi}_i$ are the slack variables; C represents the flatness of the function; n_p represents the number of training points. By applying kernel functions, $K(\alpha, \beta) = \langle \phi(\alpha), \phi(\beta) \rangle$, under KKT conditions, the original problem is mapped into a higher dimensional space. The dual form of SVR for nonlinear regression could be represented as

$$\begin{aligned} & \text{Max} \left\{ \sum_{i=1}^{n_p} \alpha_i y_i - \varepsilon \sum_{i=1}^{n_p} |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^{n_p} \alpha_i \alpha_j \langle \phi(x_i), \phi(x_j) \rangle \right\} \\ & \text{subject to} \\ & \sum_{i=1}^{n_p} \alpha_i = 0, \quad -C \leq \alpha_i \leq C \quad \text{for } i = 1, \dots, n_p \end{aligned} \tag{2.8}$$

The standard Kernel functions used in SVR are listed in Table 2.2. The performance of SVR depends on its penalty parameter C and kernel parameters γ , r , and d . The parameter $C > 0$ is user-specified and represents the trade-off between flatness and the amount up to which errors larger than ε are tolerated. Figure 2.3 shows a typical cost function, the ε -insensitive loss function. The above formulation is the primal form of the quadratic programming problem. In most cases, the dual form with fewer number of constraints is easier to solve, and is widely used to define the final form of the approximation. It can be shown that the dual form is convex and therefore has a unique minimum. Typical mapping functions allowed are radial basis functions, such as the gaussian function.

2.2 Model Uncertainty: Quantification and Impact

Computational approximation models, both physics-based models(e.g., FEA) and statistical models (e.g., surrogates), are crucial building blocks of most design processes. An informed application of such computational models demands the knowledge of how uncer-

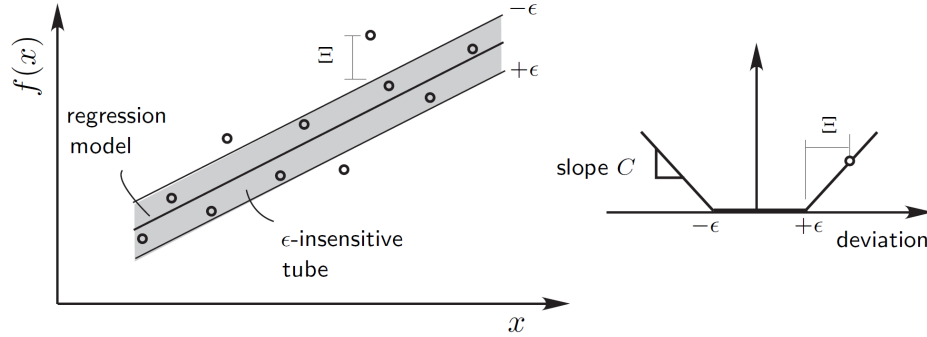


Figure 2.3: ϵ -insensitive loss function [36]

Table 2.2: Basis or Kernel functions and their hyper-parameters in the Surrogate Models

Type of model	Type of basis/correlation/kernel function	Hyper parameter
RBF	Linear:	r
	Cubic:	r^3
	Thin plate spline:	$r^2 \ln(r)$
	Gaussian:	$e^{(-r^2/2\sigma)}$
	Multiquadric:	$(r^2 + \sigma^2)^{1/2}$
Kriging	Linear:	$\max(1 - \theta r, 1)$
	Exponential:	$e^{(-\theta r)}$
	Gaussian:	$e^{(-\theta r^2)}$
	Cubic:	$1 - 0.5\xi + 0.5\xi^2$; $\xi = \max(1 - \theta r, 1)$
	Spherical:	$1 - 3\xi^2 + 2\xi^3$; $\xi = \max(1 - \theta r, 1)$
SVR	Linear:	$(x_i^T x_j)$
	Polynomial:	$(\gamma x_i^T x_j + r)^d$, $\gamma > 0$
	Radial Basis Function:	$e^{-\gamma \ x_i - x_j\ ^2}$, $\gamma > 0$
	Sigmoid:	$\tanh(\gamma x_i^T x_j + r)$
		kernel parameters; γ , r , and d

tainties, both *Aleatory* and *Epistemic*, propagate through the design process. Aleatory uncertainty refers to the inherent variability that exists in physical process (e.g., uncertainty in model inputs and in the parameter of the design). Epistemic uncertainty refers to the uncertainty introduced by approximation models (e.g., uncertainty due to the insufficient data, over-simplification of complex physical phenomena, and model assumptions) [2].

2.2.1 Uncertainty in Physics-based Modeling

A Physics-based model is consider to reflect a real physical systems. However, “a Physics-based model is just a reduced and parsimonious representation of a physical, chemi-

cal, or biological system in a mathematical, abstract, numerical, or experimental form [37]”. Constructing a Physics-based model is the process of idealizing the real system into a relatively simplified form by making a set of assumptions. Mathematical or computational models are generally unlikely to provide an exact representations of the real system due to the assumptions made during the modeling process [38]. Physics-based models also depends on the modeling process (e.g., different element types, shape functions, and mesh sizes can be used in Finite Element Analysis of a system). A prediction error of a Physics-based model is considered as an unknown variable, and called *predictive uncertainty* [38, 39]. Generally, this uncertainty is represented by a probability distribution when statistically sufficient experimental data are presented to estimate the error distribution. Different methodologies have been developed to propagate probability distributions representing predictive uncertainty. The three popular approaches for uncertainty propagation are [2, 38, 40]: (i) sampling methods (i.e., Monte Carlo Simulation (MCS)), (ii) local expansion-based methods (i.e., polynomial approximation using Taylor series), and (iii) the most probable point (i.e., First-order reliability method (FORM)).

- **Sampling Methods:**

In these methods, a set of sample points are generated for the input variables. At each generated sample, the value of the random functions are computed to attain stochastic information about system responses. The implementation procedure of Monte Carlo Simulation (MCS) [41], which is the most popular method in this category, include (i) Selecting the distributions to represent random variables, (ii) Sample generation based on the selected distribution, (iii) Simulation the outputs using sample data, and (iv) Estimating the probabilistic information of the output [38].

- **Local Expansion-based Methods:**

The concept of these methods is to calculate statistical moments of system output based on a small perturbation in input variables. Taylor series expansion method, which is the most popular method in this category, involves expansion of a system output as a series in terms of a small perturbation in input variables and the subsequent solution of the series coefficients [38]. In this approach, the mean and standard deviation of the random function are approximated by first building a polynomial approximation of the function constructed using a Taylor series expansion about a point of interest, followed by determining the moments of this polynomial approximation [2].

- **Most Probable Point Methods:**

The first-order reliability method (FORM) and second-order reliability methods (SORM) [42] in this category are the two popular approximation methods that estimate the probability of failure associated with a limit-state function. The distinction between the two methods depends on the utilization of a first-order or a second-order Taylor series expansion to approximate a limit-state function at the Most Probable Point (MPP) [2, 38, 40].

2.2.2 Uncertainty in Surrogate Modeling

Methods for quantifying the fidelity of or (inversely) the error in statistical models can be broadly classified into: (i) methods that require additional data, and (ii) methods that use existing data [43]. The former can be significantly expensive and is thus not a practical option in a majority of applications. Error quantification methods can also be classified into global and local error estimation methods [44]. The performance of the surrogate over the

entire domain is evaluated by global error measures, while local or point-wise error measures provide the surrogate accuracy in different locations of the design domain. Table. 2.3 provides a list of popular surrogate modeling methods, and the error estimation techniques commonly used for these surrogates.

Table 2.3: Techniques for surrogate model construction and validation

	RMSE	F -test	ANOVA	Split sample	Bootstrapping	Cross-validation	AIC
PRS	✓	✓	✓				✓
Kriging	✓	✓	✓	✓	✓	✓	✓
RBF				✓	✓	✓	
E-RBF				✓	✓	✓	
ANN	✓	✓	✓				
SVR	✓	✓	✓				✓

2.2.2.1 Global Error Measurement

Popular approaches of model independent global error measures include [13]: split sample, cross-validation, and bootstrapping. In a split sample strategy, the sample data is divided into training and test data. The former is used to construct a surrogate; and the latter is used to test the performance of the surrogate.

Limited data sets make determining prediction error of surrogate models more difficult (assuming there is not enough data available to hold out as an independent test sample). In this scenario, the alternative approaches which enable the estimation of surrogate model error using the training data (i.e., without requiring additional system evaluations such as bootstrapping and cross validation) are becoming increasingly important [45, 46, 47]. The idea for cross-validation (which is still the current paradigm) originated in the 1930s [48], and it was developed and refined in the 1970s by Stone [49] [50].

Cross-validation is a popular technique to estimate the error of surrogate without

investing any additional system evaluations. The q -fold cross-validation technique operates through the following five steps [51]:

1. splits the sample points randomly into q (approximately) equal subsets;
2. leaves out each of these subsets in turn (one at a time), and construct an intermediate surrogate using the remaining $q - 1$ subsets;
3. estimates the error of the intermediate surrogate using the omitted subset;
4. aggregate the q sets of errors evaluated to yield a global error measure.

A k -fold cross-validation approach is a variation of q -fold approach, in which all the possible subsets of size k are used to evaluate cross-validation error. In *leave-one-out cross-validation* approach ($k = 1$), at each iteration, the training set is created by taking all sample points except one, and the left out point is used for estimating the error between the surrogate prediction and the actual value. The bootstrapping approach generates m sub-samples from the sample points. Each sub-sample is a combination of all samples with replacement. Different variants of the bootstrapping approach can be used for model identification and determining confidence intervals for surrogates [13].

The mean squared error (MSE) or root mean square error (RMSE) provides a global error measure over the entire design domain. The RMSE evaluated at a set of test points (N_{Test}) is given by:

$$RMSE = \sqrt{\frac{1}{N_{Test}} \sum_{i=1}^{N_{Test}} (y_i - \hat{y}_i)^2} \quad (2.9)$$

where y_i and \hat{y}_i are the actual and the predicted values at the i^{th} test point, respectively. The RMSE thus provides information about the accuracy of the actual surrogate. However RMSE requires additional system evaluations at test points in the case of interpolating surrogates.

The maximum absolute error (MAE) and relative absolute error (RAE) are indicative of local deviations:

$$MAE = \max_{i=1, \dots, N_{Test}} |y_i - \hat{y}_i| \quad (2.10)$$

$$RAE_i = \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (2.11)$$

The prediction sum of square (PRESS) is based on the *leave-one-out cross-validation* error:

$$PRESS = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - \hat{y}_i^{(-i)})^2 \quad (2.12)$$

where \hat{y}_i and $\hat{y}_i^{(-i)}$ are the surrogate estimations at the i^{th} training point respectively predicted by the surrogate constructed using all sample points and the surrogate constructed using all sample points except the i^{th} point [13]. Goel et al. [52, 53] compared different error measures including PRESS and MSE for Kriging, and found that PRESS provided better performance compared to the other methods, as a fidelity prediction and as a surrogate model selection criterion. Acar and Rais-Rohani [54] used the same cross validation error metric to construct weighted surrogates. In 2013, Aute et al. [55] used the cross validation error as a sequential sampling criterion to determine the location of sample point in the design space.

Meckesheimer et al. [43] used the root mean square error of k -fold cross-validation ($RMSE_{CV}$), i.e., root mean square of PRESS ($PRESS_{RMS}$), to measure the global accuracy of the surrogate over the entire design domain:

$$RMSE_{CV} = \sqrt{\frac{1}{k} \sum_{i=1}^k (\hat{y}_i - \hat{y}_i^{(-i)})^2} \quad (2.13)$$

where k is the number of omitted sample points. The variation of k from 1 to 10 was studied;

for each k value, the average of the error measured on all possible combinations of k points were used to provide a measure of the global accuracy of the surrogate. They also compared this error measure with the actual error estimated on additional test points to show the practicality of the RMSE_{CV} as a fidelity characterizing method without using additional system evaluations.

Viana et al. [51] applied the RMSE_{CV} approach as a criterion in surrogate model selection and in constructing a weighted average surrogate. They also showed that better results can be achieved using the *leave-one-out* approach. Later, Viana et al. [56] used RMSE_{CV} to estimate the safety margin for a conservative surrogate model. Zhang et al. [57] applied relative absolute error of cross-validation (RAE_{CV}) for characterizing the uncertainty in surrogate models, where the normalized RAE_{CV} at the i^{th} training point is defined based on the *leave-one-out* approach. This characterization was used in conjunction with support vector machines to segregate the design space into sub-spaces based on the level of model errors.

2.2.2.2 Local Error Measurement

Standard local error measures include: (i) the mean squared errors for Kriging [17], and (ii) the linear reference model (LRM) [58]. In stochastic surrogate models like Kriging, the errors at two different points of the design domain are not independent; and the correlation between the points is related to the distance between them. When the distance between the two points is small, the correlation tends to one, and when the distance is large, the correlation tends to zero. According to this correlation strategy, if the point x^* is close to sample points, the prediction confidence at that point is higher than that when the point is far away from all the sample points. This concept is reflected in the local error estimation

method for Kriging predictor at the special point x^* . This error is equal to zero at the sample points and is equal to the approximation error variance in the stochastic process. The LRM is a model independent method for quantifying the local performance of a surrogate. The LRM considers the region with oscillations as a high-error location. This method categorizes errors of a surrogate in the design domain based on the deviation of the surrogate from the local linear interpolation [58].

2.2.3 Observations and Research Needs

From the literature survey of surrogate model error estimation methods provided above, we identify that the state-of-the-art surrogate model error measures are subject to one or more of the following limitations:

- They are model-dependent (and hence may not be suitable for global model comparison, and effective model selection);
- They require additional system evaluations (and hence are expensive);
- They quantify errors in intermediate surrogate models (e.g., PRESS), which are often not representative of the errors in the actual surrogate model that will be used for design and analysis; and
- They generally do not provide an understanding of the uncertainty introduced by a surrogate in a design process (i.e., provides deterministic error measures).

It is also evident that there is a need for the development of new error measures, or the modification of existing approaches to extract the maximum pertinent information from the limited resources generally available to construct surrogate model. It is also desirable that such error quantification approaches have the following characteristics:

- i Be independent of the type of model (e.g., Radial basis function, Polynomial response surface, Gaussian processes),
- ii Provide a predictive estimate of the error of the actual model, and
- iii Provide direct opportunity for model selection and model refinement

2.3 Model Selection

2.3.1 Existing Model Selection Method

The selection of a suitable surrogate model for a given experimental or simulated data set is critical to the intended use of surrogates. While a suitable surrogate model can be selected intuitively (experience-based selection), based on an understanding of the data characteristics and/or the application constraints [59, 60, 61], the development of general guidelines might not be practical due to the diversity of system behavior among design applications. In the context of practical application, there are several common pitfalls to intuitive model selection. First, only a few candidate surrogates, which the user is well acquainted, are generally considered, thereby often failing to exploit the large and fast-growing pool of available surrogates. More importantly, in designing complex multidisciplinary systems, it's highly unlikely that single users will have significant experience in multiple disciplines to enable them to make a reliable, insightful choice of models, that will represent complex system behavior. Automated model selection approaches, based on the quantitative decision-making techniques, could fill this crucial gap in the surrogate modeling paradigm.

In the literature, error measures have been used to select the model type and basis functions for a given sequence of data [25, 62, 63, 64]. Consequently, the best surrogate

model, \bar{F}^* , among a set of candidates, $\bar{F} \in \Gamma$, is defined as the one fulfilling

$$\bar{F}^* = \arg \min_{\bar{F} \in \Gamma} \varepsilon(\bar{F}) \quad (2.14)$$

where Γ and $\varepsilon(\bar{F})$ respectively represent the set of candidate surrogates and the error in the predictions of surrogate \bar{F} . Popular error measures used for model selection include [13]: (i) split sample, (ii) cross-validation [65], (iii) bootstrapping, (iv) Akaike's information criterion (AIC) [66, 67], and (v) Schwarz's Bayesian information criterion (BIC) [67]. A definition of i-iv methods are provided in Section 2.2. In Akaike's information, the performance of the surrogate is relatively estimated based on a penalized likelihood. AIC is equal to the sum of a negative log likelihood and a penalty term, as given by

$$AIC = -2 \log L(\hat{\theta}) + 2k \quad (2.15)$$

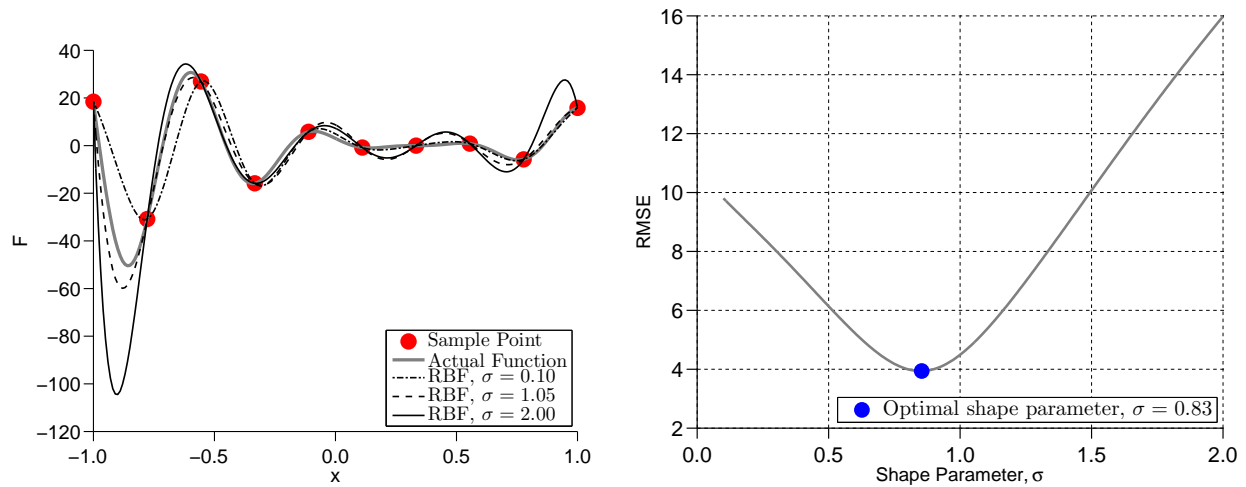
In this equation, $L(\hat{\theta})$ is the maximized likelihood function, and k is the number of free parameters in the model, which is a measure of complexity or the compensation for the bias in the lack of fit when the maximum likelihood estimators are used [66]. Bayesian information criterion (BIC) is just a variation of AIC in which the penalty term is increased, as given by

$$BIC = -2 \log L(\hat{\theta}) + k \ln n \quad (2.16)$$

The selection of the model type and kernel function without considering the effect of hyper-parameter(s) could lead to suboptimal selections, a concern that often remains unaddressed in error-based model selection. To further elucidate this issue, we will consider

the example of the Multiquadric radial basis function, $\psi(\|x - x^i\|, \sigma) = (\|x - x^i\|^2 + \sigma^2)^{1/2}$, where the function $\bar{f}(x) = \sum_{i=1}^{n_p} w_i \psi(\|x - x^i\|)$ is a universal RBF approximation if the shape parameter, σ , is allowed to vary and the summation runs over all the training points.

Figure 2.4(a) shows the different functional forms predicted by the RBF Multiquadric surrogate model for the one-dimensional test problem, when different values of σ are prescribed. This figure clearly demonstrates the strong sensitivity of the predicted function to the hyper-parameter. In this case, we see that $\sigma = 1.05$ approximately leads to the RBF models that resembles the actual function most closely. Further evidence towards this observation is provided by Fig. 2.4(b), which shows the variation of the model error (predicted on additional test points) with the hyper-parameter, σ .



(a) RBF predictions for different shape parameter (σ) (b) Variation of Root Mean Square Error (on 100 test points) by changing a RBF shape parameter, σ

Figure 2.4: The impact of shape hyper parameter (σ) in a RBF surrogate with Multiquadric basis function constructed for a 1-D problem; $f(x) = (6x - 2)^2 \sin[2(6x - 2)]$, $x \in [-1 \ 1]$

To mitigate the possibility of constructing a suboptimal version of a surrogate model with a selected Kernel (or basis) function, one can perform hyper-parameter optimization. Hyper-parameter optimization generally involves minimizing the estimated model errors with

respect to the hyper-parameters [68, 69]. Examples of hyper-parameter(s) optimization for a given model and Kernel function exist in the literature. Martin and Simpson [35] used maximum likelihood estimation (MLE) and cross-validation methods to find the optimum hyper-parameter value for the Gaussian correlation function in Kriging. The likelihood function in that case defines the probability of observing the training data for a particular set of parameters. Mongillo [70] used maximum likelihood estimation (MLE) and leave-one-out cross-validation methods to select an optimal shape parameter in a Gaussian RBF. These efforts however did not consider model type selection. Gorissen et al. [60] provided the leave-one-out cross-validation and AIC error measures in the SURrogate MOdeling (SUMO) Toolbox to select the best model type and hyper parameter value(s) for a given problem using a genetic algorithm. These important contributions further demonstrate the need for automated model selection towards realizing the full potential of surrogate models/meta-models in complex system design and analysis. Unified approaches to perform automated surrogate model selection simultaneously at all three levels (model type, kernel type, and hyper-parameter value) is however rare in the literature.

2.3.2 Observations and Research Needs

Existing automated model selection techniques generally operate only at one of the following levels:

- (i) Selecting the **model type** (e.g., Quadratic Response Surfaces (QRS) [25], Kriging [8, 9], Moving Least Square [71, 72], Radial Basis Functions (RBF) [10], Support Vector Regression(SVR) [73], and Neural Networks [11]),
- (ii) Selecting the **basis function, kernel function, or correlation function** (e.g., Linear, Gaussian, and Multiquadric functions), and

(iii) Optimizing the **hyper-parameter(s)** of a selected function. Hyper-parameters are control parameters (e.g., shape parameter in RBF), that are more often than not prescribed by users and not estimated during the standard training process.

As such, it is unfortunate that unified approaches to performing automated surrogate model selection simultaneously at all three levels (model type, kernel type, and hyper-parameter value) has remained a largely unexplored research topic. The capability of implementing selection at three levels, as illustrated in Fig. 2.5, avoids limiting assumptions on the choice of constituent functions and hyper-parameter values.

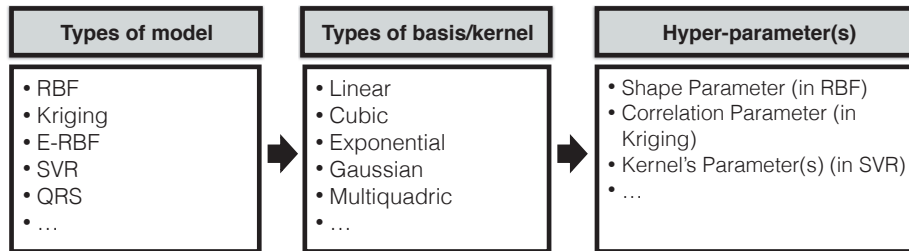


Figure 2.5: Elements of 3-level Surrogate Model Selection

In the absence of a three-level model selection approach, we will usually fall short of exploiting the full potential of the surrogate models at our disposal. A systematic surrogate model selection framework that coherently operates at all the three levels listed above is thus needed.

An effective model selection framework should offer these important features:

- (i) Allows simultaneous selection of model type, kernel type, and optimized hyper-parameter values
- (ii) Uses an error metric (as selection criteria) that is model independent and reliable (i.e., accurate and less sensitive to the DOE)

- (iii) Can search a comprehensive candidate pool of diverse model types and Kernel types within a reasonable time frame.

2.4 Variable Fidelity Optimization

Variable fidelity models refers to models with different levels of fidelity, where the computational cost of the model is generally related to the accuracy of the model estimation. In addition to low, medium, and high fidelity physics-based models, *surrogate models* (or mathematical approximation models) can also be used as candidates within a set of variable fidelity models.

Besides direct implementation of a surrogate model as a black-box function (directly substituting a high fidelity model or data), low fidelity physic-based models can also be combined with a surrogate model to achieve a hybrid model of greater accuracy than its individual components (as illustrated in Fig. 2.6). Low fidelity physic-based models (e.g., the vortex lattice computational fluid dynamics method) are generally less complex than a high fidelity model and often provides a less faithful representation of the physics of the system[74]. In many cases, these models can be obtained by simplifying either the analysis model (e.g., using coarse finite element mesh) or the original physical formulation (e.g., using simplified boundary conditions or geometry). To their advantage, low fidelity physics-based models often inherit the major features of true models, while being significantly less expensive. Hence, these models could provide a reliable foundation for the construction of high-quality hybrid approximation models. These hybrid models, also called *tuned low fidelity models*, are expected to reflect the most prominent physical features of the system, and at the same time preserve computational efficiency. Two well-known approaches for constructing tuned low fidelity (TLF) models are *multiplicative* and *additive* approaches, as

given in Eqs. 2.17 and 2.18, respectively [75].

$$\text{Multiplicative approach : } y_{TLF} = A \times y_{LF} \quad (2.17)$$

$$\text{Additive approach : } y_{TLF} = B + y_{LF} \quad (2.18)$$

In both of these approaches, the tuning functions (A and B) are trained using the associated values of the high and low fidelity model for a given DoE, as shown below:

$$A(X) = \frac{y_{HF}(X)}{y_{LF}(X)}$$

$$B(X) = y_{HF}(X) - y_{LF}(X) \quad (2.19)$$

where $X = \{X_1, X_2, X_3, \dots, X_{N_s}\}$

N_s : Number of sample points

and where $y_{HF}(\cdot)$ and $y_{LF}(\cdot)$ respectively represent the functional responses of the low and the high fidelity models. In surrogate-based tuned low fidelity models, the tuning (or correction) of a low fidelity model are performed using surrogate model constructed through a DoE of the high fidelity model[76, 77, 78].

2.4.1 Model Switching & Model Management

The major pitfall in using low fidelity models in optimization is that they can often mislead the search process, leading to suboptimal or infeasible solutions. To address this issue and provide optimum designs with high fidelity system evaluations, model switching or model management strategies can be applied. Different model management strategies

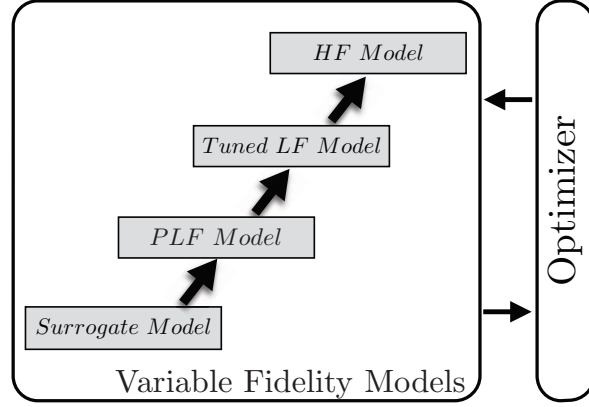


Figure 2.6: Variable Fidelity Models

have been reported in the literature, for integrating low fidelity models within optimization algorithms. One class of model management strategies are developed based on the *Trust-Region* methods [79, 80, 81, 82, 83]. The basic idea of the Trust-region is to solve an optimization problem, $\text{Min}_{x \in \mathbb{R}^P} f(x)$, using the high fidelity model ($f(x)$) where x^0 is the initial guess; in solving this optimization problem using a gradient-based algorithm, the k^{th} iteration is computed as $x^{k+1} = x^k + \lambda \Delta x$, where λ is the step length and Δx is the decent direction. As Δx is fixed, the problem reduces to a one-dimensional optimization problem: $\text{Min}_{\lambda} f(x^k + \lambda \Delta x)$. To improve the computational efficiency of the problem, the low fidelity model, $\hat{f}(x)$, can be used in the latter optimization problem. Assuming the low fidelity model is only valid in the vicinity of x^k (e.g., $x^k + \gamma$), the optimization search for λ is changed to the following constrained optimization problem:

$$\text{Min}_{\lambda} f(x + \lambda \Delta x), \quad \text{subject to: } \|\lambda \Delta x\| < \gamma \quad (2.20)$$

where γ is the *trust-region radius*. In the *Trust-Region* based model management methods developed by Alexandrov et al. [84] and by Toropove and Alvarez in 1998 [85], the parameter γ is adaptively increased (or decreased) depending on how well the low fidelity model, $\hat{f}(x)$,

predicts the improvement in the high fidelity model. This criterion is estimated by computing the ratio of the actual to the predicted improvement in the objective function, as given by

$$\frac{f(x^k) - f(x^k + \lambda^k \Delta x^k)}{\hat{f}(x^k) - \hat{f}(x^k + \lambda^k \Delta x^k)} \quad (2.21)$$

The *Trust-Region* method seeks the agreement of the function and its gradient values in the low fidelity model with those estimated in the high fidelity model. However, these techniques may not be directly applicable in problems where gradients are expensive to evaluate, or where zero-order algorithms are being used for optimization.

The model management approaches used in heuristic optimization algorithms can be broadly classified into two different approaches which are (i) *individual-based evolution control*, and (ii) *generation-based evolution control*[86]. In the individual-based approach, selected individuals (*controlled individuals*) within a generation are evaluated using a high fidelity model. In the generation-based approach, the whole population at a certain generation (*controlled generation*) is evaluated using the high fidelity model. Graning et al. [87] explored different individual-based evolution frameworks such as (i) the Best Strategy[88], where the best individuals at each generation are selected as controlled individuals, (ii) the Pre-Selection method [89], where the offspring of the best individuals are selected as controlled individuals, and (iii) the Clustering Technique [90], where the *k-means* clustering technique is used to find the “controlled individual cluster” based on the distance from the best individual.

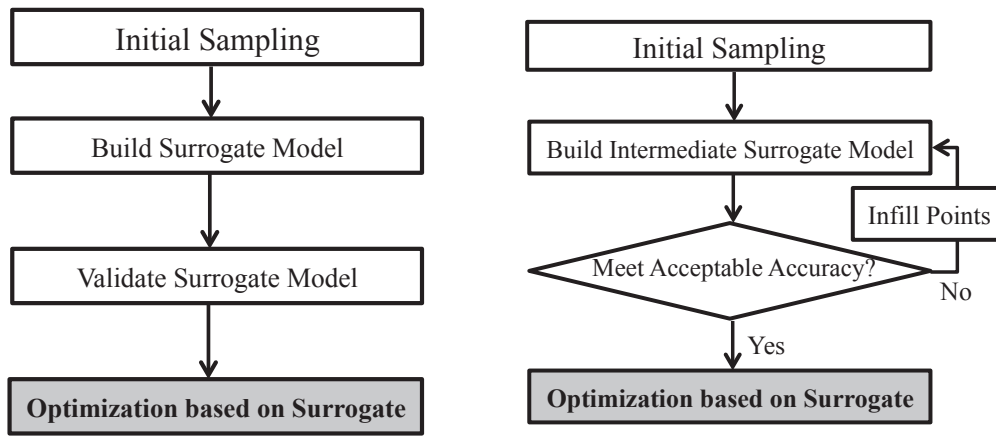
2.4.2 Model Refinement & Surrogate-based Optimization

“Model Refinement” approaches offer a powerful solution to this challenge by providing the surrogate models with reasonable accuracy during the optimization process. In the typical SBO approach, the optimization process is repeated multiple times, where each time the surrogate model is improved by the addition of new sample points.

Another important area of application of surrogate model error measures is “model refinement”. In these approach, that is developed for non-physics-based low fidelity models (e.g., surrogate model), the accuracy and robustness of the surrogate model is improved by adding points where additional evaluations of the high fidelity model / physical experiment are desired to be performed.

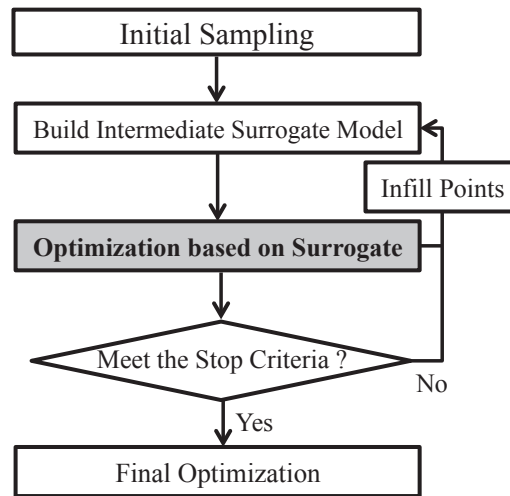
In the literature, there are three approaches for generating sample points, which are (i) Single stage sampling, (ii) Traditional sequential sampling, (iii) Adaptive sampling. These approaches are illustrated in Fig. 2.7. In the *Single stage* (or space filling) method, all the sample points are generated in one stage (Fig. 2.7(a)). This approach attempts to assign the locations of all sample points over the entire design space in one step. In the *Sequential sampling* approach, sample points are generated in an iterative process (Fig. 2.7(b)). In this approach, the intermediate surrogate model is fitted to the response of a portion of the sample points and is updated at each iteration by including the response of new infill points (for improving the surrogate accuracy). In one class of sequential sampling method, the location information of the current global optimum at each iteration is also used for adding infill points and updating the surrogate model. In the literature, the word *sequential* is sometimes referred to as *adaptive* or *application driven*, and the new sample points are known as *infill points* or *update points*.

The ‘Model Refinement’ strategies used to improve the accuracy of the surrogate models



(a) Single stage sampling

(b) Sequential sampling



(c) Surrogate-based optimization

Figure 2.7: Sampling strategies

during the optimization process are known as “*Surrogate-Based Optimization*” strategies (Fig. 2.7(c)). Surrogate-Based Optimization (SBO) generally refers to the concept of reduction of analysis and optimization time by adopting surrogates for the objective and constraint functions. SBO has been shown to be effective and robust for the design of computationally expensive system functions. SBO has been widely used in single or mul-

tidisciplinary optimizations with various applications (aerospace design, automotive design, electronic packaging, structure optimization, decision making) [20]. An important objective in SBO is to achieve an acceptable level of accuracy while attempting to minimize the overall computational effort.

In SBO approaches, Infill points are generally added in (i) the region where the optimum is located (local exploitation); and/or (ii) the entire design space to improve the global accuracy of the surrogate (global exploration)[16, 78, 91]. Infill points can be added in a fully sequential (one-at-a-time), or can be added in a batch sequential manner. There exist various criteria for determining the locations of the infill points including (i) Index-based criteria (e.g., (Integrated- and maximum) Mean Square Error (MSE) and Maximum Entropy criteria) and (ii) Distance-based criteria (e.g., Euclidean distance, Mahalonobis distance, and Weighted distance criteria) [17, 92, 93, 94, 95, 96]. Jin et al. [97] reviewed different one-at-a-time sequential sampling criteria, and illustrated their potential benefits over single stage methods [98, 99]. Loepky et al. [100] explored different batch and one-at-a-time sequential criteria for the Gaussian process model. They pointed out that these criteria perform better under the batch sequential approach, and one-at-a-time augmentation would be likely impractical due to the higher computational cost (especially in the case of physical experiments).

Over the last two decades, different Surrogate-Based Optimization strategies have been developed [17, 101, 102, 103]. Trosset and Torczon in 1997 [104] proposed an approach where the balance between exploitation and exploration was considered using the aggregate *merit function*, $\hat{f}(x) - \rho d_{min}(x)$, where, $d_{min}(x) = \text{Min}_x \|x - x^i\|$, $\rho > 0$. It is important to note that, this technique is independent of the type of surrogate modeling technique being considered. Another approach which is independent of the type pf surrogate modeling, is

Adaptive Sequential Sampling (ASS) developed by Mehmami et. al. [18]. Adaptive Sequential Sampling (ASS) is a recently developed adaptive approach to add infill points during SBO. In this approach, both local exploitation and global exploration aspects are considered for updating the surrogate during optimization, where multiple iterations of the SBO process is performed to increase the quality of the optimal solution. This approach adaptively improves the accuracy of the surrogate in the region of the current global optimum as well as in the regions of higher relative errors. Based on the initial sample points and the fitted surrogate, the ASS method adds infill points at each iteration in the locations of: (i) the current optimum found based on the fitted surrogate; and (ii) the points generated using cross-over between sample points that have relatively higher cross-validation errors.

Jones et al. in 1998 [17] developed a well-known model management strategy that is based on an *Expected Improvement (EI)* criterion, and is called *Efficient Global Optimization (EGO)*. This powerful approach is however generally limited to surrogate models based on Gaussian processes. Assuming f_{min} is the objective function value of the optimum in the training data, the expected improvement in an infill point x is given by $\mathbb{E}(I(x)) = \mathbb{E}(\max(f_{min} - F(x), 0))$. In this case, $F(x)$ is a Gaussian distribution, $F(x) \sim \mathcal{N}(\hat{f}(x), \sigma^2(x))$, where the posterior mean, $\hat{f}(x)$, is used as a surrogate model, and the posterior variance $\sigma^2(x)$ gives an estimate of the uncertainty involved in the surrogate prediction. The expected improvement can be estimated by

$$\mathbb{E}(I(x)) = (f_{min} - \hat{f}(x))\Phi\left(\frac{f_{min} - \hat{f}(x)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{f_{min} - \hat{f}(x)}{\sigma(x)}\right) \quad (2.22)$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ denote the standard normal density and distribution functions, respectively [101]. Subsequently, an infill point can be found by maximizing the expected

improvement, $X^{infill} = \arg \max_x (\mathbb{E}(I(x)))$.

In the literature, we can find a wide spectrum of optimization algorithms used for SBO: gradient-based methods, evolutionary methods, and swarm-based methods, which are listed in Table 2.4.

Table 2.4: Optimization algorithms

Gradient-based methods	Population-based methods	Other methods
1) Adjoint equation	1) Genetic algorithm	1) Random search
2) Newton’s method	2) Memetic algorithm	2) Grid search
3) Steepest descent	3) Swarm based optimization	3) Simulated annealing
4) Conjugate gradient	4) Harmony search	4) Direct search
5) Sequential quadratic programming	5) Ant colony optimization	5) Indirect Optimization based on Self-Organization
	6) Bees algorithm	

2.4.3 Observations and Research Needs

A survey of existing model switching and model refinement approaches for integrating models with different levels of fidelity into an optimization process is provided in this section. Several of the existing model refinement strategies are found to be defined for specific types of low fidelity model, e.g., EGO works primarily for Gaussian process-based surrogate models. On the other hand, most of existing model switching techniques generally consider the combination of only two models of different fidelities (e.g., Trust-region methods, and individual- and generation-based techniques). In addition, some of them need to estimate the gradient of low and high-fidelity models (e.g., Trust-region method which seeks the agreement of the function and its gradient values in the low-fidelity model with those estimated in the high-fidelity model.). It is evident from the above discussion that there is appreciable scope (and need) for advancing the state of the art in Variable Fidelity Optimization. Specifically, the development of a model switching strategy that can be applied to different types of low fidelity models (i.e., physics-based and non-physics-based low fidelity models),

and allows adaptive switching between more than two models are need to be pursued. There is also significant interest in developing model-independent Model Refinement approach that be capable of refining different surrogate models during optimization process using heuristic optimization algorithms.

PART II

Surrogate Model Uncertainty

Quantification

CHAPTER 3

Predictive Estimation of Model Fidelity (PEMF)

How do we develop a reliable surrogate model error measure, which extracts maximum pertinent information from the limited available resources?

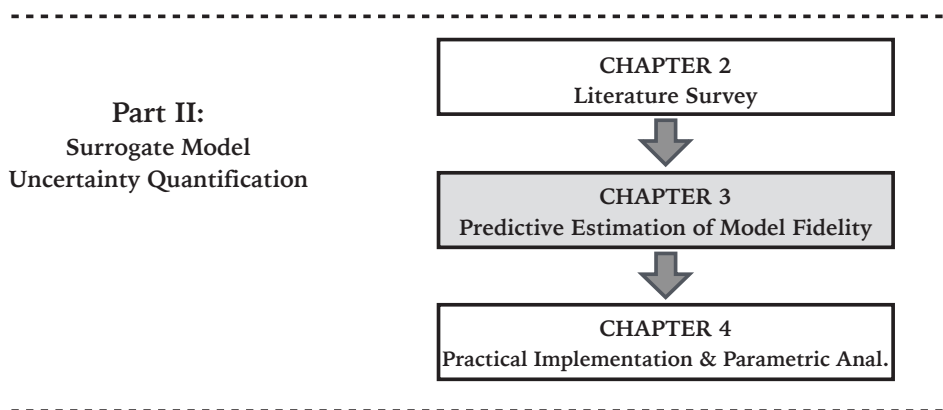


Figure 3.1: Chapter 3 orientation

In this chapter, we present a new model-independent approach for quantifying the fidelity of surrogate models, called “Predictive Estimation of Model Fidelity (PEMF)”. The PEMF method advances the current paradigm in providing reliable information regarding the fidelity of surrogate models, which improves the flexibility of model selection and the confidence in surrogate-based optimization. The PEMF method is derived from the hypothesis that “the accuracy of an approximation model is related to the resources leveraged to train

Portions of this chapter previously appeared as:

Mehmani, A., Chowdhury, S., and Messac, A., “Predictive Quantification of Surrogate Model Fidelity based on Modal Variations with Sample Density,” *Structural and Multidisciplinary Optimization*, 2015,

Mehmani, A., Chowdhury, S., Zhang, J., and Messac, A., “Quantifying Regional Error in Surrogates by Modeling its Relationship with Sample Density,” *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA, Boston, Massachusetts, April 2013, and

Mehmani, A., Chowdhury, S., Zhang, J., and Messac, A., “Regional Error Estimation of Surrogates (REES),” *14th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Indianapolis, Indiana, September 2012.

the model".

The remainder of the chapter is organized as follows: Section 3.1 presents the overview of this research; the formulation of the proposed model fidelity quantification approach is described in Section 3.2; numerical experiments and results are respectively presented in Sections 3.4 and 3.5, respectively. The Chapter summary is provided in Section 3.6.

3.1 Chapter Introduction

The method developed in this chapter, called Predictive Estimation of Model Fidelity (PEMF), is designed to address two primary objectives. The first objective is to develop a model-independent methodology for quantifying the fidelity of the *‘actual surrogate’ without requiring additional system evaluations*. In this context, the ‘actual surrogate’ refers to the surrogate model constructed using all available training points (and not any subset of training points, as used in cross-validation). With this approach, we provide error information for the actual model that is to be used for function analysis or optimization (assuming surrogate construction is not an end in itself). The second objective is to track the variation of the surrogate model error with an increasing density of training points, thereby creating opportunities for global model selection and model refinement.

3.2 Predictive Estimation of Model Fidelity (PEMF)

The PEMF method determines the model fidelity within the design domain (defined by the user) by analyzing the variation in the model error distribution with increasing number of training points. This measure of fidelity can provide uniquely useful information about the accuracy of the surrogate model in regions of interest or the entire design domain, thereby significantly improving the usability of the surrogate in model-based design applications. In the context of surrogate-based design, this information can be directly applied for (i)

surrogate model refinement [97, 105, 106], (ii) surrogate model selection [35, 60, 107], and (iii) uncertainty analysis [108, 109]. In these applications, PEMF is either enabling more informed use of the constructed surrogate model or the platform to identify and use the surrogate that is most suited to the concerned design process. A flowchart describing the PEMF algorithm is illustrated in Fig. 3.2, and the major steps are described below:

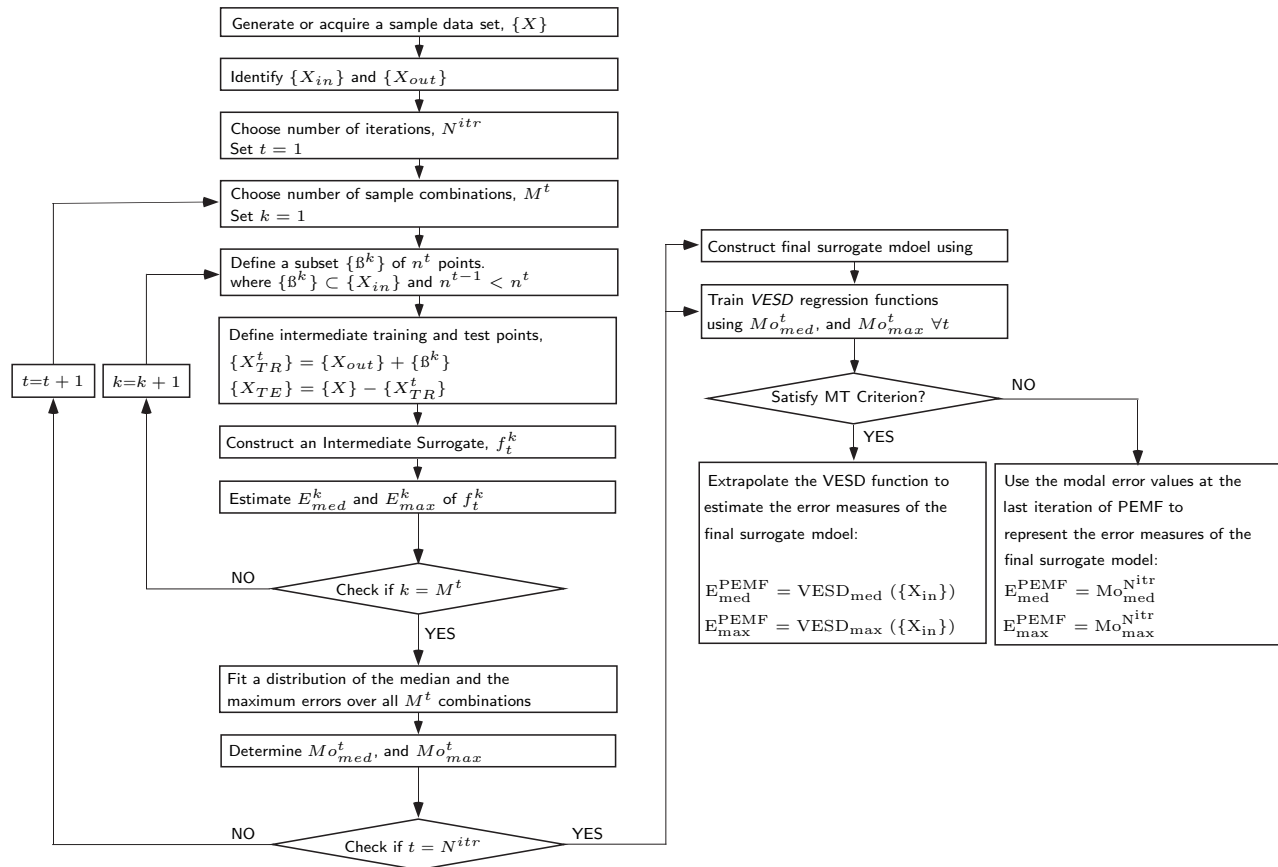


Figure 3.2: The Predictive Estimation of Model Fidelity (PEMF) algorithm

3.2.1 Generating sample data

In this step, a set of experimental designs are generated based on a user-specified distribution or given by a DOE. The expensive system evaluation (simulation or physical experiments) is then performed over the sample data points. The entire set of sample points is represented by $\{X\}$. This sample input-output data might also be given by other measured

data sources.

3.2.2 Defining the region of interest

The entire set of sample points is divided into inside and outside sets based on the boundaries of the user-defined region of interest, and are represented by $\{X_{in}\}$ and $\{X_{out}\}$, respectively.

3.2.3 Quantifying the variation of the error distribution with sample density

This step is an iterative process, where the number of iterations (N^{itr}) is to be guided by the dimension of the problem, the number of sample points in the inside set, and the observed stability of error variation. In this first development and implementation of PEMF, the number of iterations are fixed (prescribed by the user). At each iteration, sample points are divided into a set of intermediate training data $\{X_{TR}\}$ and a set of intermediate test data $\{X_{TE}\}$, i.e.,

$$\begin{aligned} \{X_{TR}^t\} &= \{X_{out}\} + \{\beta^k\} \\ \{X_{TE}^t\} &= \{X\} - \{X_{TR}^t\} \end{aligned} \tag{3.1}$$

where, $\{\beta_1^t\} \subset \{X_{in}\}$

$$t = 1, 2, 3, \dots, N^{itr}$$

where $\{\beta^k\}$ represents a k^{th} subset of inside-region sample points. At each iteration t , the size of $\{\beta^k\}$ is defined by n^t , where $n^{t+1} > n^t$ and $n^1 \geq 1$.

At each iteration, the total number of possible sample combinations is defined by M^t where $M^t \leq \binom{N_i}{n^t}$; the term N_i represents the number of inside-region sample points. In low dimensional problems, all possible subsets of size n^t could be used, while in high

dimensional problems, a fraction of subsets could be used to avoid intractable computational cost. Intermediate surrogates f_t^k , are constructed at the t^{th} iteration for all combinations, $k = 1, 2, \dots, M^t$, using the intermediate training points. These surrogate models are then evaluated over the intermediate test points.

The median and the maximum errors for each combination (E_{med}^k and E_{max}^k) are then estimated using the RAE values on the intermediate test points (e_1, e_2, \dots, e_{m^t}). Probability distribution functions are used to represent the distribution of median and maximum errors over the M^t different combinations at each iteration, thereby providing an understanding of the uncertainty directly associated with the surrogate model. A chi-square (χ^2) goodness-of-fit criterion [110] is used to select the most suitable type of distribution from a list of candidates such as lognormal, Gamma, Weibull, logistic, log logistic, t location scale, inverse Gaussian, and generalized extreme value distribution. The χ^2 criterion is defined based on the error between the observed and the assumed PDF of the distribution. Assuming observed frequencies of m intervals of the random variable are represented by o_1, o_2, \dots, o_m ; the corresponding theoretical frequencies are represented by t_1, t_2, \dots, t_m ; and the χ^2 criterion is given by

$$\chi^2 = \sum_{i=1}^m \frac{(o_i - t_i)^2}{t_i} \quad (3.2)$$

The mode of the median and the maximum error distributions at each iteration (Mo_{med}^t and Mo_{max}^t) are evaluated to provide a measure of central tendency, and are used to relate the variation of the surrogate error with the sample density.

3.2.4 Predicting the fidelity of the final surrogate model

The final surrogate model is constructed using the entire set of training data. Regression models are applied to represent the statistical mode of the median error distribution (Mo_{med}) and that of the maximum error distribution (Mo_{max}) at each iteration as monotonic functions of the number of inside-region training points, n^t . These regression functions are called the *variation of error with sample density* (VESD), and are expressed as

$$\begin{aligned} {}^t E_{med}^{Mo} &= F_{med}(n^t) \\ {}^t E_{max}^{Mo} &= F_{max}(n^t) \end{aligned} \tag{3.3}$$

The VESD regression functions are then used to predict the fidelity of the final surrogate under the condition that a Monotonic Trend (MT) Criterion is satisfied. The MT criterion is defined to statistically test the feasibility of a monotonic decrease of the model fidelity with increasing sample density. If the MT criterion is not satisfied, a stable implementation of the k -fold cross-validation (PEMF-based k -fold) is used instead of the VESD functions to represent the surrogate model fidelity. The formulation and application of the MT criterion is described in the following subsection.

3.2.4.1 Constructing and Testing Functions to Represent the Variation of Error with Sample Density (VESD)

The selection of the type of regression function is critical to quantify the variation of error with training point density. In this chapter, two types of regression functions are used to represent the variation of maximum and median error as functions of the number of inside-region training points. These functions are

Type 1 Exponential regression function

$$F(n^t) = a_0 e^{-a_1 (n^t)} \quad (3.4)$$

Type 2 Multiplicative regression function

$$F(n^t) = a_0 (n^t)^{-a_1} \quad (3.5)$$

where a_0 and a_1 are regression coefficients known as initial value and rate of decay, respectively. These coefficients are determined using the least square method such that $a_0, a_1 > 0$, and $n^t \geq 1$. The smaller the rate of decay (a_1), the lower the sensitivity of surrogate model accuracy to sample density. Numerical experiments exploring linear, polynomial, multiplicative, exponential, and other standard regression functions indicated that exponential and multiplicative functions are the most suitable choice in this context. In this chapter, the root mean squared error metric is used to select the best-fit regression model. The choice of these regression functions assume *a smooth monotonic decrease of the model error with increasing density or number of training points within the region of interest.*

Intuitively, most surrogate modeling scenarios are expected to conform to the monotonically decreasing trend of the model error (with sample density). However, in certain cases, the variation of the estimated error (Mo_{med}^t and Mo_{max}^t ; $t = 1, 2, \dots, N^{\text{itr}}$) with sample density (over iterations) may not follow the monotonically decreasing trend. Such cases may arise due to a highly non-uniform distribution of sample points or a highly skewed distribution of nonlinearity of the actual output function over the input space. To consider the possibility of such non-conforming cases, and avoid unreasonable inaccuracies in error predictions using

the fitted VESD function, the feasibility of the monodically decreasing trend of model error is statistically tested using a criterion called the Monotonic Trend (MT) criterion. If the MT criterion is satisfied by the fitted VESD regression function, this VESD function is considered admissible and is used to predict the error in the final surrogate model (constructed over all sample points). Otherwise, a stable implementation of k -fold cross-validation, called *PEMF-based k -fold cross-validation*, is used to represent the error in the final surrogate. The *PEMF-based k -fold cross-validation* method is defined as the modal value of the error of the intermediate surrogate model constructed using n^{Nitr} sample points (i.e., the modal error value in the last iteration of PEMF). The median and the maximum error measures of the final surrogate are then respectively given by $\text{Mo}_{\text{med}}^{\text{Nitr}}$ and $\text{Mo}_{\text{max}}^{\text{Nitr}}$ (as is further defined in Eq. 3.10).

In the MT test, we use the Pearson correlation coefficient [111] to measure the linear correlation between the log-level and the log-log transformation of Error with respect to the Sample density in the Type 1 and Type 2 VESD functions. In this test, the following standard mathematical properties are employed:

- (i) if a (x, y) input/output data set follows an exponential relationship, the transformed $(x, \log y)$ follows a linear relationship, and
- (ii) if a (x, y) input/output data set follows a purely p^{th} order power relationship, the transformed $(\log x, \log y)$ follows a linear relationship.

In this test, the Pearson coefficient for the Type 1 and Type 2 VESD functions is defined as:

$$\rho_{\lambda, \varphi} = \frac{\text{Cov}(\lambda, \varphi)}{\sigma_{\lambda} \sigma_{\varphi}} \quad (3.6)$$

where

$$\lambda = \ln(Mo^{t=1,2,\dots,N^{itr}}) \quad (3.7)$$

$$\varphi = \begin{cases} n^{t=1,2,\dots,N^{itr}} & \text{VESD Type 1} \\ \ln(n^{t=1,2,\dots,N^{itr}}) & \text{VESD Type 2} \end{cases} \quad (3.8)$$

In Eq. 3.6, Cov is the *covariance*, σ is the standard deviation, and $\rho_{\lambda,\varphi}$ is a *Indicator of Monotonicity* (defined by the Pearson Correlation Coefficient). The closer $\rho_{\lambda,\varphi}$ is to -1 , the stronger is the monotonic (decreasing) relationship between the model error measure and sample density. If the threshold condition value of $\rho_{\lambda,\varphi}$ is given by ρ_{cr} , then $p = \frac{\rho_{cr}}{\sqrt{(1-\rho_{cr}^2)/d_F}}$ follows a *t*-distribution with the degrees of freedom given by $d_F = N^{itr} - 2$. The threshold condition value, ρ_{cr} , for a C_{cr} level of confidence can then be determined by

$$\rho_{cr} = \frac{p_{C_{cr},d_F}}{\sqrt{(p_{C_{cr},d_F}^2 + d_F)}} \quad (3.9)$$

For a given level of confidence (C_{cr}), if the *Indicator of Monotonicity*, $\rho_{\lambda,\varphi}$, is not greater than the threshold condition value of ρ_{cr} , the monotonically decreasing characteristics is considered to hold. In that case, the fitted regression function (VESD) is used to predict the fidelity of the final surrogate model (*VESD* ($\{X_{in}\}$)). Conversely, if the Indicator of Monotonicity ($\rho_{\lambda,\varphi}$) is greater than the threshold condition value of ρ_{cr} , the VESD function is discarded, and instead a stable implementation of *k*-fold cross-validation (PEMF-based *k*-fold) is used, as shown below:

$$E^{PEMF} = \begin{cases} VESD(\{X_{in}\}) & \rho_{\lambda,\varphi} \leq \rho_{cr} \\ Mo^{N^{itr}} & \rho_{\lambda,\varphi} > \rho_{cr} \end{cases} \quad (3.10)$$

3.3 Investigating the Stability of PEMF error measures

The individual steps of the PEMF method at each iteration is somewhat analogous to k -fold cross-validation. In k -fold cross-validation, errors are generally evaluated for a specific value of k . In the PEMF method, the errors are evaluated iteratively for different values of k , in order to predict the level of error in the final surrogate. This novel approach gives PEMF the predictive capability that is otherwise lacking in standard k -fold or leave-one-out cross-validation. More importantly, PEMF seeks to formulate and use more stable measures of error compared to those in cross-validation. In k -fold cross-validation, for one specific k , mean errors are used to represent the level of error, as shown in the pseudo code under Algorithm 1. In the PEMF method, the median error on each heuristic set of intermediate test points is first estimated, followed by the determination of the statistical mode of the median errors. The modal error value is then used to represent the level of error at any given k . The novel use of the modal value of median errors (and similarly modal value of maximum errors) promotes a monotonic decrease of the error measure with sample density, as opposed to the untraceable noisy variation that is often characteristic of mean or RMS error measures.

To illustrate the potential greater stability of PEMF error measures (with respect to their relation to sample density) compared to conventional measures, a two design variable benchmark problem (Branin-Hoo function (Sec.3.4.2)) is considered. The size of the sample data set, N , is defined to be 500, and the region of interest is set to be the entire design

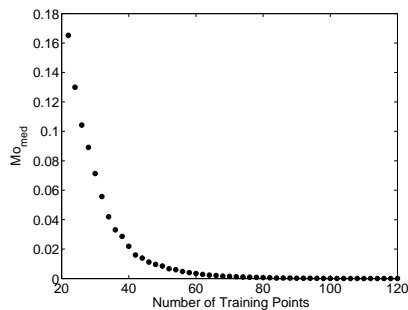
Algorithm 1 : Standard K -fold cross validation

INPUT:
Set k Set $N = \binom{N_i}{k}$ **for** $j = 1, 2, \dots, N$ **do** **for** $i = 1, 2, \dots, K$ **do** Estimate actual value on i^{th} training point; $y_i = \text{System}(x_i)$ Estimate predicted value on i^{th} training point; $\hat{y}_i = \text{Surrogate}(x_i)$ Estimate RAE on i^{th} training point; $RAE_i = \left| \frac{y_i - \hat{y}_i}{y_i} \right|$ **end for**

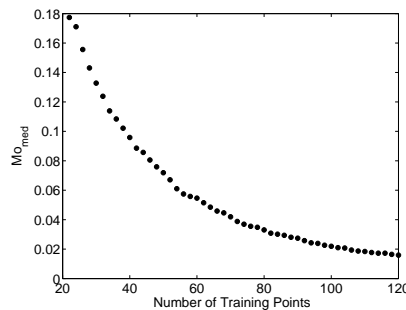
Evaluate mean of the relative absolute errors value;

 $E_j = \text{mean}(RAE_i), i = 1, 2, \dots, K$ **end for**

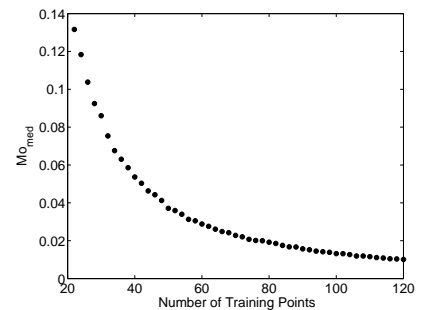
Evaluate mean of the mean errors value;

 $E^{k\text{-fold}} = \text{mean}(E_j), j = 1, 2, \dots, N$ 

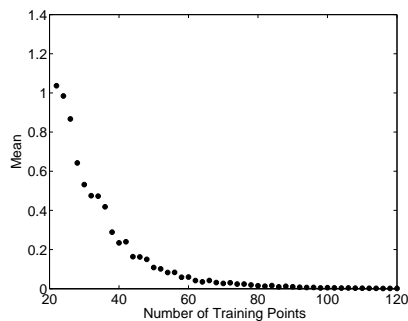
(a) Kriging



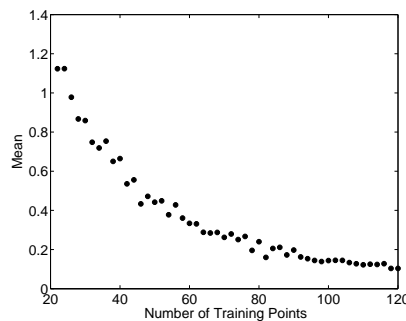
(b) RBF



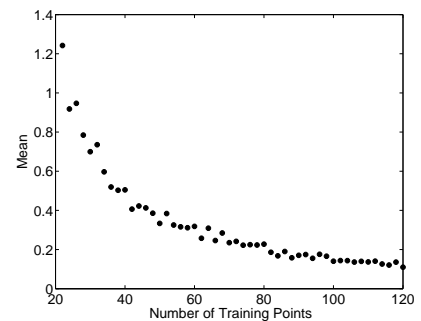
(c) E-RBF

Figure 3.3: Variation of Estimated Modal values of Median Errors with Sample Density using PEMF (Branin-Hoo Function)


(a) Kriging



(b) RBF



(c) E-RBF

Figure 3.4: Variation of typical Mean Error with Sample Density, as given by standard k -fold cross-validation (Branin-Hoo Function)

space. The numerical settings of this problem is given by

$$N_i = N, n^t = 20 + 2t, \text{ and } M^t = 200$$

$$, t = 1, 2, 3, \dots, 50$$

Based on this definition, at the first iteration, the intermediate surrogates are constructed using 22 sample points, and tested over 478 points, i.e., $N_{TR}^1 = 22$, and $N_{TE}^1 = 478$, where N_{TR}^1 and N_{TE}^1 represent the number of training points and test points, respectively. In the same way, at the last iteration, $N_{TR}^{50} = 120$, and $N_{TE}^{50} = 380$.

The relation of *the Mode of Median* and *the Mean of the Mean* errors with sample density in different surrogates (Kriging, RBF, and E-RBF) are illustrated in Fig. 3.3 and Fig. 3.4, respectively. These figures illustrate that the mode of median errors decrease with a practically monotonic trend, with increasing number of training points. Such a monotonic tendency cannot be associated with the *Mean of the Mean* error, as shown in Figs. 3.4(a)- 3.4(c). It is also important to consider that model errors may not follow a normal distribution, and that the most probable value of error (modal value) might be of greater practical importance (more insightful) than a mean value. From these perspectives, PEMF error measures are likely to provide a more meaningful understanding of the fidelity of a surrogate model in the context of its use in analysis and design.

3.4 Application of the PEMF Method

3.4.1 Surrogate models

The effectiveness of the PEMF method in predicting the regional and global errors is explored for application with Kriging, RBF, and E-RBF surrogate on Branin-Hoo function

with two design variables, Perm function with 10 design variables, and Dixon & Price function with 50 design variables. The median and maximum errors evaluated using PEMF are compared with the actual error evaluated on additional test points. To perform a fair comparison, the estimated median and maximum errors on heuristic subsets of additional test points are used to fit error distributions. The statistical mode of the median and the maximum error distributions (estimated over additional test points) are used as reference values (as shown in Algorithm 2), to investigate performance of PEMF.

Algorithm 2 : Quantifying the mode of median and maximum errors, estimated on additional test points (for performance validation of PEMF)

INPUT:

Set Number of additional test points N^{test}

Set Size of combinations, n ($n < N^{test}$)

Set Number of combinations, P ($P < \binom{N^{test}}{n}$)

for $i = 1, 2, \dots, N^{test}$ **do**

 Estimate actual value on i^{th} test point; $y_i = \text{System}(x_i)$

 Estimate predicted value on i^{th} test point; $\hat{y}_i = \text{Surrogate}(x_i)$

 Estimate RAE on i^{th} test point; $RAE_i = \left| \frac{y_i - \hat{y}_i}{y_i} \right|$

end for

for $j = 1, 2, \dots, P$ **do**

 Select a random subset of size n from RAEs,

 Evaluate the median and maximum of the selected RAEs value RAE_{med} and RAE_{max} ;

$RAE_{med} = \text{median}(RAE_i)$, $RAE_{max} = \text{max}(RAE_i)$, and $i = 1, 2, \dots, n$.

end for

Fit a distribution of the median and the maximum errors over all P combinations

Determine the mode of the error distributions; E_{Mo-med}^{Actual} and E_{Mo-max}^{Actual}

To illustrate the potential greater effectiveness of the PEMF error metric over popular *cross-validation* error, the mean and maximum errors given by *leave-one-out cross-validation* are also compared with the actual mean and maximum errors. In this case, the mean and maximum errors estimated on additional test points (as described in Algorithm 3) are used as reference values.

To implement the Kriging method, the DACE (design and analysis of computer experiments) package developed by Lophaven et al. [112] is used. The bounds on the correlation parameters in the nonlinear optimization in Kriging, θ_l and θ_u , are specified to be 0.1 and 20, respectively. In the Kriging surrogate, the zero-order polynomial function is used as the regression model. To implement the RBF surrogate, the multiquadric radial basis function [10] is used, where the shape parameter is set to $c = 0.9$. In implementing the E-RBF surrogate [28], the shape parameter is set to $c = 0.9$, the λ parameter is set to 4.75, and the order of monomial in the non-radial basis functions is fixed at 2.

Algorithm 3 : Quantifying the mean and maximum error estimated on additional test points (for performance validation of *cross-validation*)

INPUT:

Set Number of additional test points, N^{test}

for $i = 1, \dots, N^{test}$ **do**

Estimate actual value on i^{th} test point; $y_i = \text{System}(x_i)$

Estimate predicted value on i^{th} test point; $\hat{y}_i = \text{Surrogate}(x_i)$

Estimate RAE on i^{th} test point; $RAE_i = \left| \frac{y_i - \hat{y}_i}{y_i} \right|$

end for

Evaluate the mean and maximum of the relative absolute errors value;

$E_{\text{mean}}^{\text{Actual}} = \text{mean}(RAE_i)$ and $E_{\text{max}}^{\text{Actual}} = \text{max}(RAE_i)$,

$i = 1, 2, \dots, N^{test}$

Table 3.1: Numerical setup for the benchmark functions

Function	Type of Error Measure	Total No. of sample points, $\{X\}$	No. of iterations, N^{itr}	No. of inside-region points, $\{X_{in}\}$	No. of training points at each iteration, n^t
Branin-Hoo Case I	Regional Error	30	4	11	$2t$
Branin-Hoo Case II	Global Error	30	4	30	$19+2t$
Perm	Global Error	50	4	50	$30+4t$
Dixon & Price	Global Error	200	4	200	$150+10t$

3.4.2 Numerical Experiments

PEMF is first applied to the following popular 2D benchmark function:

Branin-Hoo function

$$f(x) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6 \right)^2 + \quad (3.11)$$

$$10 \left(1 - \frac{1}{8\pi} \right) \cos(x_1) + 10$$

where $x_1 \in [-5 \ 10]$, $x_2 \in [0 \ 15]$

The effectiveness of the PEMF method in predicting the global error in high dimensional problems is then explored by applying it to two other analytical benchmark test problems:

- Perm Function (10 variables)

$$f(x) = \sum_{k=1}^n \left\{ \sum_{j=1}^k (j^k + 0.5) \left[\left(\frac{x_j}{j} \right)^k - 1 \right] \right\}^2 \quad (3.12)$$

where $x_i \in [-n \ n + 1]$, $i = 1, \dots, n$

$$n = 10$$

- Dixon & Price Function (50 variables)

$$f(x) = (x_1 - 1)^2 + \sum_{i=2}^n i (2x_i^2 - x_{i-1})^2 \quad (3.13)$$

where $x_i \in [-10 \ 10]$, $i = 1, \dots, n$

$$n = 50$$

The numerical settings for the application of PEMF in predicting the median and maximum errors of surrogate models for the benchmark functions are provided in Table 3.1,

which lists (i) the number of training points, (ii) the number of iterations, (iii) the number of inside-region training points based on the predefined region's boundary, and (iv) the size of the training set at each iteration. Two different cases of error quantification are explored for the Branin-Hoo function. In case *I*, the PEMF method is used for regional error measurement, where the boundaries of the region of interest are defined as: $x_1 \in [-2 \ 7]$, $x_2 \in [3 \ 12]$. While in case *II*, PEMF is used as a global error measurement method to estimate the model fidelity in entire design domain. To implement the MT criterion, the *threshold condition value* (ρ_{cr}) in benchmark problems is estimated to be $\rho_{cr} = \frac{p_{0.95,2}}{\sqrt{(p_{0.95,2}^2 + 2)}} = -0.950$, for a specified level of confidence of $C_{cr} = 0.95$ and $N^{itr} = 4$.

3.4.3 Performance Criteria

The errors evaluated on additional test points, summarized under Algorithm 2 and Algorithm 3, are used to compare the performance of *PEMF* and *leave-one-out cross-validation*. Although PEMF and leave-one-out cross-validation have different mathematical characteristics, the purpose of their application in testing surrogates (for analysis/design/optimization) is the same. In this chapter, we therefore compare the PEMF method with the popular leave-one-out cross-validation method, particularly with respect to their abilities to predict the quality or fidelity of surrogate models. This comparison is driven by the premise that the primary ‘end-use’ applications of PEMF and cross-validation methods are similar, namely model testing, model selection, and model refinement. The relative differences between the actual errors and the PEMF errors are evaluated as

$$\begin{aligned} R_{PEMF}^{med} &= \left| \frac{E_{Mo-med}^{Actual} - E_{med}^{PEMF}}{E_{Mo-med}^{Actual}} \right| \times 100\% \\ R_{PEMF}^{max} &= \left| \frac{E_{Mo-max}^{Actual} - E_{max}^{PEMF}}{E_{Mo-max}^{Actual}} \right| \times 100\% \end{aligned} \quad (3.14)$$

The relative difference between the actual errors and the errors given by *leave-one-out cross-validation* are evaluated as

$$\begin{aligned} R_{CV}^{mean} &= \left| \frac{E_{mean}^{Actual} - E_{mean}^{CV}}{E_{mean}^{Actual}} \right| \times 100\% \\ R_{CV}^{max} &= \left| \frac{E_{max}^{Actual} - E_{mean}^{CV}}{E_{max}^{Actual}} \right| \times 100\% \end{aligned} \quad (3.15)$$

In this section, the number of additional test points used to compare the error measures is equal to 50 times of the number of inside-region sample points ($50 \times N_i$).

3.5 Results and Discussion

3.5.1 Investigating the Performance of PEMF

Figures 3.5 and 3.7 show the VESD regression functions constructed to predict the median errors in the surrogates for Case *I* and *II*, respectively. For the sake of illustration, all the regression coefficients in the figures are rounded to two decimal places. The *Indicator of Monotonicity*, $\rho_{\lambda, \varphi}$, for each model is also shown in these figures. The distributions of median errors, and the mode of the median error distributions (Mo_{med}) are shown for each iteration. In these figures, the shaded Δ symbol represents the quantified mode of median errors at each iteration; the hollow Δ symbol represents the predicted mode of median error in the final surrogate models.

Table 3.2: Median Error Estimated using PEMF and a large pool of additional test points (E_{Mo-med}^{Actual})

	Kriging		RBF		E-RBF	
	PEMF	Actual	PEMF	Actual	PEMF	Actual
Regional Modal Error (case <i>I</i>)	0.081	0.088	0.109	0.104	0.049	0.053
Global Modal Error (case <i>II</i>)	0.108	0.097	0.155	0.099	0.033	0.068

Actual: Error evaluated on a large pool of additional test points

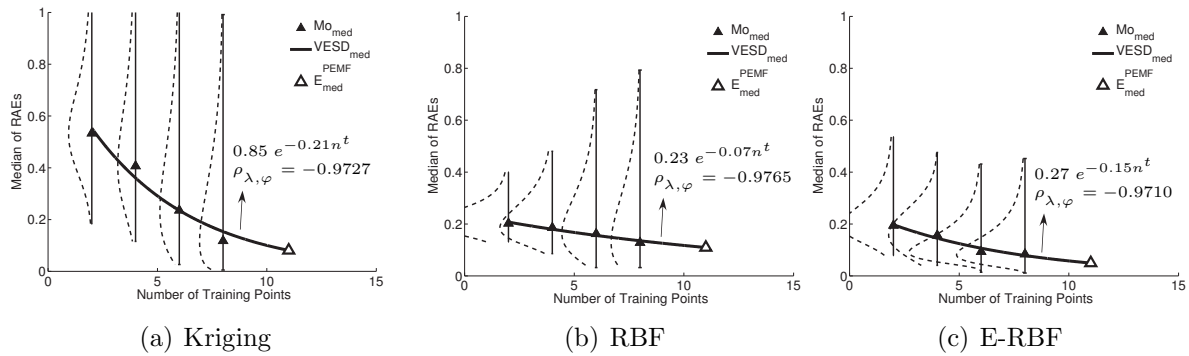


Figure 3.5: VESD function to predict the regional median error ($E_{\text{med}}^{\text{PEMF}}$) in Branin-Hoo Function (case I)

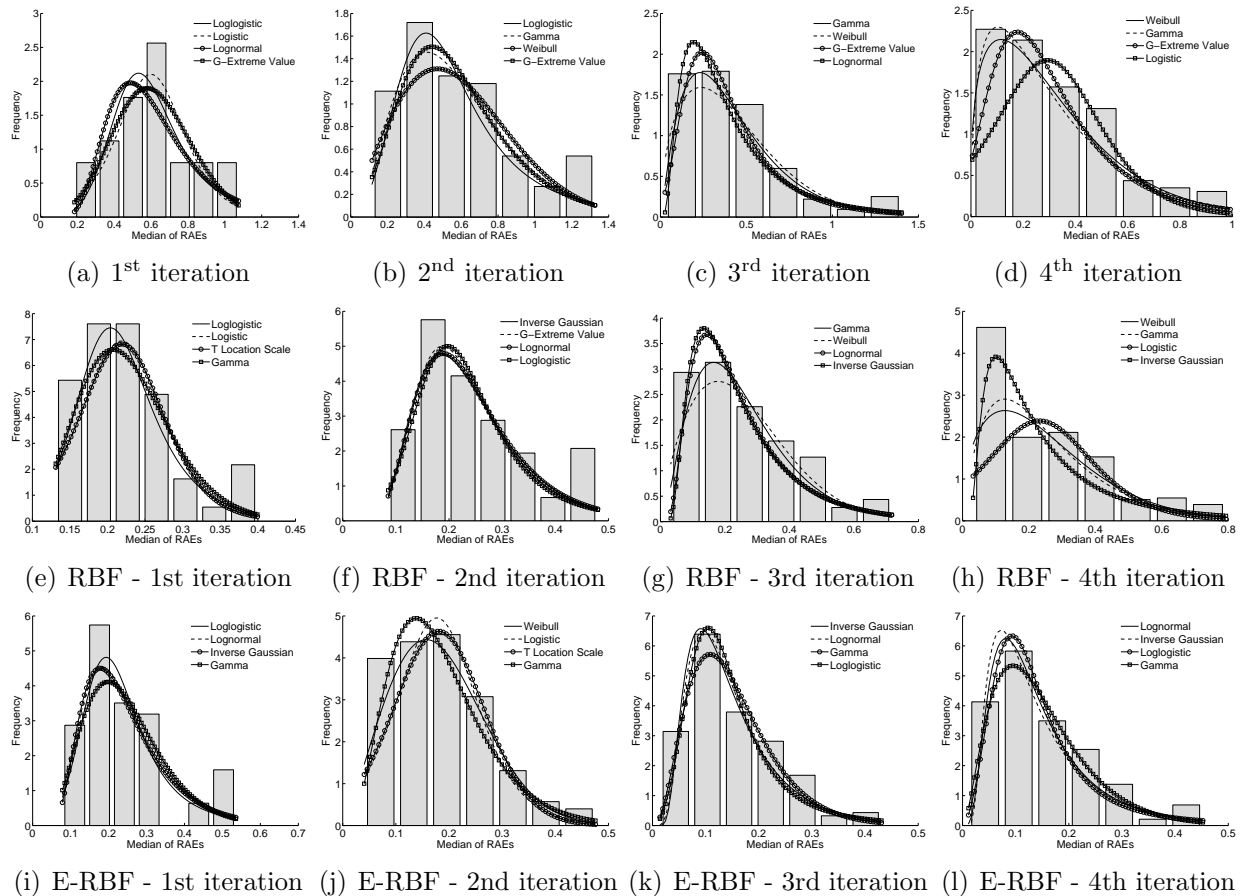


Figure 3.6: Distribution determination based on the χ^2 criterion on median regional error frequency diagrams in Kriging model constructed for Branin-Hoo Function (case I)

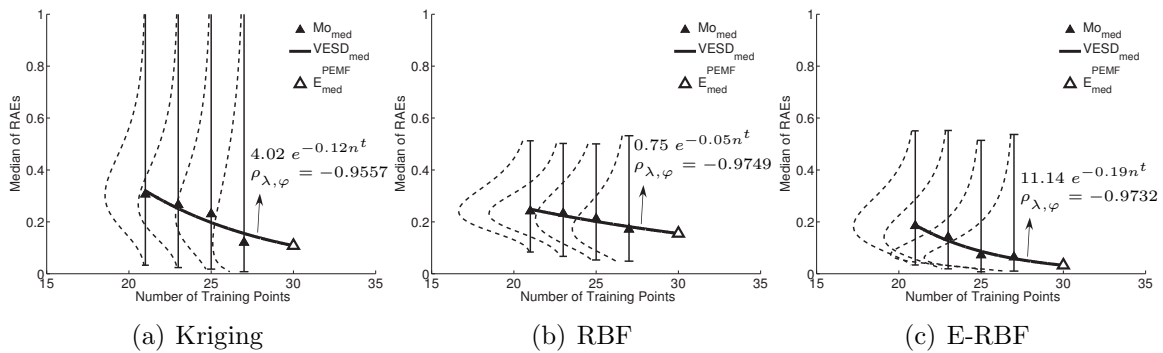


Figure 3.7: VESD function to predict the global median error ($E_{\text{med}}^{\text{PEMF}}$) in Branin-Hoo Function (case II)

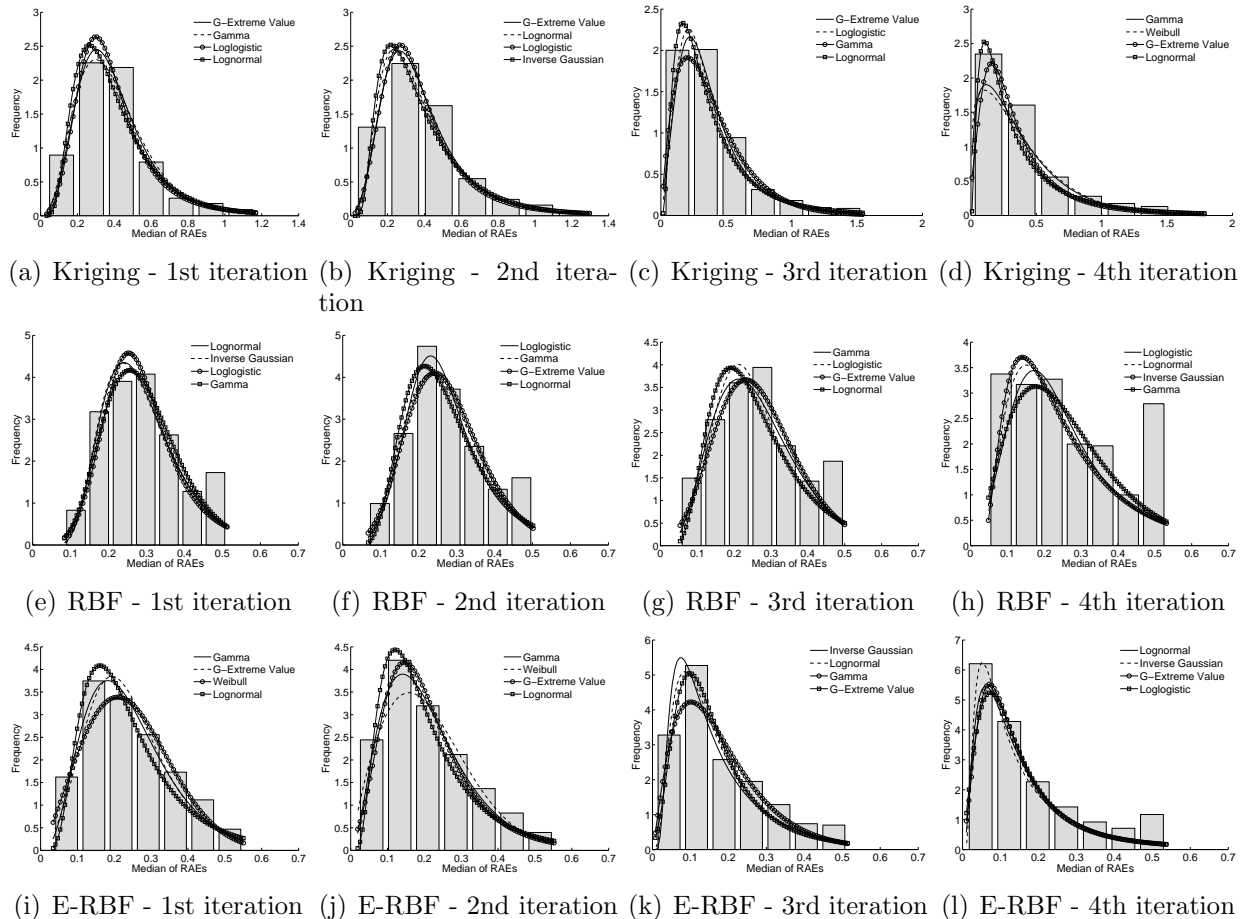


Figure 3.8: Distribution determination based on the χ^2 criterion on median global error frequency diagrams in Branin-Hoo Function (case II)

It is observed from Figs. 3.5 and 3.7 that the mode of the error distributions generally decrease with density of inside-region training points, since the estimated *Indicator of Monotonicity* in the two cases are less than the pre-estimated threshold (ρ_{cr}). Based on these observations, the variation of error with training point density can be represented by applying the VESD regression functions described in Section 2.2. The predicted median ($E_{\text{med}}^{\text{PEMF}}$) errors and actual errors in the final surrogates in Cases *I* and *II* are provided in Table 3.2. The PEMF error is within the same order of magnitude as the actual errors, which is a significant achievement in surrogate model error quantification. It is also important to note that PEMF provides additional helpful insights into the performance of the surrogates for the concerned problem. For example, in these case studies, it is observed (from Figs. 3.5 and 3.9) that the regional accuracy of the Kriging surrogate model is significantly more sensitive to the sample density than that of the RBF and E-RBF surrogates.

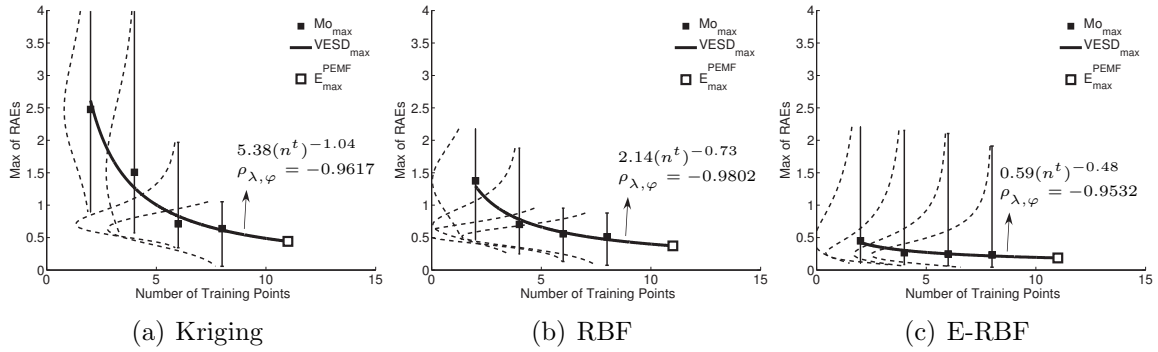


Figure 3.9: VESD function to predict the regional maximum error ($E_{\text{max}}^{\text{PEMF}}$) in Branin-Hoo Function (case *I*)

For the two cases of the Branin-Hoo problem, the VESD functions used to predict the mode of the maximum error, and the *Indicator of Monotonicity* ($\rho_{\lambda, \varphi}$) are illustrated in Figs. 3.9 and 3.10. The mode of the maximum error distributions in the intermediate surrogates is represented by the solid square symbol; the hollow square symbol represents the predicted mode of maximum error in the final surrogate. It is again observed that the

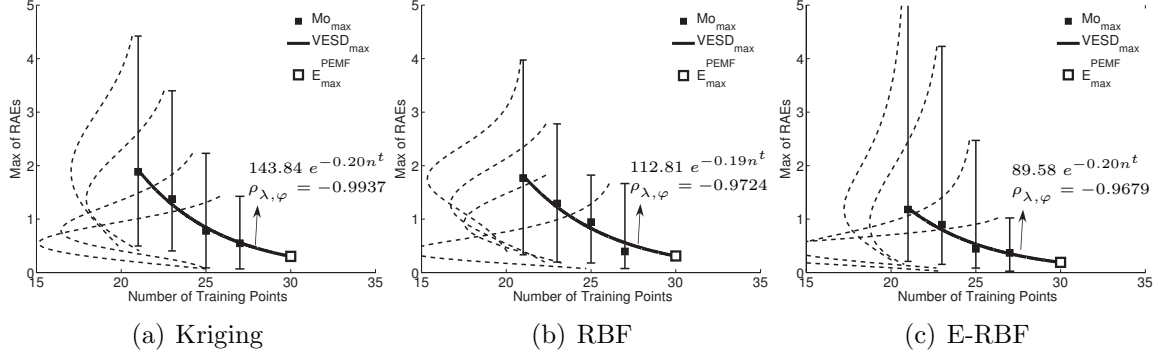


Figure 3.10: VESD function to predict the global maximum error (E_{\max}^{PEMF}) in Branin-Hoo Function (case II)

Table 3.3: Maximum Error Estimated using $PEMF$ and a large pool of additional test points ($E_{Mo-max}^{\text{Actual}}$)

	Kriging		RBF		E-RBF	
	PEMF	Actual	PEMF	Actual	PEMF	Actual
Regional Modal Error (case I)	0.442	0.336	0.373	0.402	0.187	0.394
Global Modal Error (case II)	0.358	0.302	0.370	0.309	0.191	0.330

Actual: Error evaluated on a large pool of additional test points

mode of the maximum error distributions decrease with increasing density of inside-region training points and the MT criterion is satisfied in the two cases. The predicted maximum error (E_{\max}^{PEMF}) and the actual errors evaluated on additional test points (Algorithm 2) in the final surrogate models for Cases I and II are given in Table 3.3.

3.5.1.1 Comparing PEMF with *cross-validation*

Figures 3.11 and 3.12 illustrate how the PEMF error measure and the RAE given by *leave-one-out cross-validation* compare with the actual error evaluated on additional test points. In these figures, the y-axis represents the relative difference of the error measures compared with the actual errors. These comparisons are given for all three types of surrogate models constructed, i.e., Kriging, RBF, and E-RBF.

The performance criteria defined in Section 3.3 (i.e., R-values in Eqs. 3.14 and 3.15) are

applied to compare the performance of the PEMF method and the *cross-validation* method with the actual errors in Cases *I* and *II*. The comparison results are provided in Tables 3.4 and 3.5. The smaller R-value obtained in each case is shown in boldface. From Figs. 3.11-3.12 and Tables 3.4-3.5, it is observed that the measure of surrogate model fidelity provided by the PEMF method is up to two orders of magnitude more accurate than the RAE estimated by *leave-one-out cross-validation*.

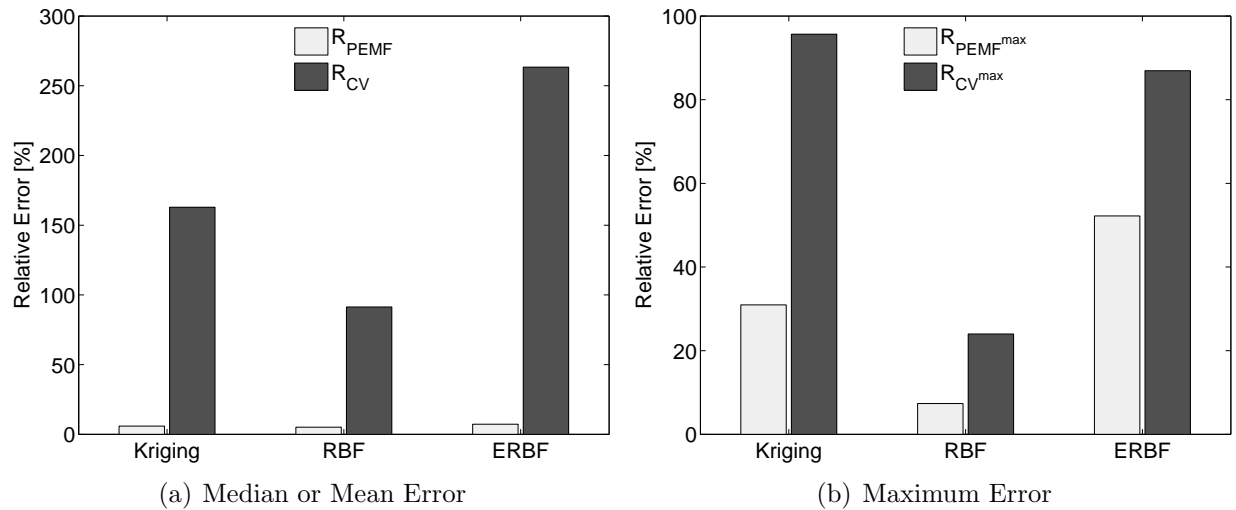


Figure 3.11: Comparison of the *PEMF* and *CV* error measures with the Actual regional error in Branin-Hoo Function (case *I*)

Table 3.4: Relative differences of the regional errors evaluated using *PEMF* and *CV* from the actual errors in Branin-Hoo Function (case *I*)

	Kriging		RBF		E-RBF	
	R_{PEMF}^{med}	R_{CV}^{mean}	R_{PEMF}^{med}	R_{CV}^{mean}	R_{PEMF}^{med}	R_{CV}^{mean}
Median or Mean Error	7.61 %	162.87 %	4.70 %	91.33 %	6.15 %	263.36 %
Maximum Error	31.18 %	95.66 %	7.07 %	23.99 %	52.63 %	86.92 %

3.5.2 Application of PEMF to higher dimensional benchmark problems

The $VESD_{med}$ and $VESD_{max}$ regression functions, and the *Indicator of Monotonicity* for these two benchmark problems are illustrated in Figs. 3.13-3.16. It is observed that in most of

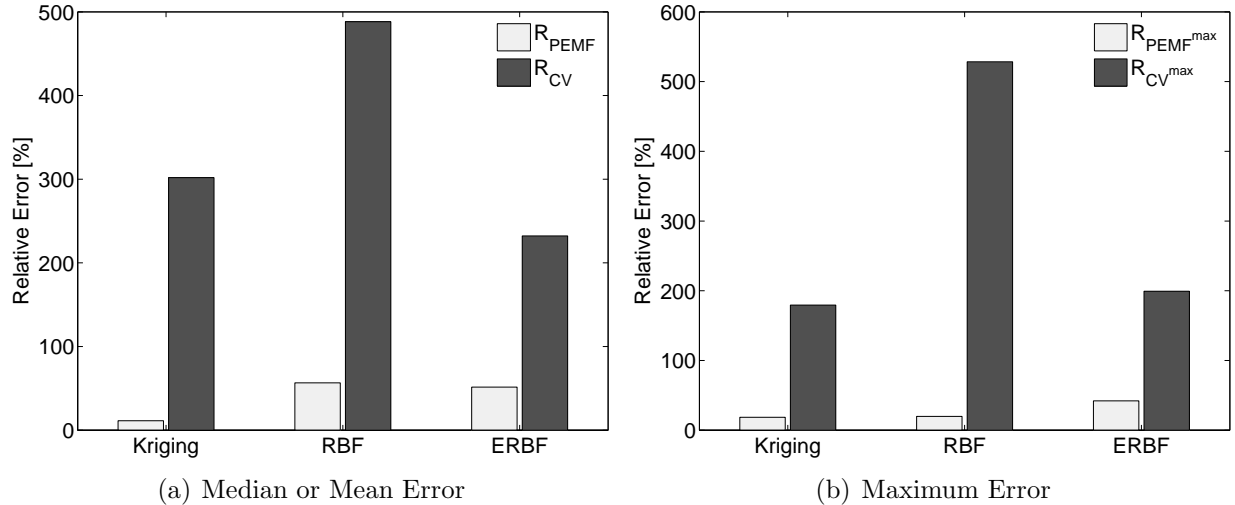


Figure 3.12: Comparison of the *PEMF* and *CV* error measures with the Actual global error in Branin-Hoo Function (case *II*)

Table 3.5: Relative differences of the global errors evaluated using *PEMF* and *CV* from the actual errors in Branin-Hoo Function (case *II*)

	Kriging		RBF		E-RBF	
	R_{PEMF}^{\max}	R_{CV}^{\max}	R_{PEMF}^{\max}	R_{CV}^{\max}	R_{PEMF}^{\max}	R_{CV}^{\max}
Median or Mean Error	11.34 %	301.89 %	56.57 %	488.25 %	51.47 %	232.15 %
Maximum Error	18.54 %	179.49 %	19.74 %	528.38 %	42.12 %	199.36 %

the cases the *MT criterion* is satisfied, and hence the VESD function can be used to predict the median and maximum errors of the constructed surrogates. The MT criterion is however not satisfied for the following cases: the median error of the Kriging model constructed for the Perm Function (Fig. 3.13(a)), and the median error of the RBF model (Fig. 3.15(b)) and the maximum error of the Kriging model (Fig. 3.16(a)) constructed for the Dixon & Price Function. Therefore, in these three cases, the PEMF-based *k*-fold cross-validation (i.e., the modal value of the error in the last iteration) are automatically considered to be the error measure of the surrogate model constructed using all sample points.

The comparison of the PEMF error measure and the RAE given by leave-one-out cross-validation with the actual error evaluated on additional test points are illustrated through

bar diagrams in Figs. 3.17-3.18. In these figures, the y-axis represents the relative difference of the error measures compared with the actual errors. The comparison results are also provided in Tables 3.6-3.7. A smaller R-value indicates a more accurate error metric. In these tables, the smaller R-values obtained for each case are shown in boldface. From Figs. 3.17-3.18 and Tables 3.6-3.7, it is observed that the PEMF method is up to two orders of magnitude more accurate than the leave-one-out cross-validation for these 10-50 dimensional test problems. These observations again establish the effectiveness of PEMF as a new more accurate approach to quantify surrogate model errors.

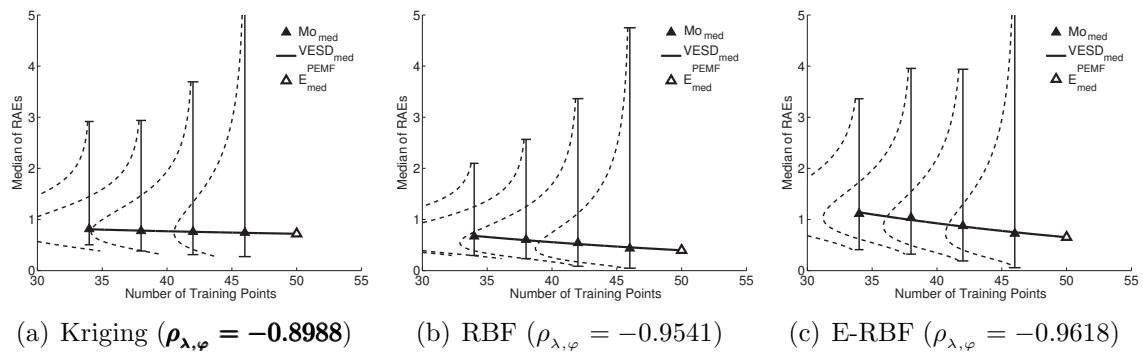


Figure 3.13: VESD function to predict the global median error ($E_{\text{med}}^{\text{PEMF}}$) in Perm Function

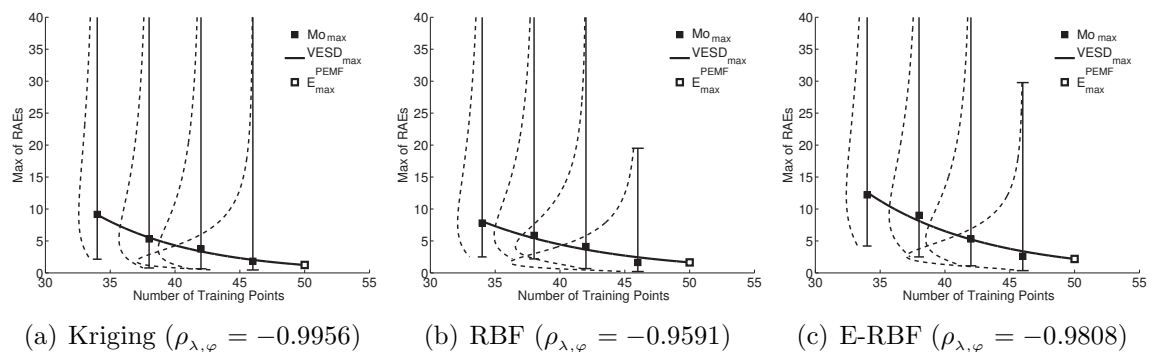


Figure 3.14: VESD function to predict the global maximum error ($E_{\text{max}}^{\text{PEMF}}$) in Perm Function

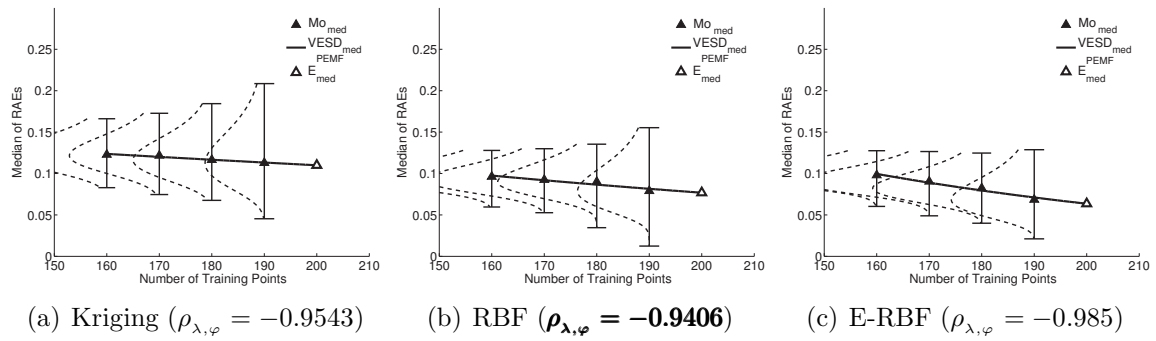


Figure 3.15: VESD function to predict the global median error ($E_{\text{med}}^{\text{PEMF}}$) in Dixon & Price Function

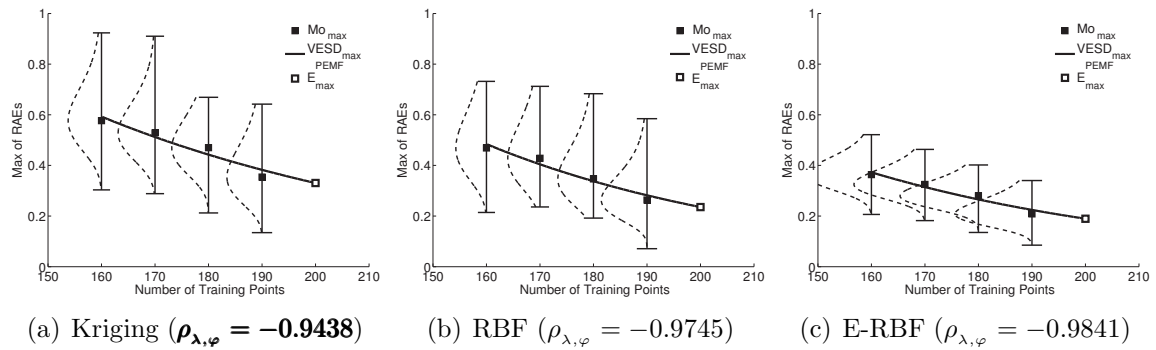


Figure 3.16: VESD function to predict the global maximum error ($E_{\text{max}}^{\text{PEMF}}$) in Dixon & Price Function

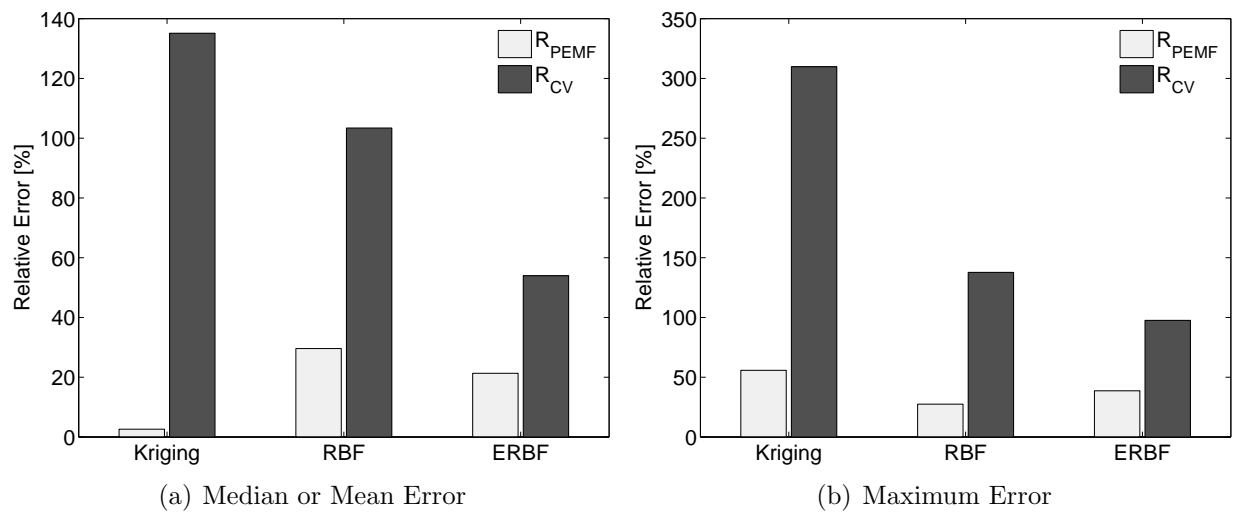


Figure 3.17: Comparison of the *PEMF* and *CV* error measures with the Actual global error in Perm Function

Table 3.6: Relative differences of the global errors evaluated using *PEMF* and *CV* from the actual errors in Perm Function

	Kriging		RBF		E-RBF	
	R_{PEMF}^{med}	R_{CV}^{mean}	R_{PEMF}^{med}	R_{CV}^{mean}	R_{PEMF}^{med}	R_{CV}^{mean}
Median or Mean Error	2.61 %	135.11 %	29.60 %	103.40 %	21.32 %	53.96 %
Maximum Error	55.75 %	309.70 %	27.45 %	137.70 %	38.59 %	97.53 %

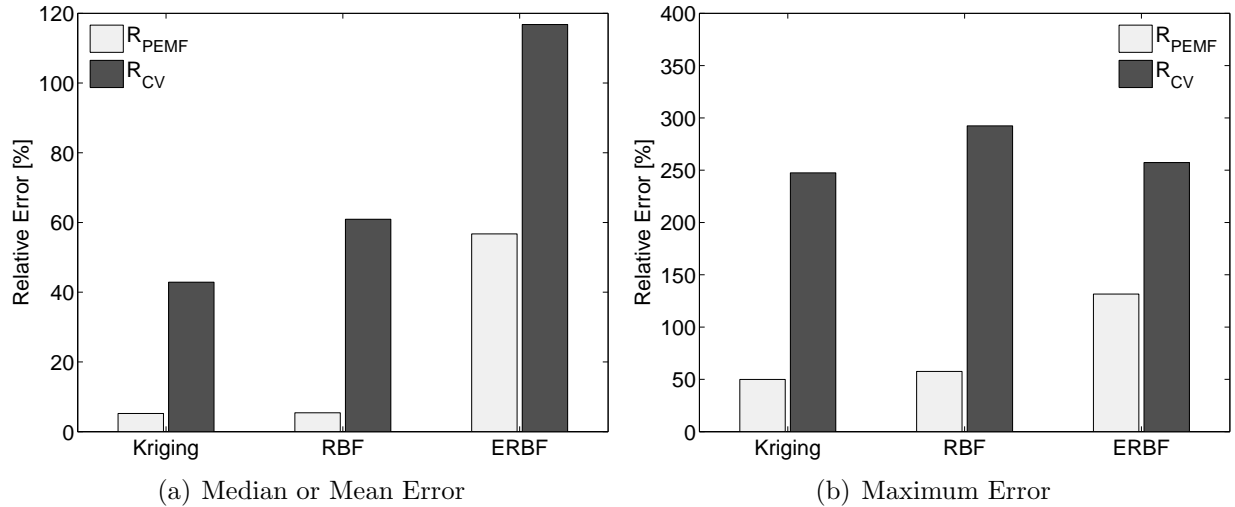


Figure 3.18: Comparison of the *PEMF* and *CV* error measures with the Actual global error in Dixon & Price Function

3.6 Chapter Summary

This chapter has made major advances in developing a new reliable approach to quantify the accuracy of surrogate models in a given region of the design domain or the entire design domain. Such an approach is useful for informed decision-making when using meta-models for analysis and optimization. This method called the Predictive Estimation of Model

Table 3.7: Relative differences of the global errors evaluated using *PEMF* and *CV* from the actual errors in Dixon & Price Function

	Kriging		RBF		E-RBF	
	R_{PEMF}^{max}	R_{CV}^{max}	R_{PEMF}^{max}	R_{CV}^{max}	R_{PEMF}^{max}	R_{CV}^{max}
Median or Mean Error	5.20 %	42.88 %	5.39 %	60.93 %	56.72 %	116.79 %
Maximum Error	49.93 %	247.45 %	56.68 %	292.39 %	131.54 %	257.32 %

Fidelity (PEMF), is a model-independent method for predicting the error in the actual surrogate constructed using all available sample points. In this method, intermediate surrogate models are constructed iteratively using multiple heuristic subsets of the available sample points. The remaining sample points are used to evaluate the error in the estimated function at that iteration. In this method, the model error at each iteration is defined by the mode of the median error and the maximum error distributions – this is expected to promote greater stability compared to mean or mean squared error measures. Regression models are then developed to represent the error in the surrogates as a function of the density of training points. These regression models are then extrapolated to predict the fidelity of the final surrogate under the condition that a *Monotonic Trend (MT) criterion* is satisfied. The *MT criterion* is a mechanism to statistically test the feasibility of the monotonic relationship between the modal error values and training sample density. If the *MT criterion* is not satisfied, the estimated modal error values from the last iteration is used to represent the measure of fidelity of the surrogate model – which is essentially a more stable implementation of k -fold cross-validation. The effectiveness of the new PEMF method is illustrated by applying it to 2-variable, 10-variable, and 50-variable benchmark problems, where the types of surrogate models constructed include Kriging, RBF, and E-RBF surrogates. These numerical experiments show that the proposed method provides up to two orders of magnitude greater accuracy in measuring surrogate model error compared to the relative absolute error estimated by *leave-one-out cross-validation*.

CHAPTER 4

Practical Implementation & Parametric Analysis of Predictive Estimation of Model Fidelity (PEMF)

How robust is the PEMF method?, and How does the PEMF method compare with other approaches, such as leave-one-out cross-validation?

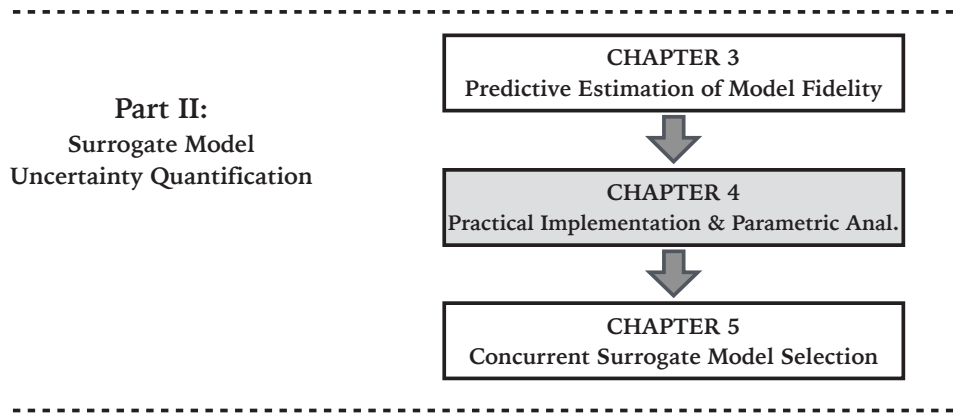


Figure 4.1: Chapter 4 orientation

4.1 Chapter Introduction

In the previous chapter, major advances in developing a new reliable approach to quantify the accuracy of surrogate models were presented. This method, called, “Predictive

Portions of this chapter previously appeared as:

Mehmani, A., Chowdhury, S., and Messac, A., “Predictive Quantification of Surrogate Model Fidelity based on Modal Variations with Sample Density,” *Structural and Multidisciplinary Optimization*, 2015, and

Mehmani, A., Chowdhury, S., Zhang, J., and Messac, A., “Quantifying Regional Error in Surrogates by Modeling its Relationship with Sample Density,” *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA, Boston, Massachusetts, April 2013

Estimation of Model Fidelity (PEMF)” is a surrogate model-independent method for predicting the error in the actual surrogate constructed using all available sample points. In this chapter, we further investigate the robustness of the PEMF method in predicting the regional and global errors for applications with Kriging, RBF, and E-RBF surrogate models. To this end, statistical testing is performed (on a set of 500 randomly generated sample sets) to measure the significance of the difference between the PEMF error and the actual error evaluated on additional test points. Likewise, the statistical test is also performed on leave-one-out cross-validation.

In Chapter 3 we observed that the PEMF method requires the specification of two parameters before implementation: (i) β , which controls the fraction fold in the first iteration of PEMF, and (ii) δ , which controls the incremental value (step size) in PEMF. This chapter provides some guidelines regarding the choice of appropriate values for these parameters. Parameter analysis of the PEMF method enables minimization or correction of the bias that measures.

In chapter 3 we also performed a preliminary assessment of the accuracy of the PEMF method. We compared the performance of PEMF with the popular leave-one-out cross-validation on different benchmark problems. There is, however, the need to further investigate the potential advantages of using the PEMF method approach over other related error measurement methods. In this chapter, we explore how the PEMF method depends on the β and δ parameters. In the present chapter, we also extend the discussion comparing the PEMF approach with three different cross-validation derived methods: (i) leave-one-out cross-validation, (ii) k -fold cross-validation, and (iii) q -fold cross-validation.

The remainder of the chapter is organized as follows: Section 4.2 presents the statistical tests performed on the results of the PEMF and the leave-one-out cross validation method.

Parametric analysis on defining guidelines regarding the two parameters under the PEMF method is presented in Section 4.3. Section 4.4 illustrates the effectiveness and the utility of PEMF in quantifying a global error in practical problems. The Chapter summary is provided in Section 4.5.

4.2 Statistical test on results of PEMF and *leave-one-out* cross-validation method

In this section, statistical tests are performed to illustrate the robustness of the PEMF method. The paired t -test [113] is applied to measure the significance of the difference between estimated error measures, and the actual error evaluated on additional test points.

A set of 500 paired sample data is generated using optimal Latin Hypercube, which is defined as $X_1, X_2, X_3, \dots, X_{500}$. The size of each sample data set is defined to be $N_i = 30$, with all sample points denoted as inside-region sample points, i.e., the PEMF method is applied as a global error measurement. In this test, it is assumed that the size of inside-region training set at each iteration, t , is given by

$$n^t = \lfloor \frac{N_i}{N^{itr} + 1} t \rfloor \quad (4.1)$$

where the function $\lfloor x \rfloor$ returns the largest integer less than or equal to x . The differences between the errors estimated using PEMF and the actual error on each pair of observation

are defined as

$$D_{med}^{PEMF} = E_{med}^{PEMF} - E_{Mo-med}^{Actual} \quad (4.2)$$

$$D_{max}^{PEMF} = E_{max}^{PEMF} - E_{Mo-max}^{Actual}$$

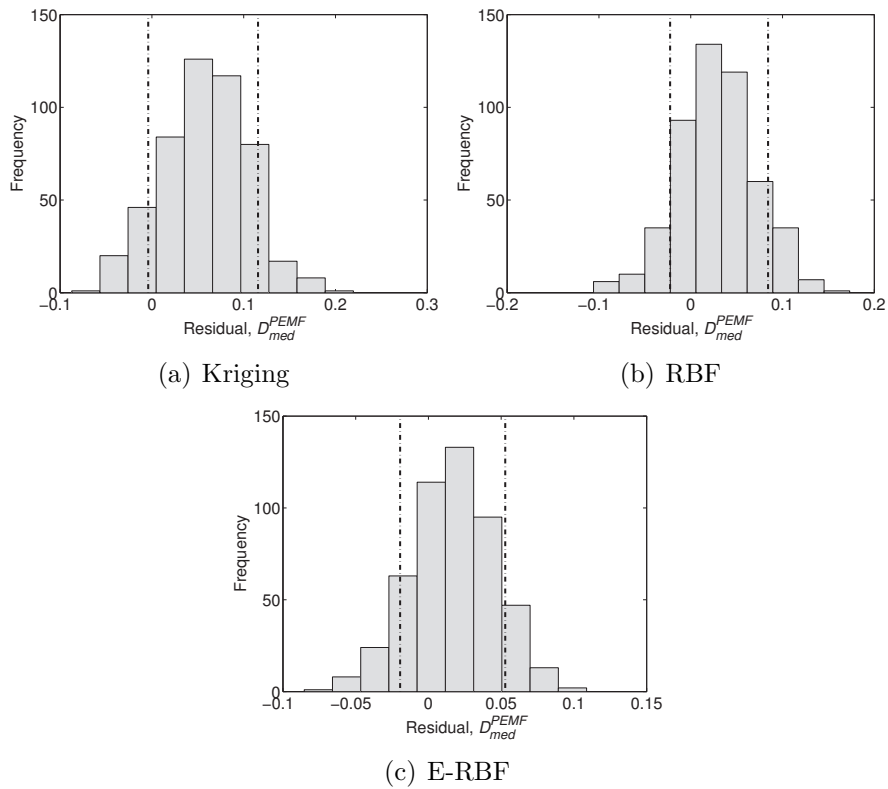


Figure 4.2: Distribution of the Residual of median error estimated using PEMF (E_{med}^{PEMF}) for Branin-Hoo Function

The frequency histogram of the residuals in different types of surrogates (Kriging, RBF, and E-RBF), for the median and maximum errors estimated using PEMF, are illustrated in Figs. 4.2 and Fig.4.4, respectively. In these figures, the 10th and the 90th percentiles of the residuals are illustrated using black dashed lines.

It is readily evident from the histograms in Figs. 4.2 and 4.4 that the residuals do not

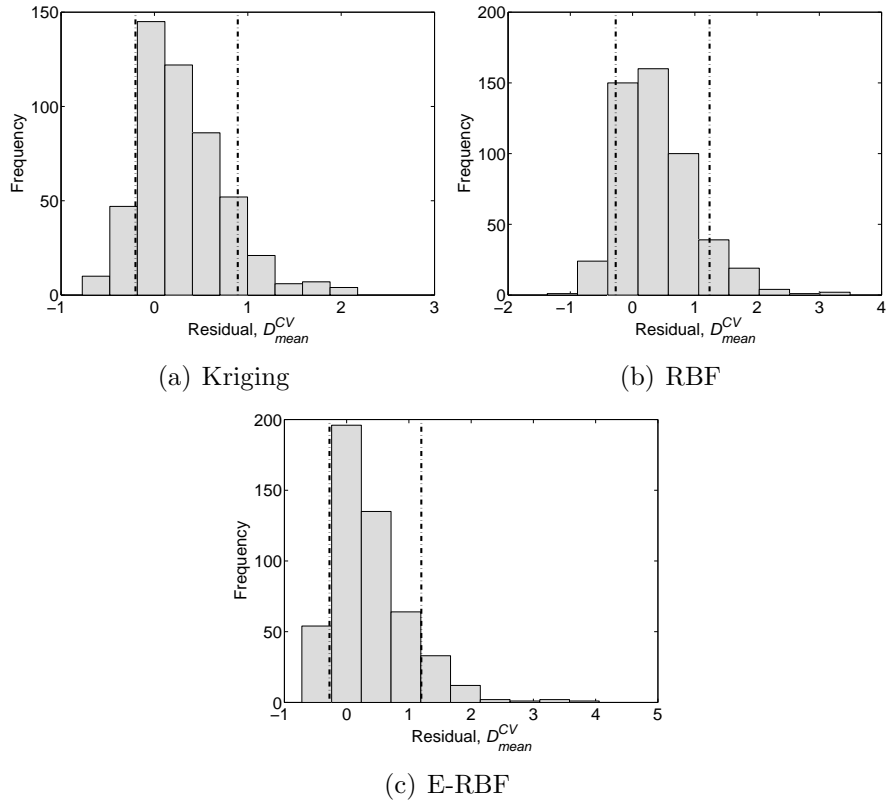


Figure 4.3: Distribution of the Residual of mean error estimated using *leave-one-out cross-validation* ($E_{\text{mean}}^{\text{CV}}$) for Branin-Hoo Function

necessarily follow a normal distribution. Hence, statistical methods based on a resampling approach such as the Bootstrapping method should be applied to perform the statistical inference [114]. In Bootstrapping method, it is assumed that the statistics of the sampling distribution is the probability of all possible values of the estimated statistics calculated from a sample of size n drawn from a given population [115]. To evaluate the 90% bootstrap confidence intervals (CI), after creating n bootstrapped sets from the original residuals, the mean value for each bootstrapped data set is independently computed. The 90% CI from the set of computed mean values are then estimated using the normal approximation method. Assuming the normal approximation is normally distributed with sample mean \bar{d}^* and standard deviation s_D^* for the $n = 1000$ bootstrapped resamples, a 90% confidence

interval on the difference in means, μ_D , is defined as

$$\bar{d}^* - t_{0.05, n-1} \times \frac{s_D^*}{\sqrt{n}} \leq \mu_D \leq \bar{d}^* + t_{0.05, n-1} \times \frac{s_D^*}{\sqrt{n}} \quad (4.3)$$

In this case, $t_{0.05, n-1}$ is the upper 0.05% point of the t distribution with $n - 1$ degrees of freedom. Confidence intervals that are narrower and closer to zero indicate higher level of robustness of the error measurement.

Table 4.1 presents the 90% confidence intervals (CI) for residuals of the mode of median and maximum errors estimated using the PEMF method in different surrogates.

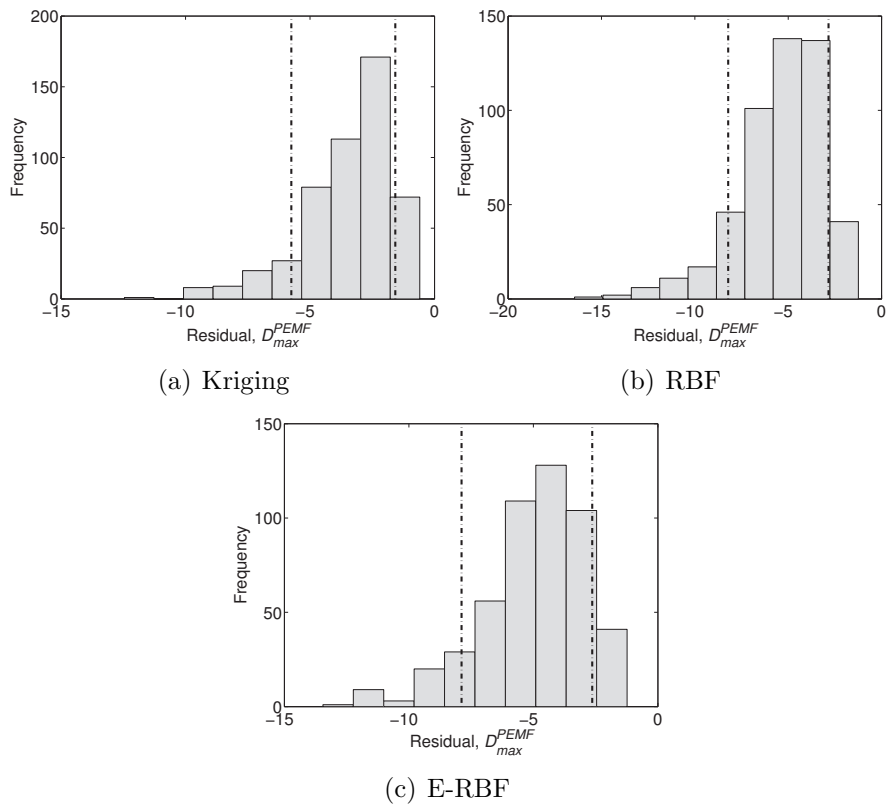


Figure 4.4: Distribution of the Residual of maximum error estimated using PEMF (E_{\max}^{PEMF}) for Branin-Hoo Function

Likewise, the statistical test is also performed for *leave-one-out* cross-validation. In this test, the same 500 paired sample data are used where the differences between the mean

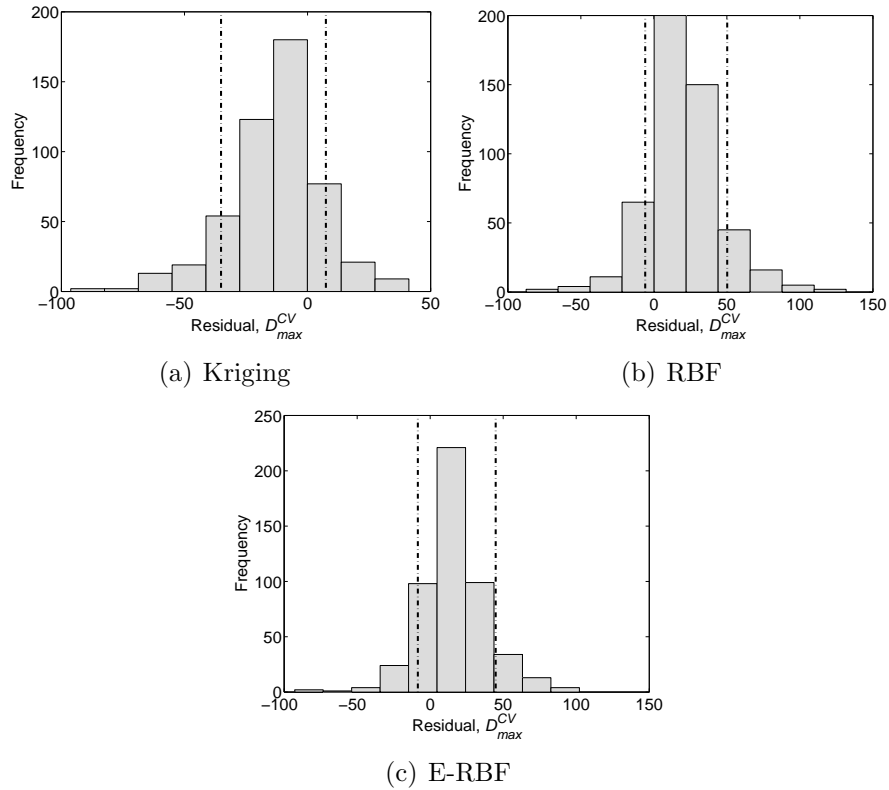


Figure 4.5: Distribution of the Residual of maximum error estimated using *leave-one-out cross-validation* E_{\max}^{CV} for Branin-Hoo Function

and maximum errors estimated by the *cross-validation* method and the actual error on each pair of observation are given by

$$D_{\text{mean}}^{CV} = E_{\text{mean}}^{CV} - E_{\text{mean}}^{\text{Actual}} \quad (4.4)$$

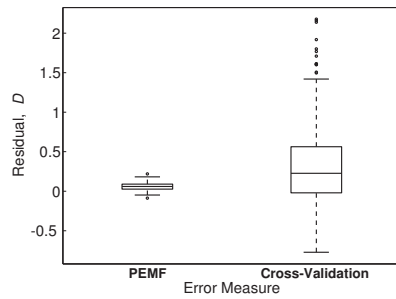
$$D_{\text{max}}^{CV} = E_{\text{max}}^{CV} - E_{\text{max}}^{\text{Actual}}$$

The actual errors are errors evaluated on additional test points, as given by Algorithm 3.

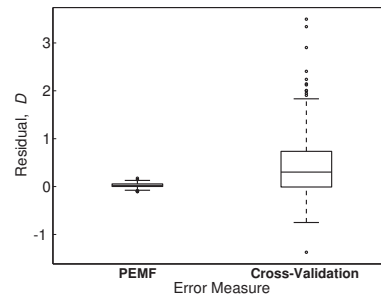
The frequency histogram of the residuals in different types of surrogates (Kriging, RBF, and E-RBF), for the mean and maximum errors estimated using *cross-validation*, are illustrated in Figs. 4.3 and 4.5, respectively. We again use the normal approximation method

Table 4.1: 90% bootstrapped CIs for residuals of E_{med}^{PEMF} and E_{mean}^{CV} in different surrogates

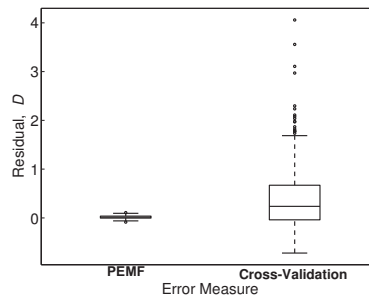
Measure	Kriging	RBF	E-RBF
E_{med}^{PEMF}	$0.057 \leq \mu_D \leq 0.059$	$0.028 \leq \mu_D \leq 0.029$	$0.017 \leq \mu_D \leq 0.017$
E_{mean}^{CV}	$0.296 \leq \mu_D \leq 0.299$	$0.400 \leq \mu_D \leq 0.404$	$0.373 \leq \mu_D \leq 0.377$



(a) Kriging



(b) RBF



(c) E-RBF

Figure 4.6: Interquartile Range box plot of the residuals of the central tendency of surrogate model errors (median in PEMF; and mean in *leave-one-out cross-validation*) for Branin-Hoo Function

Table 4.2: 90% bootstrapped CIs for residuals of E_{max}^{PEMF} and E_{max}^{CV} in different surrogates

Measure	Kriging	RBF	E-RBF
E_{max}^{PEMF}	$-3.479 \leq \mu_D \leq -3.469$	$-5.389 \leq \mu_D \leq -5.376$	$-4.991 \leq \mu_D \leq -4.980$
E_{max}^{CV}	$-12.934 \leq \mu_D \leq -12.829$	$20.175 \leq \mu_D \leq 20.317$	$16.344 \leq \mu_D \leq 16.473$

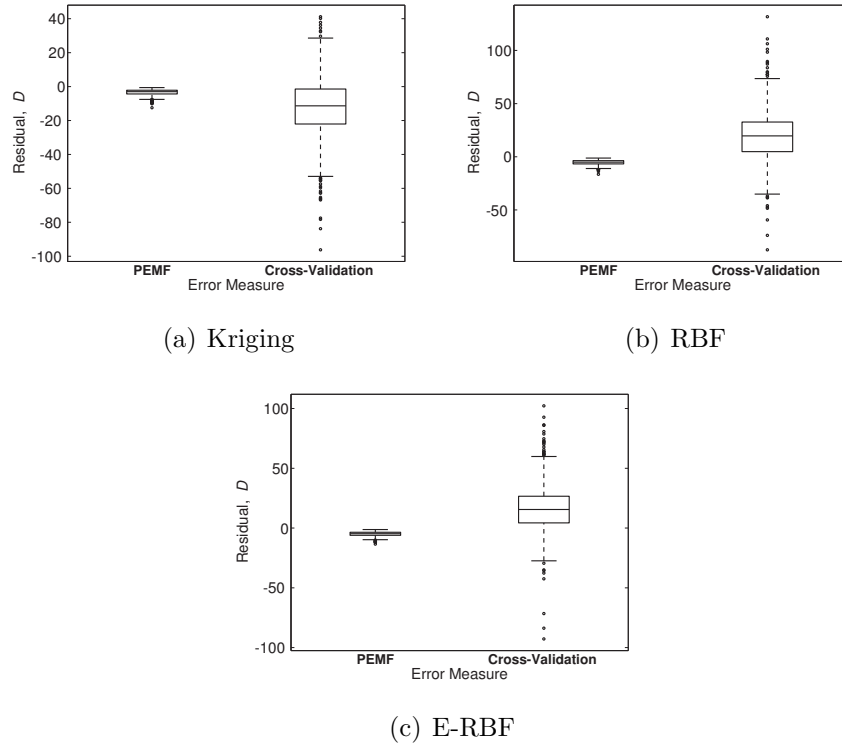


Figure 4.7: Interquartile Range box plot of the residuals of the maximum surrogate error for Branin-Hoo Function

to estimate the 90% bootstrapped confidence intervals on the difference in the means, μ_D , for the residuals of errors, which are reported in Table 4.2.

To compare the *PEMF* method and the *leave-one-out cross-validation* method, the distribution of the residuals of each surrogate is described by a box plot. The box plot indicates the deviation of the central tendency from zero (which is the expected value). Figures 4.6 and 4.7 respectively illustrate the distribution of residuals of the median and maximum errors estimated using *PEMF* and the mean error estimated using *cross-validation* in different surrogate models. From the results of the statistical test on *PEMF* and *cross-validation* (Figs. 4.2-4.7 and Tables 4.1-4.2), it is readily evident that the confidence intervals and the 25th and 75th percentiles for *PEMF* are significantly closer to zero (up to two orders of magnitude closer) compared to that for *cross-validation*. These observations show that

PEMF is expected to provide far superior robustness in quantifying surrogate model errors compared to standard *leave-one-out cross-validation*.

4.3 Parametric Analysis on PEMF's Parameters

The PEMF method requires the specification of two regularization parameters: (i) Fraction fold in the first iteration β and (ii) Normalized incremental value δ as given below

$$\beta = \frac{N(\{X_{TR}^{t=1}\})}{N(\{X\})} \tag{4.5}$$

$$\delta = \frac{N(\{X_{TR}^{t=2}\}) - N(\{X_{TR}^{t=1}\})}{N(\{X\})}$$

where $N(\cdot)$ represents the size of a set, $\{\cdot\}$; $\{X_{TR}^t\}$ represents the training set at t^{th} iteration of PEMF; and $\{X_{TE}^t\}$ represents the test set at t^{th} iteration of PEMF. As illustrated in Eq.4.5, the first regularization parameter, β controls the fraction fold in the first iteration of PEMF, and the second regularization parameter, δ controls the Incremental value or step size ($\Delta = \delta \times N(\{X\})$) in PEMF.

This section presents the analyzing of these parameters and their effects on the bias of the PEMF method. Such an analysis allows to tune the bias of the PEMF method as desired. To perform this analysis, the following assumptions have been made on the total number of iteration in PEMF (T) and the fraction fold in the first iteration:

$$T_{min} \leq T \leq T_{max}; T_{min} = 3, T_{max} = 7 \quad (4.6)$$

$$\beta_{min} = 0.5 \quad (4.7)$$

Under these assumptions, the parametric analysis is performed in two different cases defined below:

CASE I: $\beta + T \times \delta = 1$

The lower and upper bound of the two regularization parameters β and δ in this case are given by

$$\begin{aligned} 3 &\leq \frac{1 - \beta}{\delta} \leq 7 \\ \beta_{max} &= 1 - \frac{T_{min} \times D}{T} = 0.7 \\ \delta_{max} &= \frac{1 - \beta_{min}}{T_{min}} = 0.2 \\ \delta_{min} &= 0.1 \end{aligned} \quad (4.8)$$

CASE II: $\beta + T \times \delta = 1 - \delta$

The lower and upper bound of the two regularization parameters β and δ in this case are given by

$$\begin{aligned} 3 &\leq \frac{1 - (\delta + \beta)}{\delta} \leq 7 \\ \beta_{max} &= 1 - \frac{T_{min} + 1 \times D}{T} = 0.6 \\ \delta_{max} &= \frac{1 - \beta_{min}}{T_{min} + 1} = 0.15 \\ \delta_{min} &= 0.2 \end{aligned} \quad (4.9)$$

In this analysis, we consider five representative mathematical functions from 6D to 16D. These functions are illustrated in Table 4.3. Number of training points in all of these problems is equal to $10 \times D$.

Table 4.3: Functions linst for Parametric Analysis

No.	Function $f(x)$	Dimension D	Design domain
1 ^a	$f(x) = -\sum_{i=1}^4 c_i \exp\left\{-\sum_{j=1}^6 A_{ij} (x_j - P_{ij})^2\right\}$	6	$x_i \in [0 \ 1]$
2	$f(x) = \sum_{i=1}^{10} \{(\ln(x_i - 2))^2 + (\ln(x_i - 2))^2\} - (\prod_{i=1}^{10} x_i)^2$	10	$x_i \in [2 \ 10]$
3 ^b	$f(x) = \sum_{i=1}^{10} \left\{x_i \left(c_j + \ln \frac{x_j}{x_1 + x_2 + \dots + x_{10}}\right)\right\}$	10	$x_i \in [10^{-6} \ 10^{-5}]$
4	$f(x) = x_1^2 + x_2^2 + x_1 * x_2 - 14 \times x_1 - 16 \times x_2 + (x_3 - 10)^2$ $+ 4 \times (x_4 - 5)^2 + (x_5 - 3)^2 + 2 \times (x_6 - 1)^2 + 5 \times x_7^2$ $+ 7 \times (x_8 - 11)^2 + 2 \times (x_9 - 10)^2 + (x_{10} - 7)^2 + 45$	10	$x_i \in [1 \ 10]$
5 ^c	$f(x) = \sum_{i=1}^{16} \sum_{j=1}^{16} \{a_{ij}(x_i^2 + x_i + 1)(x_j^2 + x_j + 1)\}$	16	$x_i \in [0 \ 5]$

^a This function is Hartmann; the constants c , A , and P , are a 1×4 vector, a 4×6 matrix, and a 4×6 matrix, respectively.

^b $c_j = [-6.089, -17.164, -34.054, -5.914, -24.721, -14.986, -24.100, -10.708, -26.662, -22.179]$

^c This function is Hock & Schittkowski No.119; the constant a is 16×16 matrix

The relative difference between the errors evaluated on additional test points and the PEMF error is used to estimate the normalized bias of the PEMF method in different combinations of regularization parameters:

$$\gamma_{PEMF, \beta, \delta} = \left| \frac{E_{Mo-med}^{Actual} - E_{med}^{PEMF}}{E_{Mo-med}^{Actual}} \right| \times 100\% \quad (4.10)$$

Figures 4.8 - 4.12 illustrate the normalized bias of the PEMF method in different combinations of β and δ in Case *I* and Case *II* using Kriging and RBF surrogate models. In Function 1, the maximum bias of the PEMF method is estimated in Case *I* for both the Kriging and RBF models with $\delta = 0.1$ and $\beta = 0.6$ and 0.7 (from Fig. 4.8). For the second function, the maximum bias of the PEMF method is observed in Case *II* with $\delta = 0.1$ for RBF model, and $\delta \geq 1.4$ for Kriging model, as illustrated in Fig.4.9.

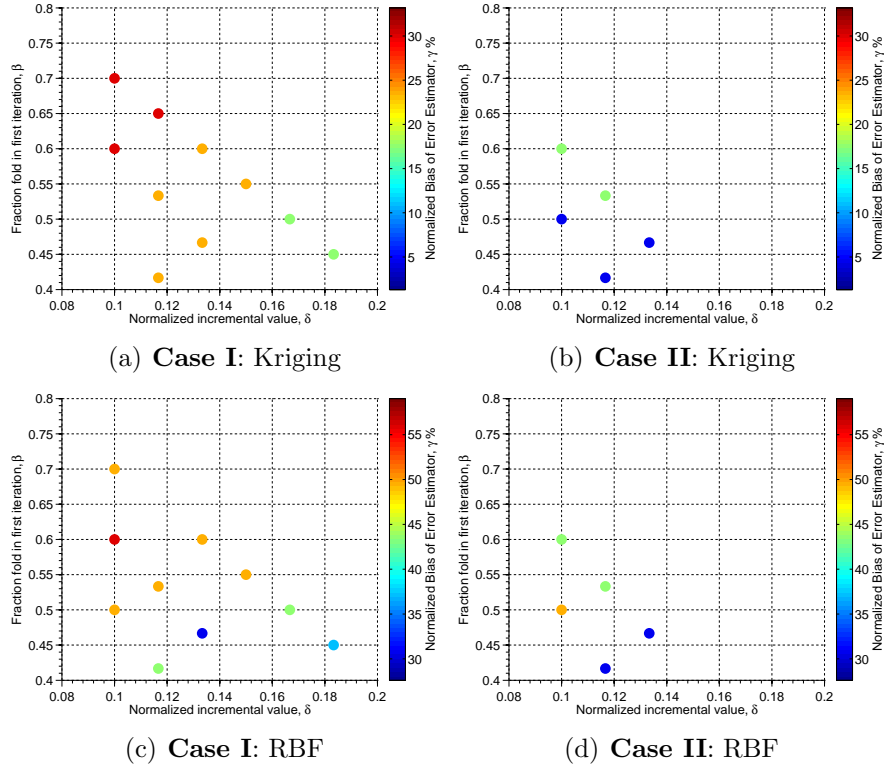


Figure 4.8: Function 1 (6D) - the normalized bias of the PEMF method in different combinations of β and δ in Case I and Case II

The bias of the PEMF method in Function 3 for the Kriging model (Figs. 4.10(a) - 4.10(b)) varies between %10 to %30 in Case I, and it varies between %10 to %60 in Case II. In this function, the maximum bias is again occurs in Case II for both the Kriging and RBF models. In Function 4, the bias of the PEMF method is less than %25 for all the combinations of δ and β in Case I and Case II in both the RBF and Kriging models. In this function, the maximum error is estimated in Case II as shown in Fig. 4.11(d). The %20 bias is estimated in Case II for both models.

In Function 5 with 16 design variables, the minimum bias is observed in Case I. In Kriging and RBF models, the bias is less than %40 with $\delta \leq 0.105$ and $\beta \in [0.5, 0.6, 0.7]$. As it is shown in Fig. 4.12. In this function, the maximum bias is estimated for Case II with higher value of the δ . Our preliminary investigations and the parametric analysis in

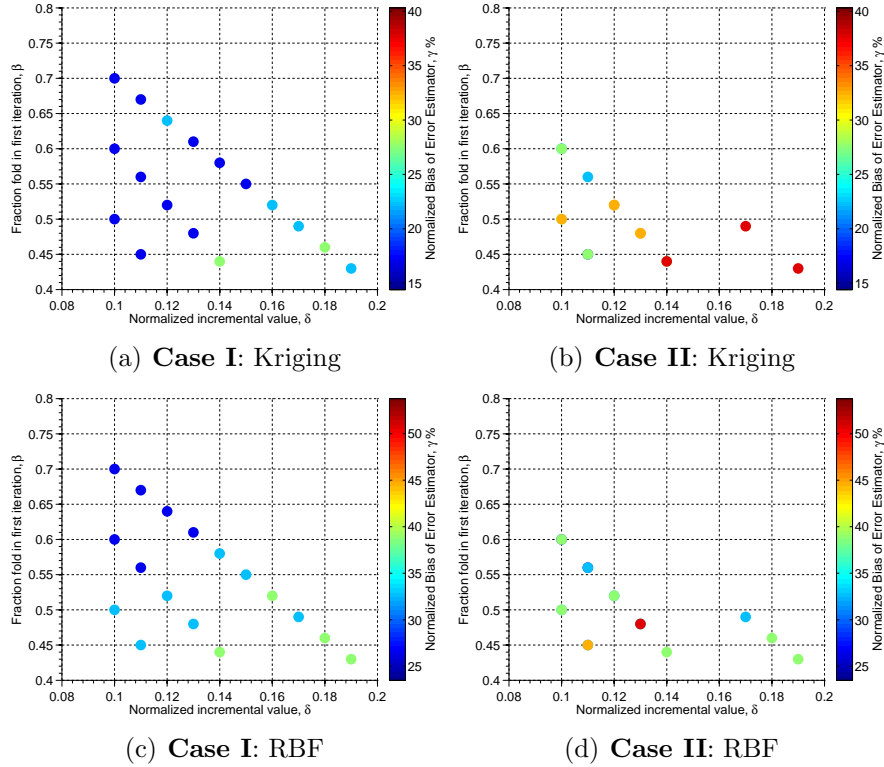


Figure 4.9: Function 2 (10D) - the normalized bias of the PEMF method in different combinations of β and δ in Case I and Case II

the results of the PEMF method by changing the regularization parameters (Fraction fold in the first iteration β and Normalized incremental value δ) shows that: (i) in most of the cases of these five test functions, the minimum bias is estimated in Case I. (ii) the effect of the δ parameter is relatively more than the β parameter, and (iii) in most of the cases we can expect the better results from PEMF with $\delta = 0.1$ and Case I.

We also compare the performance of the PEMF method with three other well-known cross-validation derived error measurement techniques, discussed previously in Chapter 2: (i) leave-one-out cross-validation, (ii) q -fold cross-validation ($q = 10$), and (iii) k -fold cross-validation ($k = 10$). This analysis will allow us to understand the relative merits and demerits of each measures. Note that, in this section we consider all the combination of PEMF regularization parameters (illustrated in Figs. 4.8 and 4.12) under the conditions

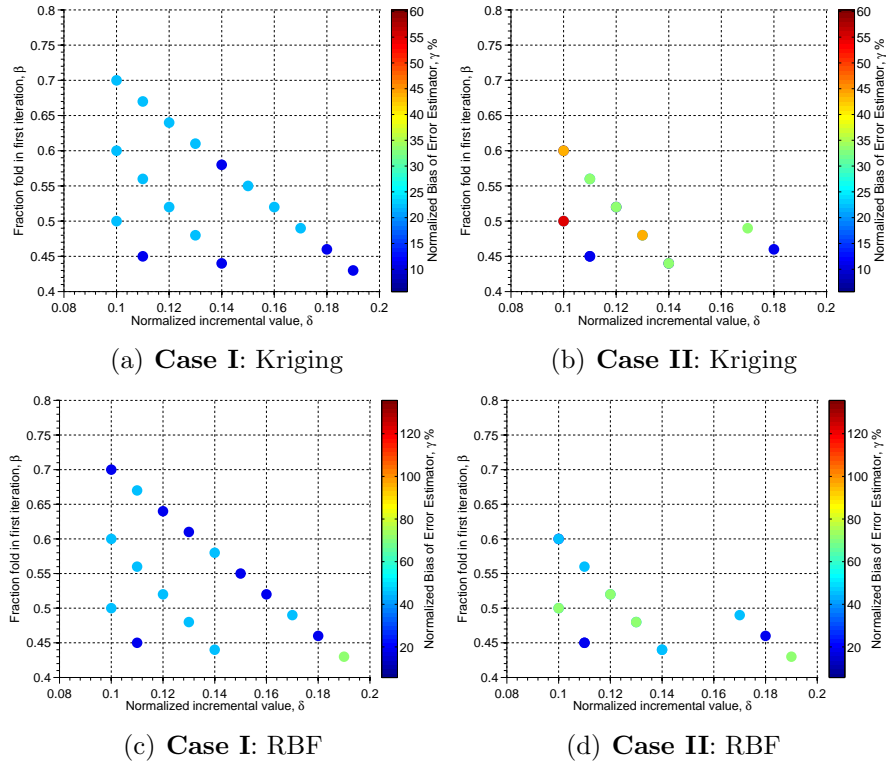


Figure 4.10: Function 3 (10D) - the normalized bias of the PEMF method in different combinations of β and δ in Case I and Case II

provided in case *I* and case *II*. The comparison results are provided in Figures 4.13 and 4.17. It is observed that, in all of the cases the results of the PEMF method is at least an order of magnitude better than those estimated using cross-validation derived error measurement techniques. These results also show that in the worst scenario of the regularization parameters selection, the results of the PEMF method are still highly accurate compared to the other three approaches.

4.4 Application of the PEMF Method on Wind Farm Power Generation Problem

In this section, the effectiveness and the impact of the PEMF method is illustrated using a wind farm power generation model, adopted from the Unrestricted Wind Farm Layout

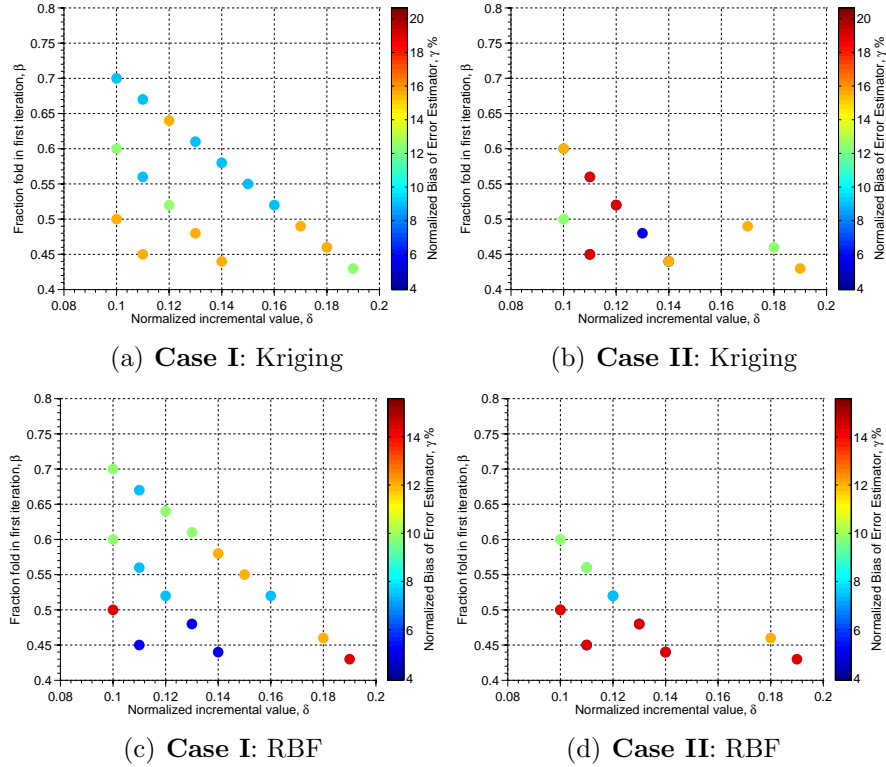


Figure 4.11: Function 4 (10D) - the normalized bias of the PEMF method in different combinations of β and δ in Case I and Case II

Optimization (UWFLO) framework[116, 117]. Surrogates are developed using Kriging, RBF, E-RBF, and QRS to represent the *power generation of an array-like wind farm*. It is assumed that turbines are arranged in a row-column pattern over the farm site. Hence, the wind farm power generation can be represented as a function of the *streamwise spacing* and *spanwise spacing* between turbines, with respect to the south direction (as shown in Fig. 4.18).

The annual average power generation of a wind farm is a complex and expensive function of the turbine features, the turbine arrangement (or farm layout), and the local wind resource variations. A surrogate model offers a more tractable (and inexpensive) representation of the farm power generation in terms of key design parameters. To train the surrogate model, the actual annual-average wind farm power generation is estimated using an advanced power generation model developed by Chowdhury et al. [116, 117]. The reliability

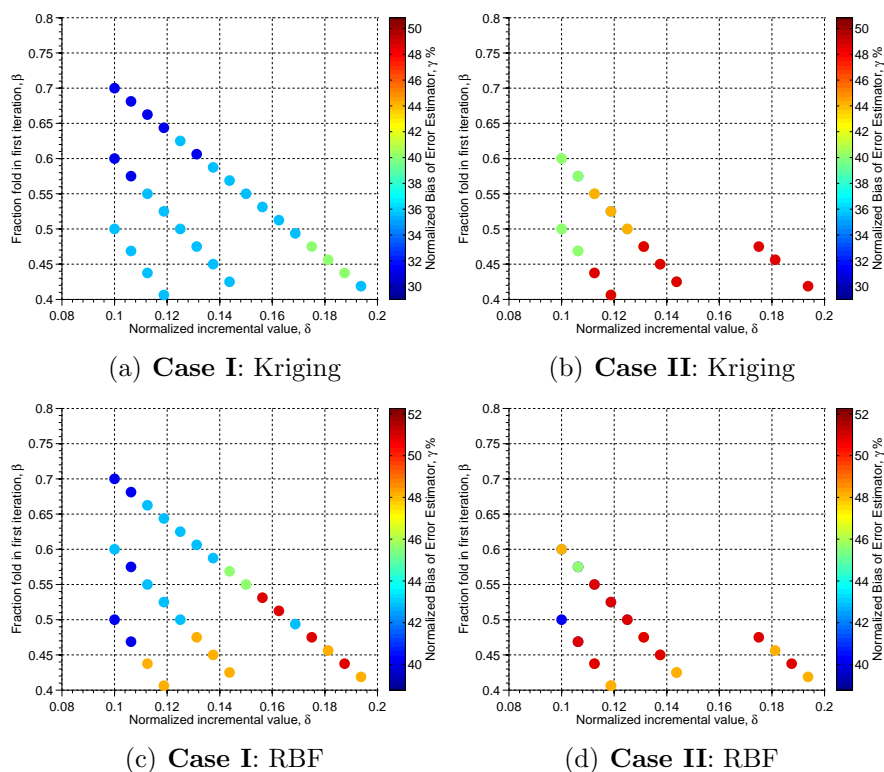


Figure 4.12: Function 5 (16D) - the normalized bias of the PEMF method in different combinations of β and δ in Case I and Case II

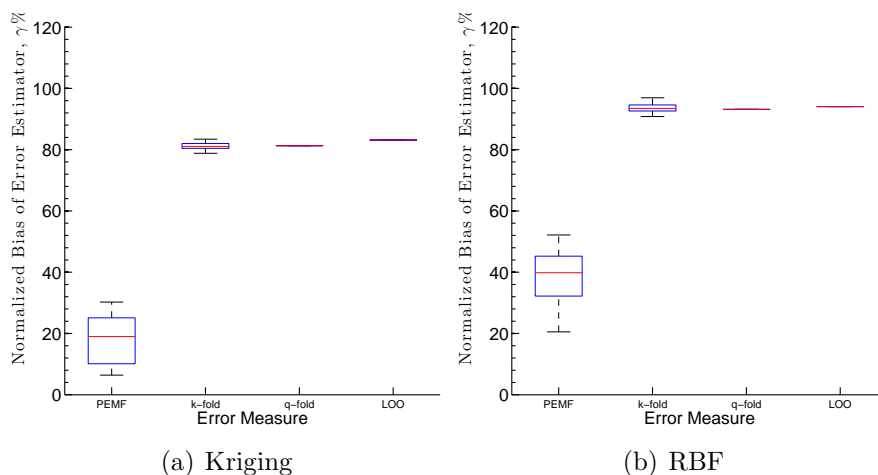


Figure 4.13: Function 1 (6D): Comparison of the PEMF and CV-derived error measures

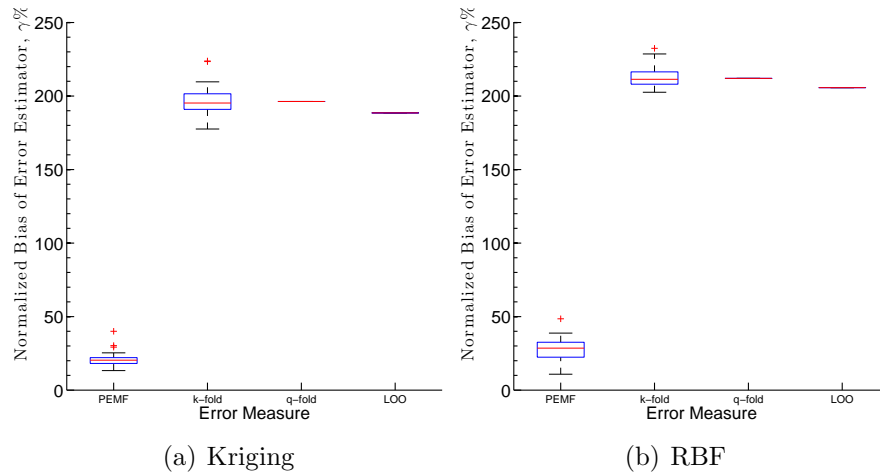


Figure 4.14: Function 2 (10D): Comparison of the PEMF and CV-derived error measures

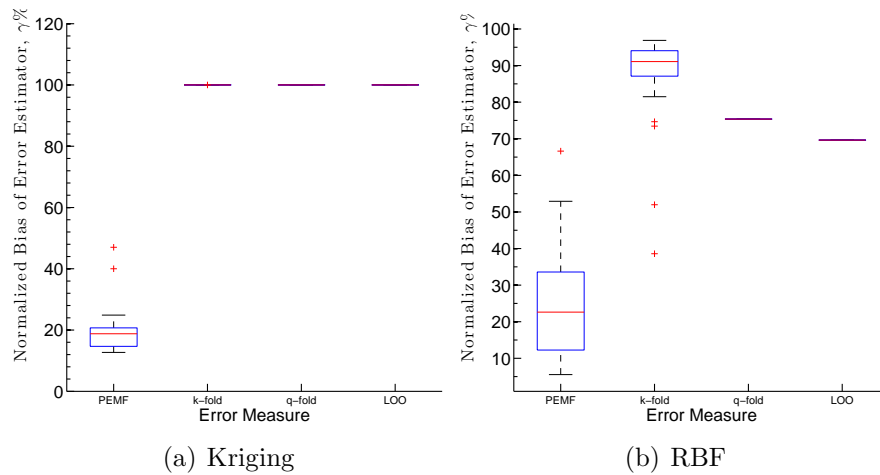


Figure 4.15: Function 3 (10D): Comparison of the PEMF and CV-derived error measures

of surrogate applications in wind farm design and analysis is subject to the accuracy of the surrogate in the concerned parameter ranges.

The PEMF method enables a unique quantification of the fidelity of the surrogate, thereby allowing more informed decision-making in wind farm design and analysis (than possible with conventional surrogate application). This case is expected to illustrate the effectiveness of the PEMF method for a complex practical engineering problem. In this problem, the surrogates are constructed to represent the power generation of an array-like

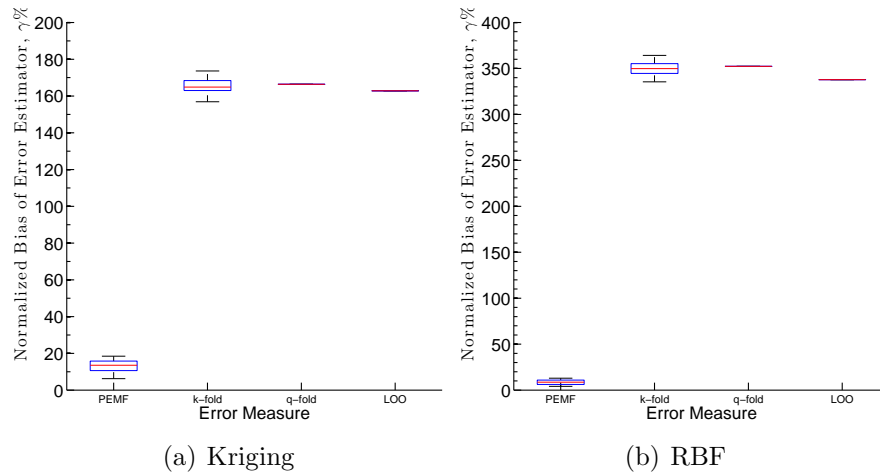


Figure 4.16: Function 4 (10D): Comparison of the PEMF and CV-derived error measures

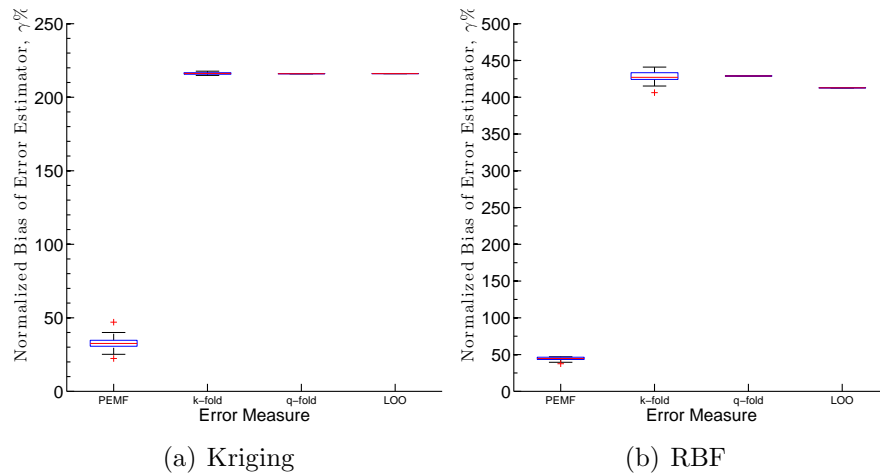


Figure 4.17: Function 5 (16D): Comparison of the PEMF and CV-derived error measures

100-turbine wind farm as a function of the *streamwise spacing* (x_h) and the *spanwise spacing* (x_l) between turbines. The turbines are arranged in a 10×10 patterns in this case. The bivariate normal distributions of wind data obtained for a site in North Dakota [118] is used for this problem. The lower and upper bounds of x_h and x_l , based on the wind turbine rotor diameter (D), are specified as

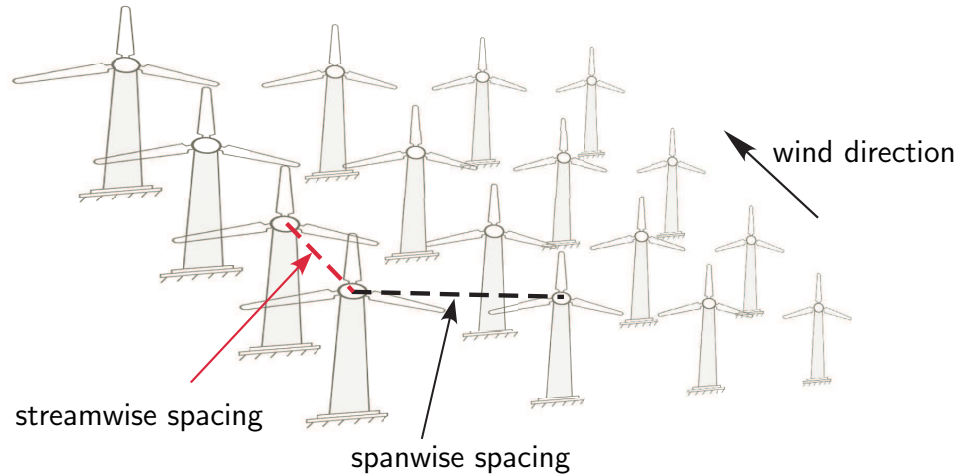


Figure 4.18: Wind farm array schematic

$$5D < x_h < 30D$$

$$1.1D < x_l < 10D$$

To predict the level of the surrogates error in the entire domain in the wind farm power generation problem, the region of interest is defined as equal to the design space ($\{X\} = \{X_{in}\}$ and $N(\{X\}) = 30$). In this problem, the size of the inside-region training points at each iteration is defined by the following ascending series; $n^t = 19, 21, 24, 27$.

The predicted levels of error estimated using PEMF and Actual (on large number of validation points) in different surrogates are shown in Fig. 4.19. This figure shows the great potential of the the PEMF method in predicting the surrogate model error. It is also observed that, Kriging has relatively higher accuracy in this problem based on the PEMF's predicted errors and actual errors evaluated on test points. This observation shows that PEMF can be effectively used as a model selection method to select the best surrogate for applications such as optimization.

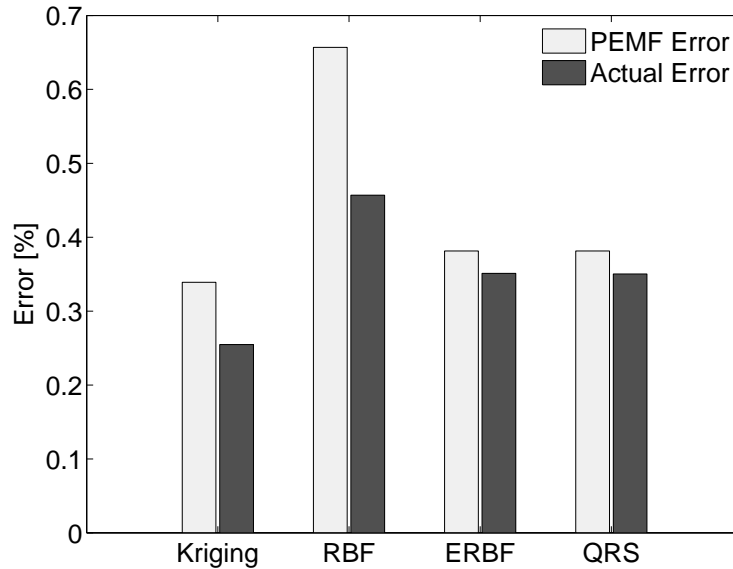


Figure 4.19: Comparison of the performance of the PEMF with Actual error in the Wind Farm Power Generation problem

The comparison results based on the performance criteria (R -values define in Chapter 3) between PEMF and leave-one-out cross-validation, are also provided in Figs. 4.20 in different surrogate models. It is observed that the PEMF method is up to two orders of magnitude more accurate than leave-one-out cross-validation in predicting levels of errors.

4.5 Chapter Summary

In this chapter, statistical testing was performed to investigate the robustness of the PEMF method in predicting regional and global errors. This test was also performed on leave-one-out cross-validation. It was observed that the PEMF method is statistically more stable in predicting model fidelity than those based on cross-validation. Considering the results of the previous chapter (Chapter 3) and the current chapter, the PEMF method can be considered the ideal model-independent approach for quantifying the fidelity of surrogate models.

In this chapter, we also explored the dependency of the PEMF method on β and δ

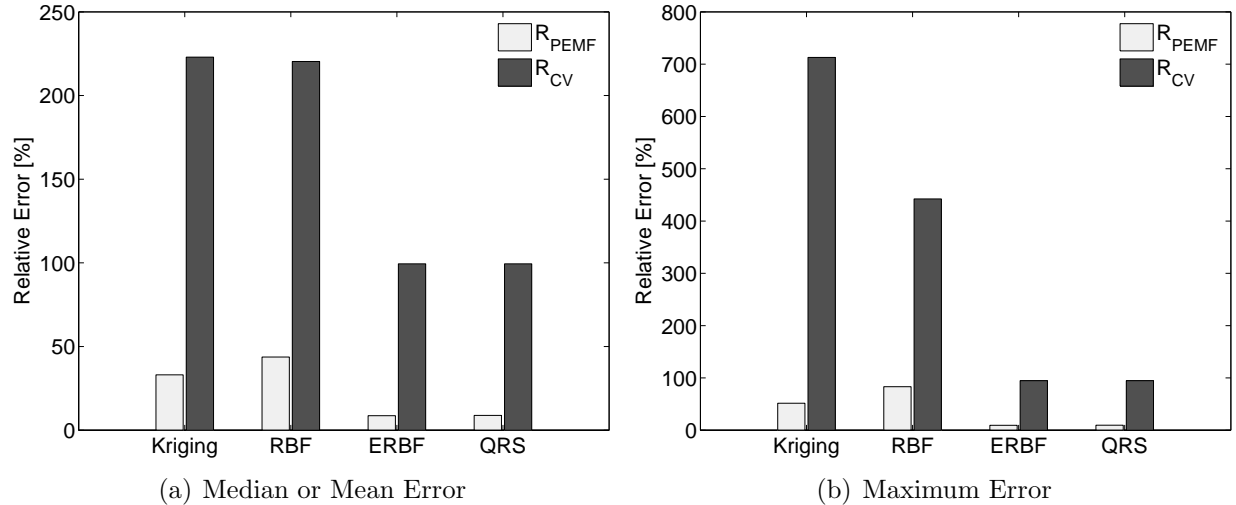


Figure 4.20: Comparison of the *PEMF* and *CV* error measures with the Actual global error in Wind Farm Power Generation problem

regularization parameters. A series of mathematical examples is used to demonstrate the effectiveness of the PEMF method under varying regularization parameters. This preliminary investigations on PEMF parameters revealed that PEMF results are not unduly sensitive to β and δ . We have also provided useful insights into the performance of three different cross-validation derived methods: (i) leave-one-out cross-validation, (ii) k -fold cross-validation, and (iii) q -fold cross-validation. In particular, we found that, for the benchmark problems considered, PEMF results are highly accurate (at least an order of magnitude) compared to the other three approaches.

PART III

Model Selection & Variable Fidelity

Optimization

CHAPTER 5

Concurrent Surrogate Model Selection (COSMOS) based on Predictive Estimation of Model Fidelity

How do we search for a globally-competitive surrogate model?

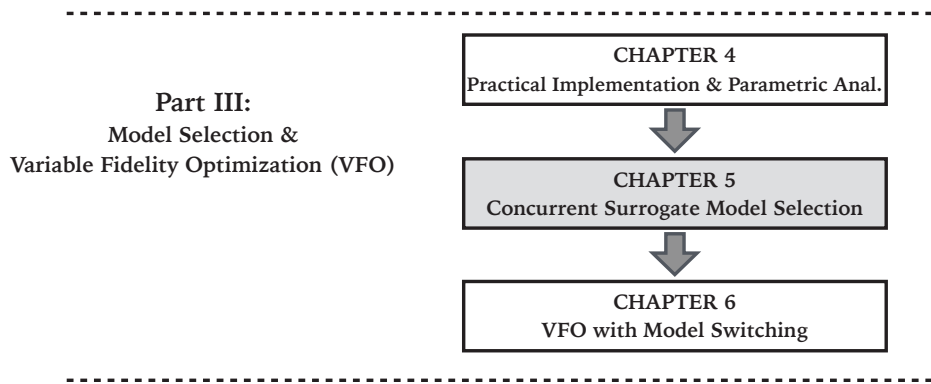


Figure 5.1: Chapter 5 orientation

The vision of this chapter is to enable the identification of globally best surrogate models for any given application, a technique that is neither limited to any particular class of models nor makes a limiting assumption regarding the surrogate/kernel type and the value of hyper-parameters. In doing so, this chapter outlines the development of a novel model selection framework to perform optimal model searching concurrently at all the three levels necessary to facilitate optimal selection, i.e., (i) optimal model type (e.g., RBF), (ii) optimal kernel type

Portions of this chapter previously appeared as:

Chowdhury, S., Mehmani, A., and Messac, A., “A Framework for Collaborative Surrogate Model Selection (COSMOS),” ASME 2014 International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE), Paper No. DETC2014-35358, Buffalo, New York, August 17-20, 2014, and

Mehmani, A., Chowdhury, S., and Messac, A., “A Novel Approach to Simultaneous Selection of Surrogate Models, Constitutive Kernels, and Hyper-parameter Values,” AIAA SciTech Conference, Paper No. 6.2014-1487, National Harbor, Maryland, January 13-17, 2014.

(e.g., *multiquadric*), and (iii) *optimal values of hyper-parameters* (e.g., *shape parameter*). This automated model selection framework is called “*Concurrent Surrogate Model Selection (COSMOS)*”.

The remainder of the chapter is organized as follows: Section 5.1 presents the overview of the Concurrent Surrogate Model Selection (COSMOS) framework; Section 5.2 presents the concept and the formulation of the COSMOS framework; The formulation of Predictive Error Estimation of Model Fidelity (PEMF) with certain modifications in COSMOS, is provided in Section 5.3. Section 5.4 describes the surrogate models used to demonstrate COSMOS, and the different benchmark problems to which COSMOS is applied; the numerical settings and results of case studies are also shown and discussed in this section. The chapter summary is provided in Section 5.6.

5.1 Chapter Introduction

The objectives of this chapter are to develop, implement, and test the effectiveness of the new *3-level automated model selection* framework called COSMOS, which uses the Predictive Estimation of Model Fidelity (**PEMF**) approach to quantify and compare the accuracy of candidate models. Two different approaches, with the same objective to perform the three level selection, are developed and investigated in this chapter. These approaches are:

- (I) *Cascaded technique*: In this approach PEMF-based selection is performed in a nested loop, with “hyper-parameter optimization” at the innermost level and “model-type selection” at the outermost level . Although simple in implementation and reliable in outcome, this approach could become computationally expensive as the number of candidate model-types and candidate kernel/basis functions increase.
- (II) *One Step technique*: In this approach, model selection is performed at *the three levels simultaneously* by formulating and solving a mixed integer nonlinear programming

(MINLP) problem that minimizes the model errors computed by PEMF. This approach is more complex and also more sensitive to the optimization strategy, compared to the *Cascaded* approach. However, the *One-Step* technique is expected to be significantly less expensive (more time efficient) and unique in implementation, and hence more universally applicable.

5.2 Development of COSMOS

5.2.1 3-Level automated model selection

The 3-level model selection is performed by minimizing the model error, where the error measure of interest depends on the type of application and the user preference. For example, when surrogate models are used for analysis (or design domain exploration) or as empirical model representing experimental data, median error could be treated as the selection criteria (e.g., power generation of a wind farm). On the other hand, in structural optimization applications that demand conservative surrogates (i.e., to model vehicle crash simulation [119]), maximum error would be a more suitable selection criteria. Similarly, if robustness is a concern, then the variance of the surrogate model error could be deemed an important selection criteria.

Depending on the problem and the available data set, these criteria might be mutually conflicting or mutually promoting, which could then respectively lead to a set of Pareto models or a single optimum model. In this chapter, these criteria are evaluated using a new advanced surrogate error estimation method called Predictive Estimation of Model Fidelity (PEMF) [120, 121]. The inputs and outputs of PEMF (in the COSMOS framework) are

illustrated in Fig. 7.2, and can be expressed as

$$[E_{med}^{mo}, E_{max}^{mo}, E_{med}^{\sigma^2}, E_{max}^{\sigma^2}, E_{med,\alpha}^{mo}] = F_{PEMF}(m, k, u, \{X\}) \quad (5.1)$$

where m , k , and u respectively represent the model type, the kernel type, and the vector of hyper-parameter values for that model-kernel combination; $\{X\}$ represents the set of sample data used for training the surrogate. In Eq. 5.1, the terms E_{med}^{mo} and E_{max}^{mo} respectively denote the predicted modal values of the median and the maximum errors in the surrogate model; the terms $E_{med}^{\sigma^2}$ and $E_{max}^{\sigma^2}$ respectively represent the variance values of the median and maximum errors in the surrogate—these two measures reflect the robustness of the surrogate. Together, these four criteria provide an understanding of the uncertainty in the surrogate.

Another selection criteria ($E_{med,\alpha}^{mo}$) formulated in this chapter is the (expected) most probable value of the median error on a hypothetically-larger sample set if the same surrogate (i.e., same m , k , and u) was constructed over α -times greater number of sample points (assuming the additional sample point would not impact the current sample distribution, $\{X\}$). This novel error measure, $E_{med,\alpha}^{mo}$, indicates the expected capability of candidate surrogates if a larger set of sample data could be made available, thereby helping users understand the cost-to-benefit ratio of performing additional high-fidelity simulations or experiments in this context. A brief description of the PEMF method in estimating these selection criteria, is provided in Section 3.2.

The two novel approaches developed to implement the 3-level model selection using PEMF, i.e., (i) *Cascaded technique*, and (ii) *One-Step technique*, are described next.

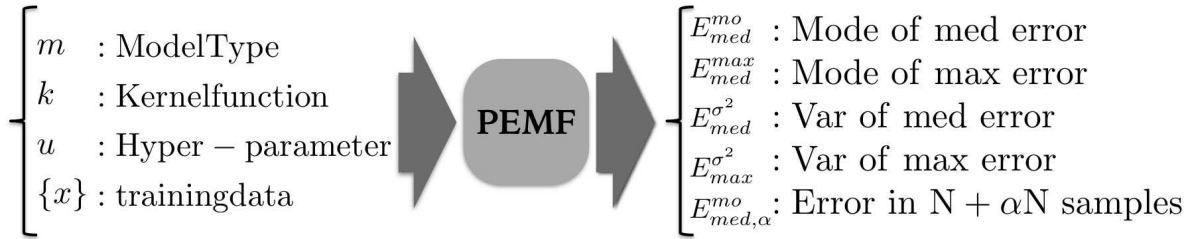


Figure 5.2: The inputs and outputs of the PEMF method

5.2.1.1 Cascaded technique

A flowchart of the 3-level *Cascaded* model selection technique is illustrated in Fig. 5.3. In this technique, for each candidate basis (or kernel) function of a candidate surrogate model, hyper-parameter optimization is performed to simultaneously minimize the five different model selection criteria. These criteria are offered by PEMF as illustrated in Fig. 7.2. For each model-kernel combination, where m denotes the *model type* and k denotes the *kernel function type*, the hyper-parameter optimization problem can be expressed as

$$\begin{aligned} & \text{Min}_{u_{mk}} \{E_{med}^{mo}, E_{max}^{mo}, E_{med}^{\sigma^2}, E_{max}^{\sigma^2}, E_{med,\alpha}^{mo}\} \\ & \text{subject to} \tag{5.2} \\ & u_{mk}^{min} \leq u_{mk} \leq u_{mk}^{max} \\ & m : \text{Model type, } m = 1, 2, \dots, N_M \\ & k : \text{Kernel function, } k = 1, 2, \dots, N_K^m \end{aligned}$$

where N_M and N_K^m respectively represent the total number of surrogate model candidates and the number of basis (or kernel) functions available for the m^{th} surrogate model; and u_{mk} is the vector of continuous variables that represent the hyper-parameter value(s) of the corresponding kernel function. Although, all five criteria functions (to be minimized) are si-

multaneously included in this optimization problem definition, in practice it is recommended to pursue bi-objective optimization – i.e., at a time considering any two objectives/criteria of interest.

The multiobjective hyper-parameter optimization can be solved using powerful heuristic algorithms, such as Evolutionary Algorithms (EA), and Particle Swarm Optimization (PSO) algorithm. In implementing the Cascaded technique, the total numbers of candidate surrogate models and constitutive Kernel types are assumed to be small enough to be directly compared, post hyper-parameter optimization of each candidate model-kernel combination. Assuming the multiobjective optimization is effective in finding the global Pareto solutions, this approach can yield reliable surrogate model choices and is practically effective when we have a small number of candidate models. An illustration of the application of the *Cascaded* technique is presented in Fig. 5.4.

Figure 5.4 presents the solutions of the hyper-parameter optimization, when using only the Support Vector Regression (SVR) model with the *Linear Kernel function*, for representing the Branin-Hoo test function. In this illustration, the median and the maximum error of the model are selected as model selection criteria. It is evident from this figure that the Pareto solutions offer helpful flexibility in deciding the best hyper-parameter (penalty parameter, C) depending on the user preference regarding the allowable error. In Fig. 5.4, the anchor solutions **A** and **C** represent the optimal hyper-parameter for the SVR surrogate with the lowest median error and the maximum error, respectively. It is also interesting to see how solution **B** offers an attractive trade-off at a small compromise in both the median and the maximum errors in comparison to the respective anchor points.

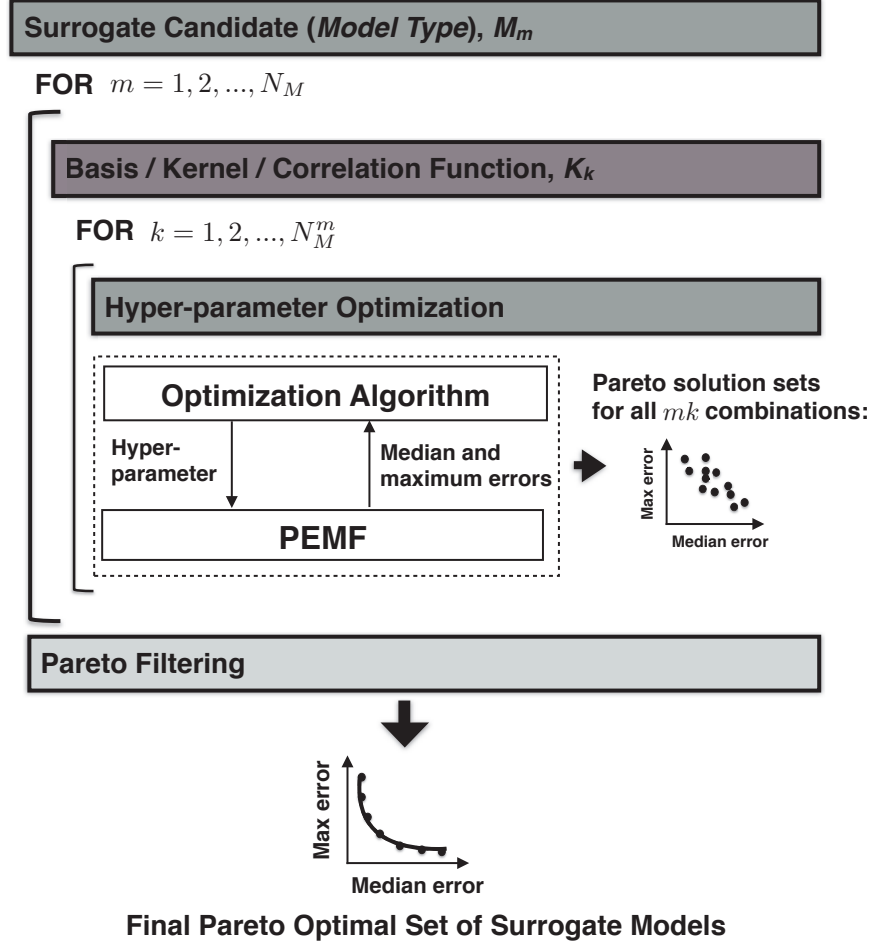


Figure 5.3: The 3-level Cascaded model selection technique based on PEMF

The computational time of the 3-level *Cascaded* model selection technique is given by

$$T = \sum_{m=1}^{N_M} \sum_{k=1}^{N_K^m} t_{mk} \quad (5.3)$$

where t_{mk} is the computational time invested in the hyper-parameter optimization for the k^{th} kernel of the m^{th} surrogate model.

In this context, it is important to note that each function call in a hyper-parameter optimization involves the error quantification of a candidate model, which involves training multiple intermediate models in PEMF. As a result, the computational time of each multi-

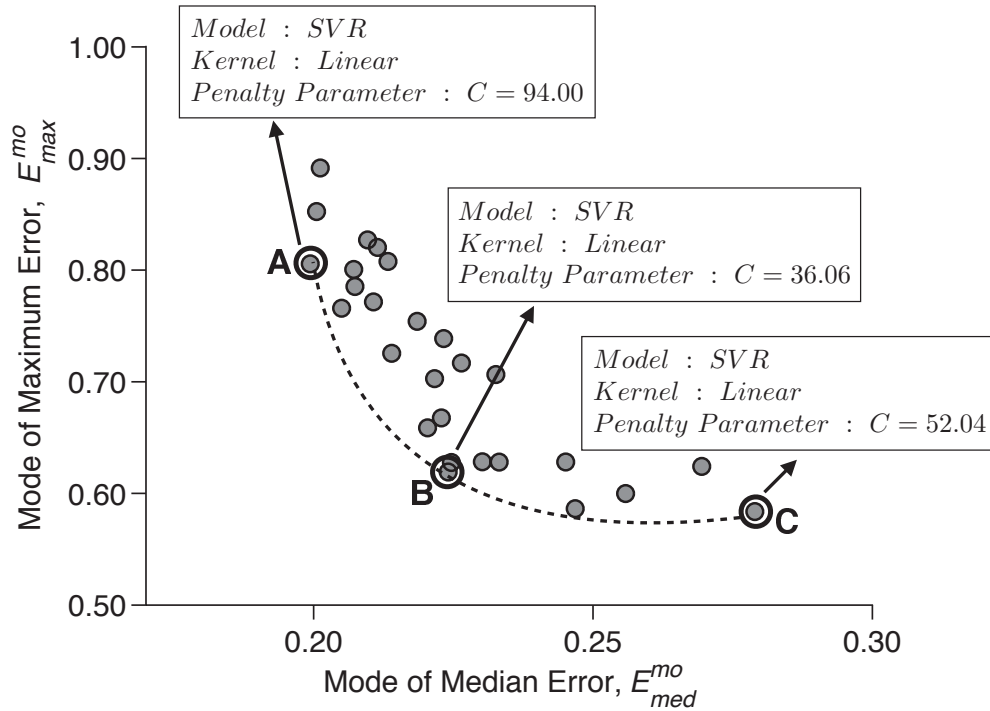


Figure 5.4: Optimal solutions given by the multiobjective hyper-parameter optimization with the SVR-Linear model for the Baranin-Hoo test function

objective hyper-parameter optimization could be significant. The *Cascaded* approach could thus turn out to be expensive if a large number of surrogate model candidates and Kernel candidates are considered (i.e., $N_M, N_K \gg 1$).

5.2.1.2 One-Step technique

To escape the potentially high computational cost of the *Cascaded* technique, the three-level automated model selection could also be performed by solving a single (uniquely formulated) mixed integer nonlinear programming (MINLP) problem. The major components and the flow of information in the *One-Step technique* is illustrated in Fig. 5.5. The general form of this MINLP problem can be expressed as

$$\begin{aligned}
& \text{Min}_{m,k,u} \{E_{med}^{mo}, E_{max}^{mo}, E_{med}^{\sigma^2}, E_{max}^{\sigma^2}, E_{med,\alpha}^{mo}\} \\
& \text{subject to} \\
& m \leq N_M, m \in \mathbb{Z}_{>0} \\
& k \leq N_K(m), k \in \mathbb{Z}_{>0} \\
& u = [u_{11} \ u_{12} \ \dots \ u_{21} \ u_{22} \ \dots \ u_{mk} \ \dots \ u_{N_M N_K}] \\
& u_{mk}^{min} \leq u_{mk} \leq u_{mk}^{max}
\end{aligned} \tag{5.4}$$

In Eq. 5.4, m and k are integer design variables that denote the model type and the corresponding kernel function type, respectively; and u_{mk} is the vector of continuous variables that represent the hyper-parameter values for the k^{th} kernel of the m^{th} candidate surrogate, with u_{mk}^{min} and u_{mk}^{max} as the vectors of specified lower and upper bounds of the hyper-parameters. Direct solution of the MINLP problem (Eq. 5.4) could be cumbersome when models with different numbers of candidate kernel choices, and kernels comprising differing numbers of hyper-parameters are considered. Since, the variation in the number of hyper-parameter (HP) among surrogate kernels is quite small ($0 \leq N_{HP} \leq 2$), it is deemed not practically worth to formulate a complex optimization that considers differing number of hyper-parameters simultaneously. Instead, the global pool of model-kernel candidates is divided into three smaller pool of model-kernel candidates based on the number of constituent hyper-parameters in them. Each candidate pool, represented by Φ_p , comprises all those model-kernel combinations which include \mathbf{p} hyper-parameter(s). Subsequently, optimal model selection is performed separately (in parallel) for each candidate pool. Each model-

Table 5.1: Candidate model-kernel combinations and their integer-codes

Surrogate	Kernel	Φ_0	Φ_1	Φ_2	Hyper-Parameter(s)
Radial Basis Function	Linear	1	-	-	-
	Cubic	2	-	-	-
	Gaussian	-	1	-	Shape parameter, σ
	Multiquadric	-	2	-	Shape parameter, σ
Kriging	Linear	-	3	-	Correlation parameter, θ
	Exponential	-	4	-	Correlation parameter, θ
	Gaussian	-	5	-	Correlation parameter, θ
	Spherical	-	6	-	Correlation parameter, θ
Support Vector Regression	Linear	-	7	-	Penalty parameter, C
	Gaussian	-	-	1	Kernel parameter, γ and Penalty parameter, C
	Sigmoid	-	-	2	Kernel parameter, γ and Penalty parameter, C

kernel combination/candidate within a particular candidate pool (Φ_p) is then assigned a single unique integer code, as opposed to two separate integer codes, as given by Eq. 5.4. The candidate model-kernels considered in this dissertation are listed in Table 5.1, where the integer code assigned to each candidate is shown under their respective hyper-parameter class (Φ_p).

For the Φ_0 class of model-kernel combinations, PEMF is applied to all the candidates, followed by the application of a Pareto filter to determine the final set of non-dominated or Pareto optimal surrogate models. For all Φ_p with $p > 0$, the mixed integer non-linear programming (MINLP) problem (Eq. 5.4) is reformulated as described in Eq. 5.5

$$\begin{aligned}
& \text{Min}_{z,u} \{ E_{med}^{mo}, E_{max}^{mo}, E_{med}^{\sigma^2}, E_{max}^{\sigma^2}, E_{med,\alpha}^{mo} \} \\
& \text{subject to} \\
& z \leq N(\Phi_p), z \in \mathbb{Z}_{>0} \\
& 0 \leq u \leq 1
\end{aligned} \tag{5.5}$$

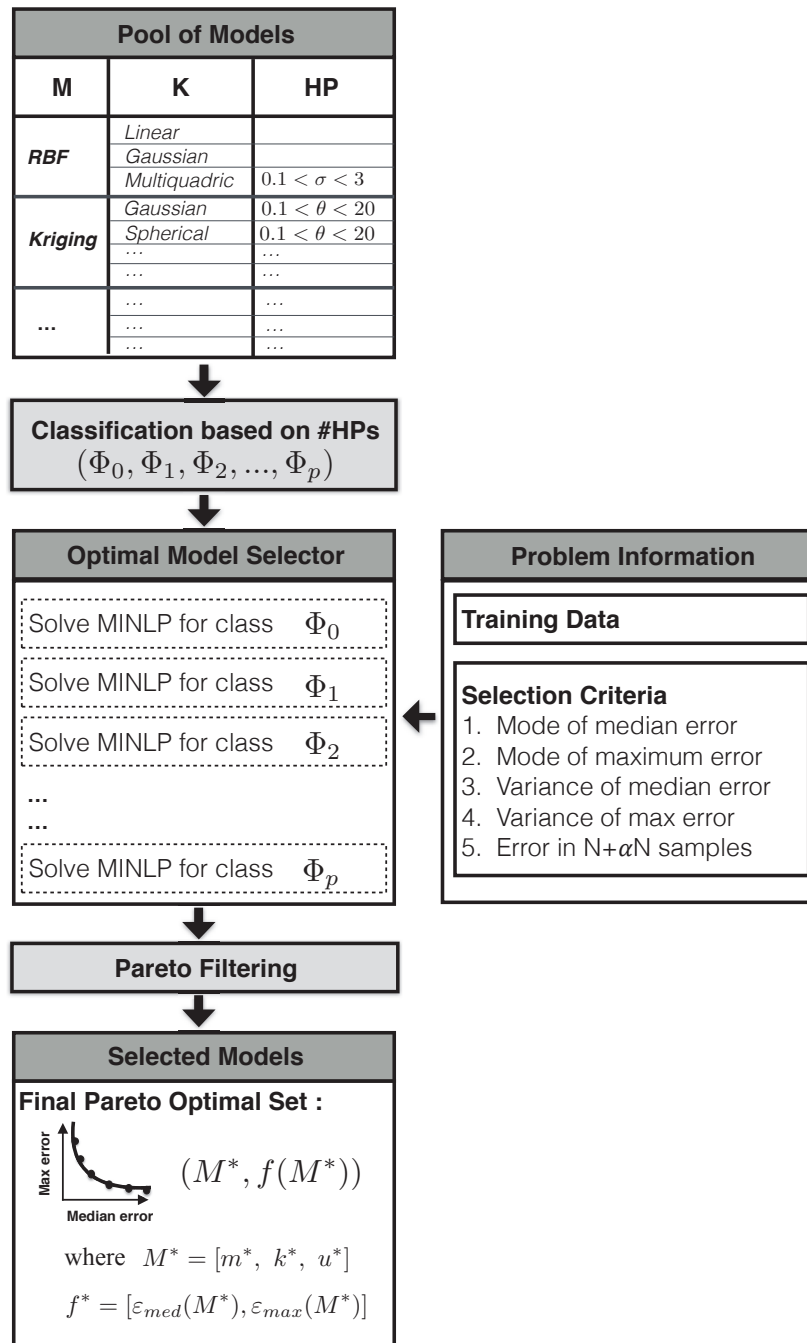


Figure 5.5: The 3-level *One-Step* model selection technique (M: Surrogate model type; K: Kernel function type; HP: Hyper-parameter; u : vector of Hyper-parameter values)

Table 5.2: Range of hyper-parameters

Surrogate	Hyper-parameter	Lower bound	Upper bound
RBF	Shape parameter, σ	0.1	3.0
Kriging	Correlation parameter, θ	0.1	20
SVR	Kernel width parameter, γ	0.1	10
SVR	Penalty parameter, C	0.1	100

In Eq. 5.5, z is the integer variable that denotes the combined model-kernel type; u is the vector of continuous variables that represent the hyper-parameter values; and $N(\Phi_p)$ represents the size of the set of Φ_p , which is the total number of candidate model-kernel types available under the p^{th} hyper-parameter class (Φ_p). It should be noted that a consistent range of (0,1) is used for each hyper-parameter where the hyper-parameters are scaled based on the user-specified upper and lower bounds. The upper and lower bounds on the hyper-parameters used in this dissertation, as listed in Table 5.2, are motivated by recommendations in the existing literature [70, 112, 122, 123]. The normalized value of the Hyper parameters are unscaled before being used in the *PEMF* method, as given by

$$\bar{u} = u^{min} + (u^{max} - u^{min}) \times u \quad (5.6)$$

where \bar{u} and u are the unscaled and scaled values of a hyper-parameter, respectively; and u^{min} and u^{max} are respectively the user-specified lower and upper bounds on the hyper-parameter value (u). It should be noted that all five model selection criteria offered by *PEMF* are included in the optimization formulation of the *One-Step technique*, however, in practice it is again recommended to pursue bi-objective optimization.

Once the Pareto optimal surrogate models for each Φ_p have been obtained, a Pareto

filter is applied to the collective set of optimal models from all Φ_p to determine the globally optimal set of surrogate models. The criteria functions in this MINLP problem could often be multimodal, since the integer codes of the model-kernel types do not necessarily follow the order of error in these candidate types. Hence, heuristic multi-objective optimization algorithms that are also capable of dealing with mixed-integer variables are considered to be an appropriate choice for solving these MINLP problems. In this dissertation, the binary Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [124] is used to solve the multi-objective MINLP problems. In addition to the set of Pareto optimal surrogate models, i.e., the optimal model-kernel types and the corresponding optimal hyper-parameter values ($M^* = [m^*, k^*, u^*]$), this technique also provides the estimated error of the optimal surrogates.

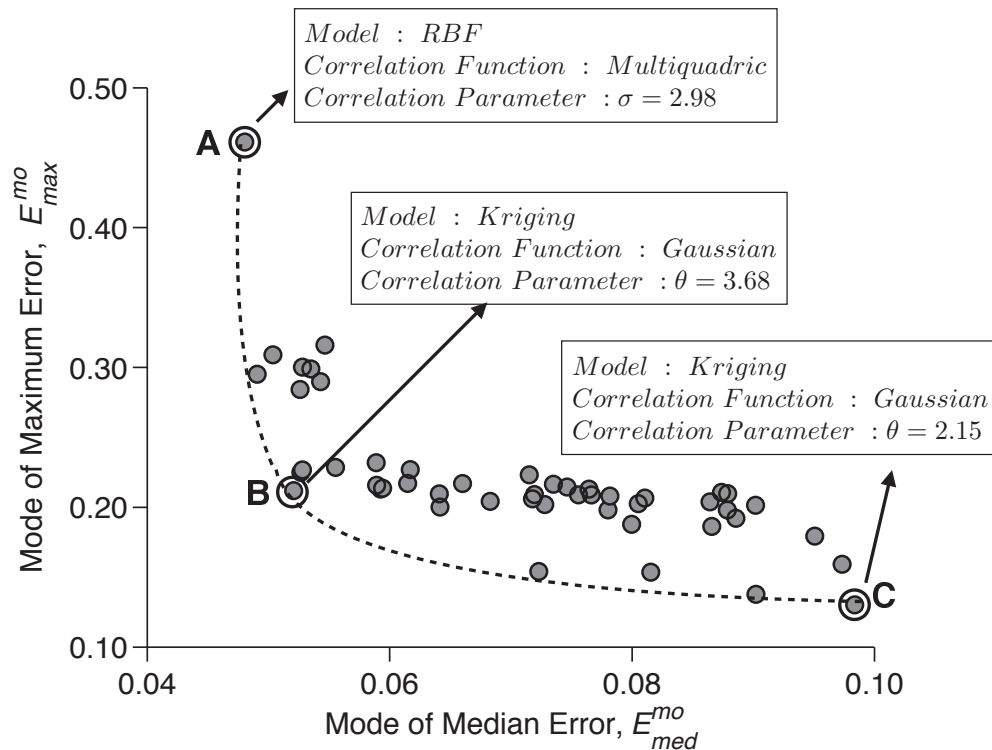


Figure 5.6: Optimal solutions given by the simultaneous model selection using One-Step technique for the Baranin-Hoo test function

An illustration of the selection of the best trade-off surrogates for the Baranin-Hoo test function using the *One-Step technique*, when the median and the maximum errors are used as selection criteria, is provided in Fig. 5.6. Three types of surrogate models with multiple Kernels function choices for each candidate model are considered in this illustrative problem. A detailed description of this problem is expressed in Section 5.4. In Fig.5.6, the surrogate model (A) with the lowest median error is the one that is constructed using *RBF* with the *Multiquadric basis function*, and a shape parameter, $\sigma = 2.98$. The surrogate model (C) with the lowest maximum error is the one that is constructed using *Kriging* with the *Gaussian correlation function*, and a hyper-parameter, $\theta = 2.15$. The error measure (PEMF) used to quantify model accuracy and compare surrogates in the model selection techniques is discussed next.

5.3 Predictive Estimation of Model Fidelity (PEMF) in COSMOS framework

In the COSMOS framework, a comprehensive set of five different selection criteria can be specified (in any combination) to drive the surrogate model selection. The first two criteria (E_{med}^{mo} , E_{max}^{mo}) respectively represent the (predicted) most probable values of the median error and the maximum error in the surrogate – which stochastically accounts for the sensitivity of the surrogate accuracy to the choice (number and location) of training points. Advancements are made to PEMF in this dissertation to formulate and use three other selection criteria. The 3rd and the 4th criteria ($E_{med}^{\sigma^2}$, $E_{max}^{\sigma^2}$) respectively represent the variance in the median and the maximum error of the surrogate model – these two measures reflect the robustness of the surrogate. Together, these four criteria provide an understanding of the uncertainty in the

surrogate. Another new selection criteria (5th criterion: $E_{med,\alpha}^{mo}$) formulated in this dissertation is the (expected) most probable value of the median error on a hypothetically-larger sample set, assuming both the actual and the hypothetically larger sets of sample points follow the same overall distribution.

In this chapter, the PEMF method (Chapter 3) is applied to estimate all the five selection criteria for any given candidate surrogate. In the PEMF method, for a set of N sample points, intermediate surrogates are constructed at each iteration, t , using M^t heuristic subsets of n^t training points (called intermediate training points). These intermediate surrogates are then tested over the corresponding remaining $N - n^t$ points (called intermediate test points). The median error is then estimated for each of the M^t intermediate surrogates at that iteration, and a parametric probability distribution is fitted to yield the modal value, $E_{med}^{mo,t}$. The smart use of the modal value of the median error promotes a monotonic variation of error with sample point density, unlike mean or root mean squared error which are highly susceptible to outliers [120]. This approach gives PEMF an important advantage over conventional cross-validation-based error measures, as illustrated by Mehmani et al. [120, 121]. A similar approach is used to estimate the modal value of the maximum error ($E_{max}^{mo,t}$) at any t^{th} iteration.

In the original PEMF method, the distribution functions to be fitted over the median and the maximum errors at each iteration were selected using the chi-square goodness-of-fit criterion [110]. The following distributions were considered: lognormal, Gamma, Weibull, logistic, log logistic, t-location scale, inverse gaussian, and generalized extreme value distribution. However, in order to control the computational expense of PEMF within model selection, only the lognormal distribution is used. This distribution has been previously observed (from numerical experiments) to be effective in general. The PDFs of the median

and the maximum errors, p_{med} and p_{max} , can thus be expressed as

$$\begin{aligned} p_{med} &= \frac{1}{E_{med}\sigma_{med}\sqrt{2\pi}} \exp\left(\frac{(\ln(E_{med} - \mu_{med}))^2}{2\sigma_{med}^2}\right) \\ p_{max} &= \frac{1}{E_{max}\sigma_{max}\sqrt{2\pi}} \exp\left(\frac{(\ln(E_{max} - \mu_{max}))^2}{2\sigma_{max}^2}\right) \end{aligned} \quad (5.7)$$

In the above equations, E_{med} and E_{max} respectively represent the median and the maximum relative absolute errors estimated over a heuristic subset of training points at any given iteration in PEMF. The parameters, $(\mu_{med}, \sigma_{med})$ and $(\mu_{max}, \sigma_{max})$ represent the generic parameters of the lognormal distribution. The modal values of the median and the maximum errors at any iteration, t , can then be expressed as

$$\begin{aligned} E_{med}^{mo}|_t &= \exp(\mu_{med} - \sigma_{med}^2)|_t \\ E_{max}^{mo}|_t &= \exp(\mu_{max} - \sigma_{max}^2)|_t \end{aligned} \quad (5.8)$$

The variance of the median and the maximum errors at any iteration, t , can then be expressed as

$$\begin{aligned} E_{med}^{\sigma^2}|_t &= (\exp(\sigma_{med}^2) - 1) \exp(2\mu_{med} + \sigma_{med}^2)|_t \\ E_{max}^{\sigma^2}|_t &= (\exp(\sigma_{max}^2) - 1) \exp(2\mu_{max} + \sigma_{max}^2)|_t \end{aligned} \quad (5.9)$$

Once we have the history of median and maximum errors at different sample size ($< N$), the variation of the modal values and the variance of the errors with sample density are then modeled using the multiplicative ($E = a_0 n^{a_1}$) or the exponential ($E = a_0 e^{a_1 n}$) regression functions. Alternatively, if the type of median/maximum error distribution is fixed (e.g., lognormal distribution with parameters μ and σ), standard regression functions

could also be used to instead model the variation of the parameters of the distribution as functions of the number of training points. Since the variance in the errors could both increase or decrease with increasing number of sample points (for practical sample sizes), other regression functions (e.g., linear and polynomial functions) are allowed to be used to fit the variation of $E_{med}^{\sigma^2}$ and $E_{max}^{\sigma^2}$. The root mean squared error metric is used to select the best-fit regression model. These regression functions are then used to predict the modal values and the variance of the median and the maximum errors in the final surrogate, where the final surrogate is trained using all the N sample points.

The PEMF error measures promote a monotonic decrease of the median and maximum errors with sample density 3. Since the rate of error decrease (given by the regression functions) is likely different for different surrogates, it is possible that the order of dominance among candidate surrogates may change if a higher number of sample points were made available – this scenario is uniquely accounted for by the 5th selection criteria in COSMOS. Since the modal value of the median error at $N + \alpha N$ sample points (the 5th selection criteria: $E_{med,\alpha}^{mo}$) is predicted by further extrapolation of the regression function, the specified value of α should be small (lower than 0.3 or 30%).

5.4 Numerical Experiments with COSMOS

The effectiveness of the new model selection framework (**COSMOS**) is investigated by considering a pool of models comprised of the following three popular model types: (i) Kriging, (ii) RBF, and (iii) SVR. In this chapter four benchmark problems are used as case studies for this investigation. The results are compared with model selection based on the actual error estimated on a large set of additional test points (for validation). The different forms of the kernel/basis/correlation functions currently considered in COSMOS are given

in Table 5.3. It could be said that COSMOS is one of the most comprehensive surrogate model selection framework both in methodology (3-level selection) and implementation (i.e., in terms of the pool of candidate model-kernels considered). The candidate surrogate models and the different forms of the kernel/basis/correlation functions currently considered in COSMOS are described in Chapter 2. COSMOS offers five different criteria for selection of optimal surrogates. Any combination of these criteria can be chosen by the user. In the current implementation, three of the most practically pertinent pairwise combinations ($E_{med}^{mo} - E_{max}^{mo}$, $E_{med}^{mo} - E_{med}^{\sigma^2}$, and $E_{med}^{mo} - E_{med,\alpha}^{mo}$) are allowed to be selected. Using more than two criteria could prove to be taxing on the complex multiobjective optimal model selection process, and is also an unlikely scenario in practice.

Table 5.3: Basis or Kernel functions and their hyper-parameters in the candidate surrogate models

Type of model	Type of basis/correlation/kernel function	Hyper-parameter
RBF	Linear: r	shape parameter, σ
	Cubic: r^3	
	Gaussian: $e^{(-r^2/2\sigma)}$	
	Multiquadric: $(r^2 + \sigma^2)^{1/2}$	
Kriging	Linear: $\max(1 - \theta r, 1)$	correlation function parameter, θ
	Exponential: $e^{(-\theta r)}$	
	Gaussian: $e^{(-\theta r)^2}$	
	Spherical: $1 - 3\xi^2 + 2\xi^3$; $\xi = \max(1 - \theta r, 1)$	
SVR	Linear: $(x_i^T x_j)$	kernel parameters; γ , r , and d
	Gaussian: $e^{-\gamma \ x_i - x_j\ ^2}$, $\gamma > 0$	
	Sigmoid: $\tanh(\gamma x_i^T x_j + r)$	

The effectiveness of the COSMOS framework is illustrated by performing model selection for the following four analytical benchmark functions: (i) the 2-variable Branin-Hoo function, (ii) the 3-variable Hartmann function, (iii) the 10-variable Perm function, and (iv) the 50-variable Dixon & Price function. These functions have been widely used for investigating new surrogate models and other related research [51, 125]. The equations of these

four test functions are given below:

Branin-Hoo function (2 variables):

$$f(x) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos(x_1) + 10 \quad (5.10)$$

where $x_1 \in [-5 \ 10]$, $x_2 \in [0 \ 15]$

Hartmann function (3 variables):

$$f(x) = - \sum_{i=1}^4 c_i \exp \left\{ - \sum_{j=1}^n A_{ij} (x_j - P_{ij})^2 \right\} \quad (5.11)$$

where $x = (x_1 \ x_2 \ \dots \ x_n)$ $x_i \in [0 \ 1]$

In this function, the number of variables, $n = 3$; the constants c , A , and P , are respectively a 1×4 vector, a 4×3 matrix, and a 4×3 matrix.

Perm Function (10 variables):

$$f(x) = \sum_{k=1}^n \left\{ \sum_{j=1}^k (j^k + 0.5) \left[\left(\frac{x_j}{j} \right)^k - 1 \right] \right\}^2 \quad (5.12)$$

where $x_i \in [-n \ n + 1]$, $i = 1, \dots, n$

$$n = 10$$

Dixon & Price Function (50 variables)

$$f(x) = (x_1 - 1)^2 + \sum_{i=2}^n i (2x_i^2 - x_{i-1})^2 \quad (5.13)$$

where $x_i \in [-10 \ 10]$, $i = 1, \dots, n$

$$n = 50$$

5.5 Results and Discussion

Table 5.4: Problem properties and optimization settings

Problem	Problem Properties		Optimization Settings		PEMF Settings	
	Dimension	Sample Size	Population Size	Maximum No. of Generations	No. of iterations, N^{it}	No. of training points at each iteration, n^t
Branin-Hoo	2	30	30	60	4	$19+2t$
Hartmann-3	3	30	30	60	4	$19+2t$
Perm	10	50	50	60	4	$30+4t$
Dixon & Price	50	200	50	60	4	$150+10t$

The sample size used for training the surrogates and the settings specified for optimal model selection in each application are summarized in Table 5.4. The numerical settings for the implementation of *PEMF* in the *COSMOS framework* are also provided in Table 5.4, which lists (i) the number of iterations, and (iv) the size of training points at each iteration (as a function of iteration). The Optimal Latin Hypercube based on Translational Propagation algorithm [126] is adopted to determine the locations of the full set of training points $\{X\}$, and the set of additional test points used for validation in the benchmark problems. In this section, the Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) [124] is applied to perform the hyper-parameter optimization in the *Cascaded* technique, and to solve the mixed integer nonlinear programming problem formulated in the *One-Step* technique.

Three different model selection tests are performed using COSMOS, each involving a unique pair of selection criteria, as listed below:

TEST I: Apply COSMOS to (i) minimize the modal value of the median error E_{med}^{mo} , and (ii) minimize the modal value of the maximum error E_{max}^{mo} ;

TEST II: Apply COSMOS to (i) minimize the modal value of the median error E_{med}^{mo} , and (ii) minimize the standard deviation of the median error $E_{med}^{\sigma^2}$;

TEST III: Apply COSMOS to (i) minimize the modal value of the median error E_{med}^{mo} , and (ii) minimize the expected modal value of the median error at 20% more number of sample points $E_{med,\alpha}^{mo}$, $\alpha = 0.2$;

TEST I: $(E_{med}^{mo} - E_{max}^{mo})$

In this test, for validation purposes, the results yielded by the *COSMOS framework* are compared with those given by using the actual median and the maximum errors evaluated on a large set of additional test points ($50 \times$ Number of Training Points). To perform a fair comparison, the estimated median and maximum errors on heuristic subsets of additional test points are also used to fit error distributions. The statistical mode of the median and the maximum error distributions are then used as the selection criteria. The method used to estimate the actual median and the maximum errors is presented as a pseudocode in Algorithm 2 in the Chapter 3.

Figure 5.7 illustrates the best trade-offs between the median and the maximum error estimated using PEMF obtained through hyper-parameter optimizations of each model-kernel combinations for the Branin-Hoo test function. To select the best model using the *Cascaded* technique, the hyper-parameter optimization results are added together and a Pareto filter is then applied to determine the final Pareto set of models.

The final optimal solutions, including the best trade-offs between the median and the maximum errors (in the Φ_0 , Φ_1 , and Φ_2 classes) for the six test problems are illustrated

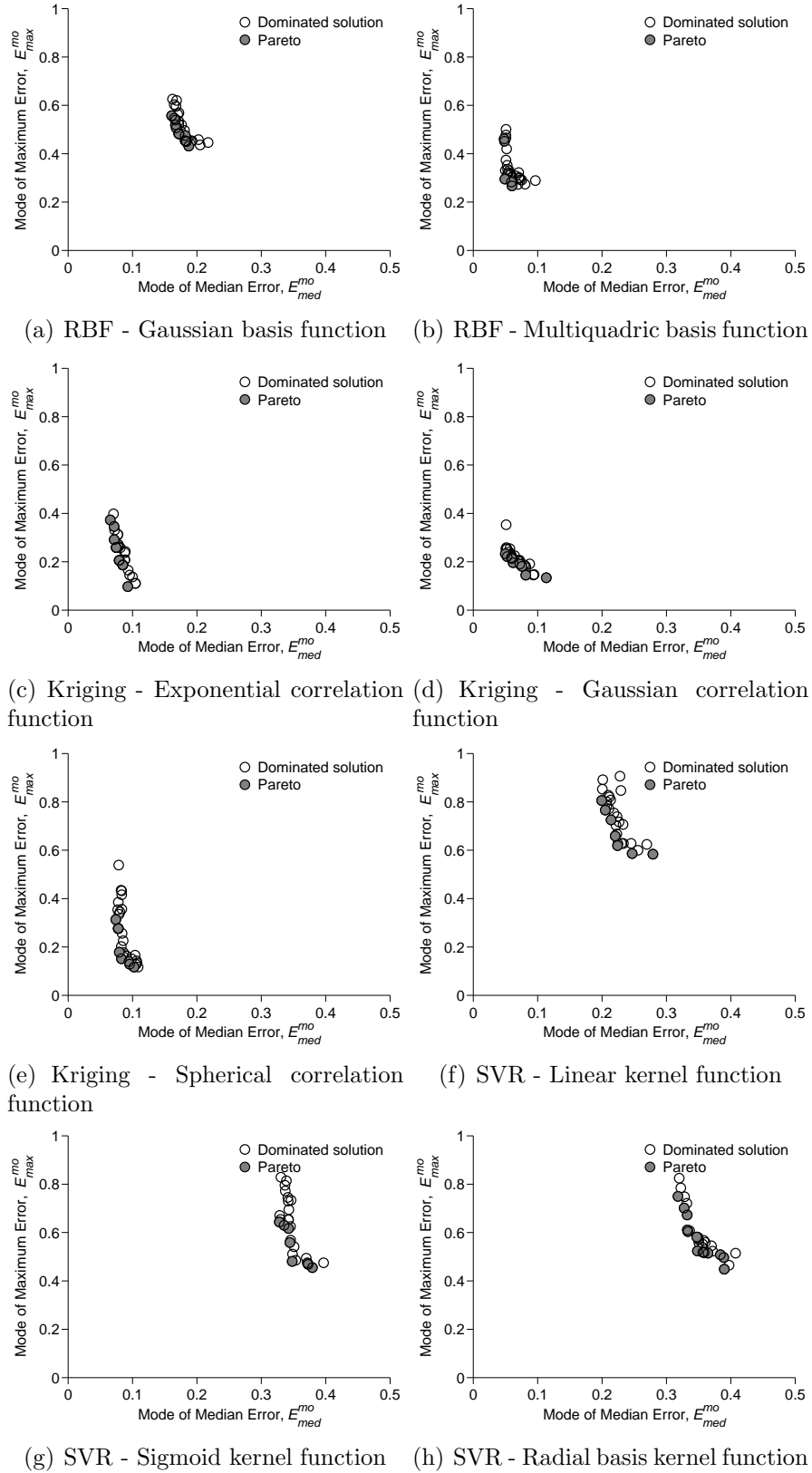


Figure 5.7: Application of COSMOS in surrogate constructed for Branin-Hoo Function

Table 5.5: The set of Pareto Surrogate Models given by (i) Cascaded technique, (ii) One-Step technique, and (iii) Model selection based on actual error

Problem	Cascaded technique	One-step technique	Actual error
Branin-Hoo	- RBF (Multi-quadric) - <i>Kriging (Gaussian)</i>	- RBF (Multi-quadric) - <i>Kriging (Gaussian)</i>	- <i>RBF (Multi-quadric)</i> - Kriging (Gaussian) - RBF (Cubic)
Hartmann-3	- <i>RBF (Gaussian)</i> - RBF (Multiquadric) - Kriging (Gaussian)	- <i>RBF (Gaussian)</i> - RBF (Multiquadric) - Kriging (Gaussian)	- <i>RBF (Gaussian)</i> - Kriging (Gaussian)
Perm	-	- RBF (Linear) - RBF (Multiquadric) - Kriging (Linear) - <i>SVR (Sigmoid)</i>	- RBF (Linear) - RBF (Multiquadric) - Kriging (Linear) - <i>SVR (Sigmoid)</i> - Kriging (Gaussian)
Dixon & Price	-	- RBF (Linear) - <i>RBF (Cubic)</i>	- <i>RBF (Cubic)</i>

in Figs. 5.8, 7.7, and 5.11-5.12. Each figure shows the results from a particular test for a particular problem. In each of these figures, the 'O', the 'X', and the '+' symbols respectively represent the Pareto solutions of model selection under Φ_0 , Φ_1 , and Φ_2 classes (i.e., different hyper-parameter classes). The final Pareto optimal solutions in these figures are encircled by square symbols, and are derived by applying a Pareto Filter.

The Pareto optimal surrogate models obtained by the *Cascaded* and the *One-Step* techniques are listed in Table 5.5. The surrogate models selected based on the actual error for each problem are also provided in Table 5.5 for validation purposes. For each of the test problems, the Pareto optimal solutions with the lowest values of the median error and the maximum error are respectively written in **boldface** and *italic*. In the practical design problems (i.e., the Airfoil design and Three-pane window problems), the model selection based on the actual error estimated using additional test points is not provided due to the high computational cost.

It is observed from Table 5.5 that in both the *Cascaded* and the *One-Step* techniques for the Branin-Hoo test function, the *RBF* with the *Multiquadratic* basis function (from Φ_1 class) is the best model with the lowest median error. The lowest maximum error is given by the *Kriging model* with the *Gaussian* correlation function (also from the Φ_1 class). We observe from Fig. 5.8 that the results of the *Cascaded* and the *One-Step* techniques follow the same trend as those given by using the actual error. Interestingly, in this problem, the Pareto solutions from different classes occupy distinct regions of the objective space, as illustrated in Fig. 5.8(a). It is readily evident from Figs. 5.8(a), 5.8(b), and 5.8(c) that the optimal solutions from the Φ_1 class are significantly superior to that from the other two classes. The Pareto solution with the lowest maximum error (the lower anchor) comes at a small compromise in the median error, and hence can be considered as an attractive best trade-off model.

For the Hartmann test function, *RBF with the Gaussian* and *Multiquadratic* basis functions and *Kriging* with the *Gaussian* correlation function under Φ_1 class constitute the set of Pareto models in both *Cascaded* and *One-Step* techniques. It is observed from Fig. 7.7 that in this problem, the Pareto solutions given by *COSMOS* for different Hyper-parameter classes have a larger spread than those given by the actual error. However, in terms of the best trade-off models, there is a fair agreement between the results of *COSMOS* and those determined from the actual errors. In the Hartmann function, unlike the Branin-Hoo test function, there is some overlap between the final solutions from the different Hyper-parameter classes.

Before we move on to the results for the higher dimensional benchmark problems (with larger sample sets), where the error estimation is more expensive, it is important to appraise the cost benefits of the *One-Step* technique compared to the *Cascaded* technique.

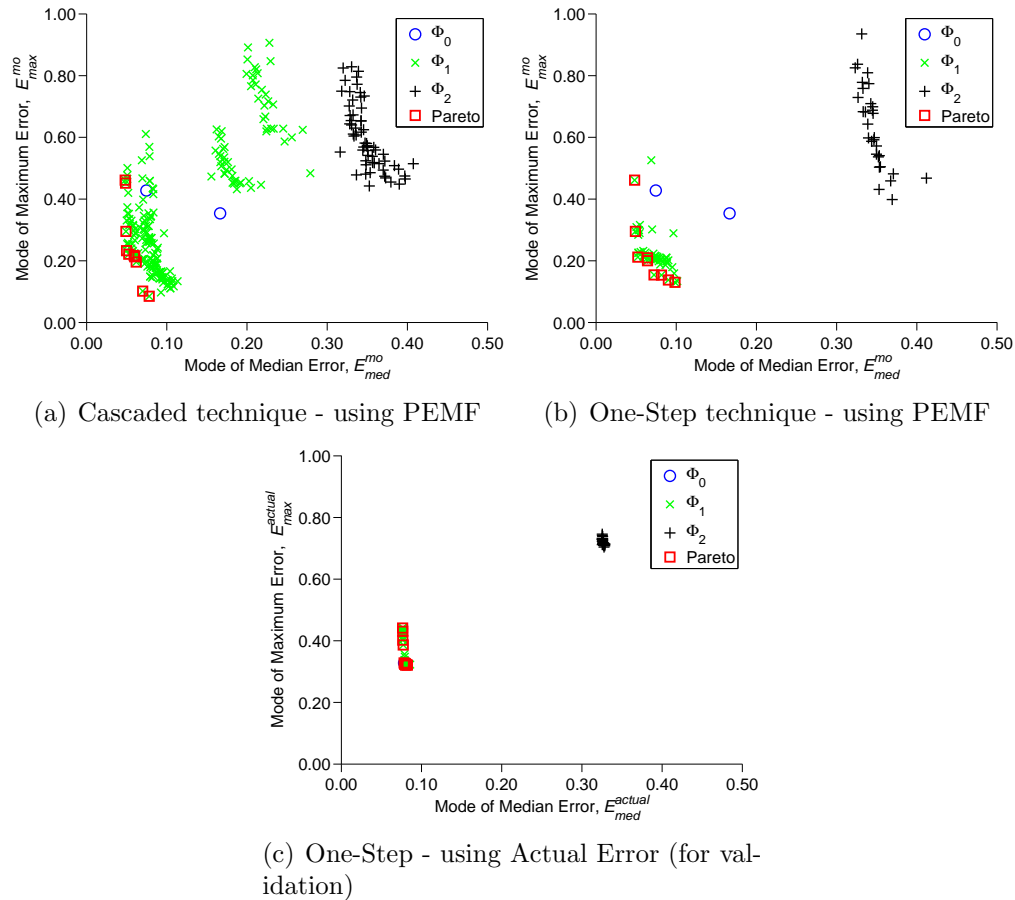


Figure 5.8: Trade-offs between modal values of median and maximum error - Branin-Hoo test function (2-variable): Pareto models and final population of models from all Φ classes

Figure 5.10 shows the computational cost of the *One-Step* and the *Cascaded* model selection techniques for the Branin-Hoo and the Hartmann test functions. This figure shows that the computational cost of the *One-Step* model selection technique is significantly lower ($< 17\%$) than that of the *Cascaded* technique. The high efficiency of the *One-Step* technique makes it uniquely helpful in the context of selecting an appropriate surrogate from a large global pool of candidate model types and kernel functions especially in the case of large sample sets (e.g., in high dimensional problems). Therefore, in this dissertation, only the *One-Step* technique is used to select the best surrogate model for the problems with high number of

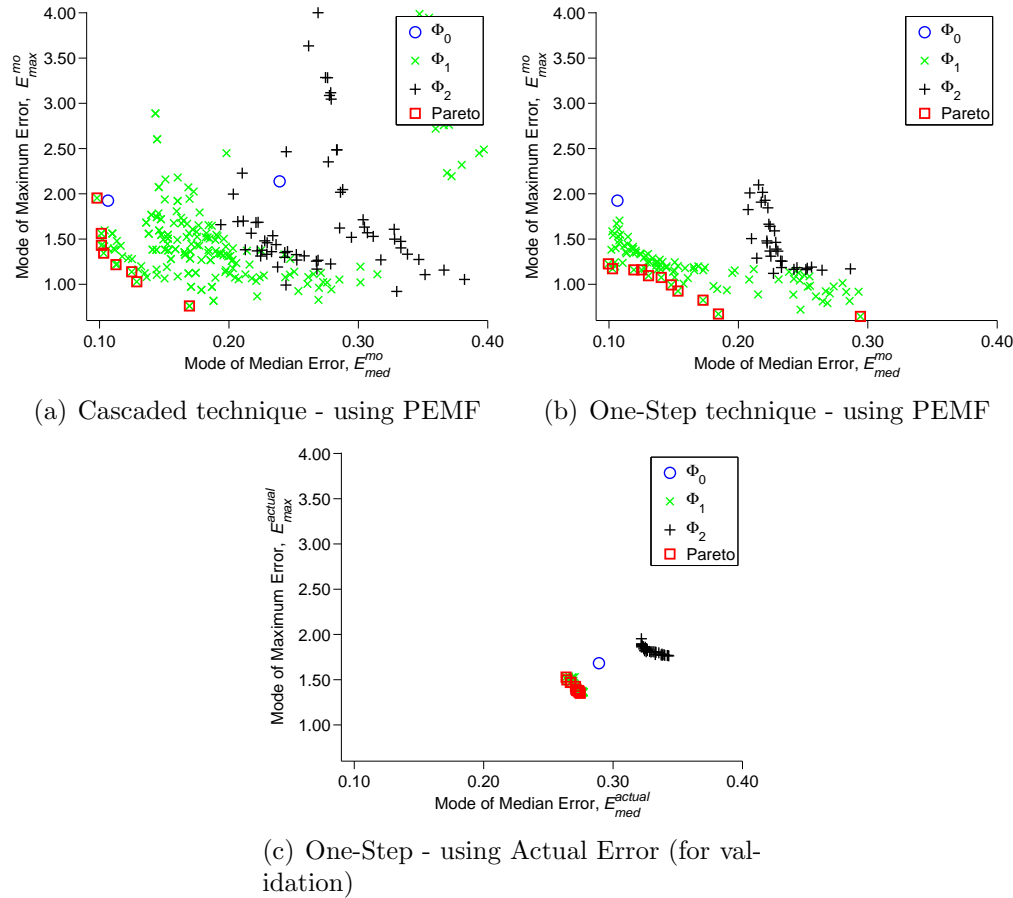


Figure 5.9: Trade-offs between modal values of median and maximum error - Hartmann test function (3-variable): Pareto models and final population of models from all Φ classes

sample points (the 10-variable Perm and the 50-variable Dixon and Price functions).

Figure 5.11 and Table 5.5 show that for the Perm test function, at least one model-kernel combination from each of the three classes (Φ_0, Φ_1 , and Φ_2) contribute to the Pareto optimal set. In this test problem, *Kriging* with *Linear correlation function* and *SVR* with *Sigmoid kernel function* are selected as the best models with the lowest median error and the lowest maximum error, respectively. It is important to note from Table 5.5 that there is promising agreement between the model-kernel combinations chosen by *COSMOS* and those chosen based on the actual error. From the *COSMOS* and the actual error results, we also observe that a Pareto solution from the Φ_0 class (*RBF-Linear*) is located at the elbow of

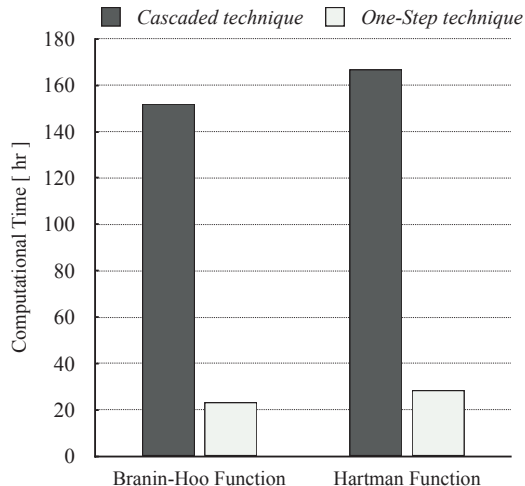


Figure 5.10: Computational time of the two different 3-level model selection techniques

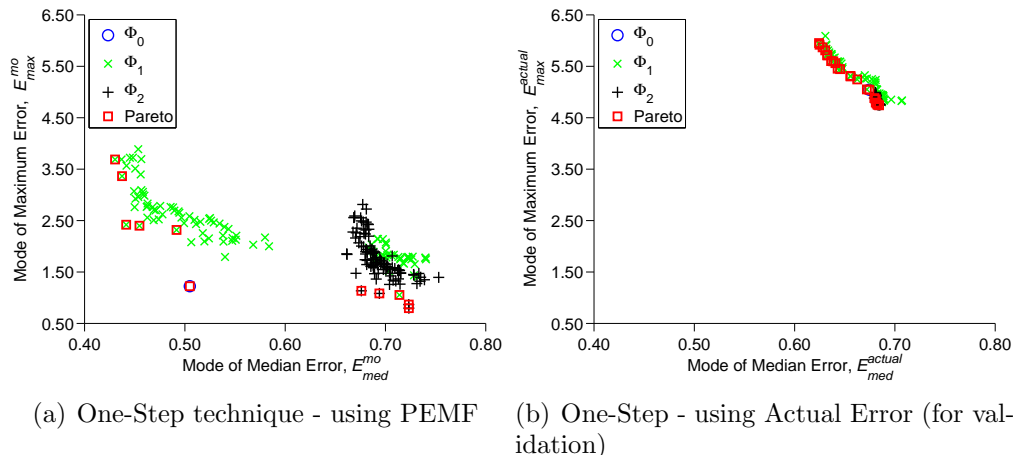


Figure 5.11: Trade-offs between modal values of median and maximum error - Perm test function (10-variable): Pareto models and final population of models from all Φ classes

the Pareto Frontier, hence represents a practically attractive best trade-off model choice.

For the Dixon & Price test function, the anchor Pareto optimal models with the lowest median and the lowest maximum errors are the *RBF-linear* and the *RBF-Cubic* models, respectively. In this test problem, the model-kernel candidates with no hyper-parameter (Φ_0 class) perform the best, and those with two hyper-parameters (Φ_2 class) perform the worst. Figure 5.12 shows that the Pareto solutions from different classes occupy distinct regions of

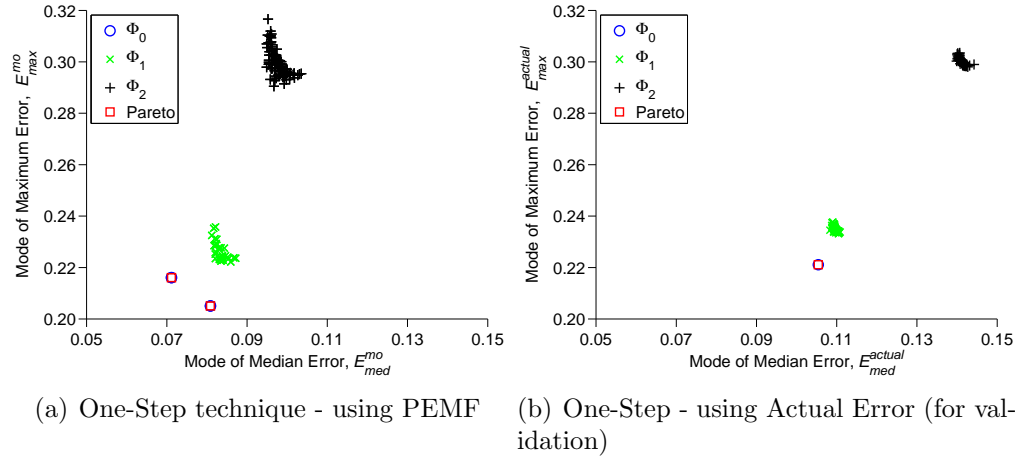


Figure 5.12: Trade-offs between modal values of median and maximum error - Dixon & Price test function (50-variable): Pareto models and final population of models from all Φ classes

the objective space. Table 5.5 shows that one model type, *RBF*, dominates the other model types in this test problem. Comparing Fig. 5.12(a) with Fig. 5.12(b), it is observed that the *RBF-Cubic* model solution from the Φ_0 class (*RBF-Cubic*) dominates the solutions from other classes in the actual result. While, in *COSMOS*, two solutions from Φ_0 class dominates all the other solutions. This small difference can be attributed to the error of the PEMF method in estimating the modal value of the median error ($< 35\%$).

Next the performance of the *COSMOS* method is investigated to show that how it performs better than simply selecting the kernel functions and the hyper parameter values recommended in the literature. The median and the maximum errors of the models selected using *COSMOS* are therefore compared with the errors of (i) the Kriging model with the Gaussian correlation function (where the correlation parameter is obtained using the MLE technique [35], (ii) the RBF with a multiquadrics basis function (where the shape parameter is set to $\sigma = 0.9$ [28], and (iii) support vector regression (SVR) with a linear kernel function (where the penalty parameter is set to $C = 50$ [122]). The median and the maximum errors of the models in each cases for the four benchmark functions are illustrated in Table. 5.6.

Table 5.6: Comparison of the performance of the COSMOS with the current paradigm in benchmark problems

Problem	Median Error					Maximum Error				
	Kriging ¹	RBF ²	SVR ³	COSMOS		Kriging ¹	RBF ²	SVR ³	COSMOS	
				Model	Error				Model	Error
Branin Hoo	0.12	0.16	0.57	RBF Multiquadric $\sigma = 2.98$	0.048	0.36	0.37	2.29	Kriging Gaussian $\theta = 2.15$	0.13
Hartmann-6	0.45	0.26	0.38	Kriging Gaussian $\theta = 2.92$	0.09	0.76	0.91	0.84	RBF Gaussian $\sigma = 0.33$	0.64
Dixon & Price	0.11	0.079	0.10	RBF Linear	0.071	0.34	0.23	0.30	RBF Cubic	0.20
Perm	0.70	0.38	0.66	Kriging Linear $\theta = 6.26$	0.43	1.20	1.72	5.61	SVR Sigmoid $C = 79.02$ $\lambda = 8.31$	0.79

¹ *Kriging-Gaussian*. Correlation parameter is obtained using a MLE technique [35]

² *RBF-Multiquadrics*. Shape parameter is set to $\sigma = 0.9$ [28]

³ *SVR-Linear*. Penalty parameter is set to $C = 50$ [122]

It is observed that, for the Branin Hoo function, COSMOS offers models which have 60% lower median error and 63% lower maximum error, respectively, than the next best models (obtained by Kriging). In Hartmann function, the models selected using COSMOS have 65% and 15% lower median and maximum error than the next best model (obtained by RBF and Kriging). The median and the maximum errors of the models selected using COSMOS for the Dixon & Price function are 10% and 13% lower than the next best model, respectively. The model selected using COSMOS based on the median error is not the best model for the Perm function. In this function, the RBF with multiquadrics basis function ($\sigma = 0.9$) has 0.11% lower error than the model selected using COSMOS. However, in this function, the model selected using COSMOS based on the maximum error criterion has 34% lower error.

TEST II ($E_{med}^{mo} - E_{med}^{\sigma^2}$) and TEST III ($E_{med}^{mo} - E_{med,\alpha}^{mo}$)

The surrogate models in the Pareto Optimal set for each benchmark problems (under Test II and Test III) are listed in Table 5.7. The Pareto optimal solutions as well as the solutions in the final population for the benchmark problems are shown in Figs. 5.13 to 5.16.

Again, each figure shows the results from Test II and Test III for a particular problem. In each of these figures, the blue 'O', the green 'X', and the black '+' symbols respectively represent the final results of model selection under Φ_0 , Φ_1 , and Φ_2 classes. The final Pareto optimal solutions in these figures are encircled by red square symbols.

Table 5.7: The set of Pareto Surrogate Models given by COSMOS

Problem	Test II, $\min[E_{med}^{mo}, E_{med}^{\sigma^2}]$	Test III, $\min[E_{med}^{mo}, E_{med,\alpha}^{mo}]$
Branin Hoo	RBF-Gaussian, RBF-Multiquadric	RBF-Multiquadric
Hartmann-3	RBF-Cubic, RBF-Gaussian, RBF-Multiquadric	RBF-Gaussian
Perm	RBF-Multiquadric, Kriging-Linear, Kriging-Exponential	RBF-Multiquadric
Dixon and Price	RBF-Linear, Kriging-Gaussian, SVR-Sigmoid	Kriging-Spherical

For the Branin Hoo function problem, RBF-Multiquadric (under class Φ_1) was selected as the best surrogate model by COSMOS in Tests III and as one of the Pareto optimal surrogates in Test I and II. For the Hartmann function, RBF-Gaussian (under class Φ_1) constituted the set of Pareto models in both Tests I, II and II; In this test problem, surrogate-kernels with one hyper-parameter performed the best and those with two hyper-parameter performed the worst.

For the Perm function, at least one surrogate-kernel combination from each of the three Φ classes contributed to the Pareto optimal set in the case of Tests I and II. In contrast, RBF-Multiquadric is the sole optimum model in Test III for the Perm function. Figures 5.15 and 5.11 show that for the Perm function,

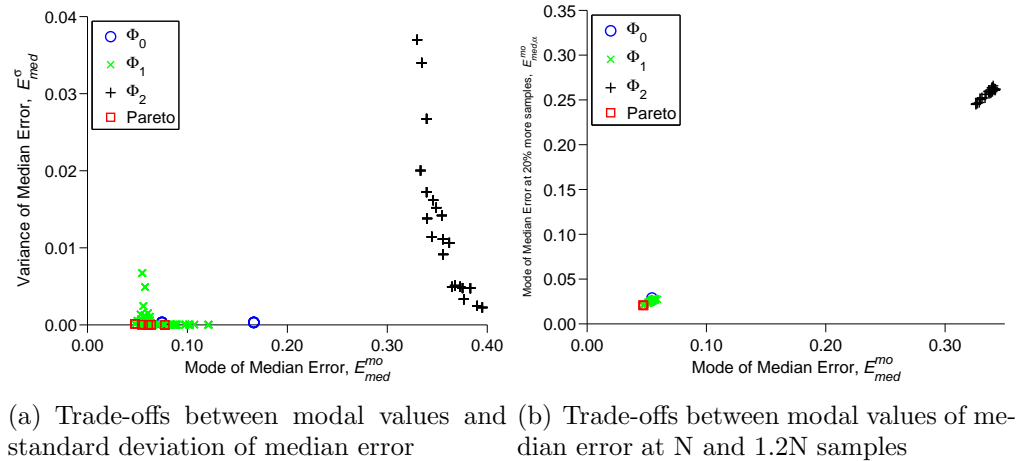


Figure 5.13: Branin-Hoo test function (2-variable): Pareto models and final population of models from all Φ classes under Test II and Test III

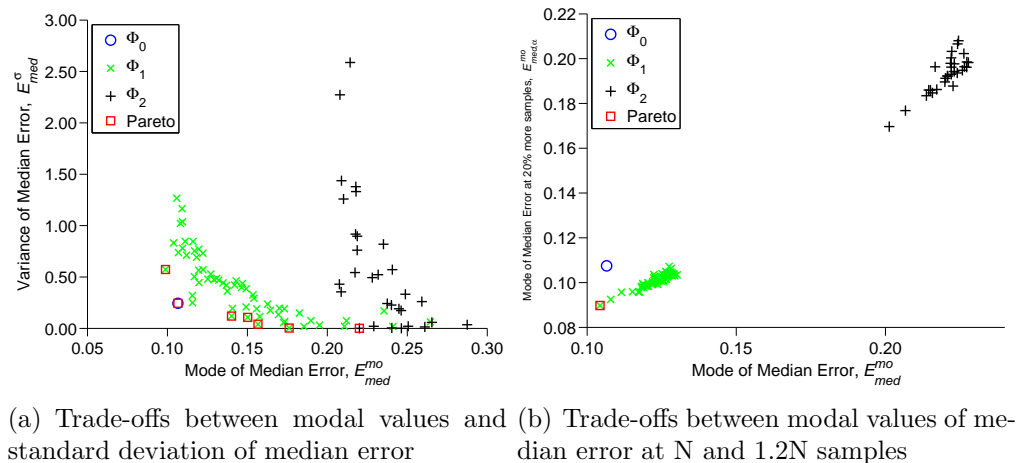


Figure 5.14: Hartmann test function (3-variable): Pareto models and final population of models from all Φ classes under Test II and Test III

5.6 Chapter Summary

This chapter presents a new model selection framework that identifies the best surrogate model (or set of models) from a diverse pool of candidates, based on error measures given by the Predictive Estimation of Model Fidelity (PEMF) method. This framework is called the *Concurrent Surrogate Model Selection (COSMOS)* framework, and is aimed to allow the selection of globally-optimal surrogates by simultaneously operating at the following three levels: (i) selecting optimal model type, (ii) selecting optimal kernel/basis function

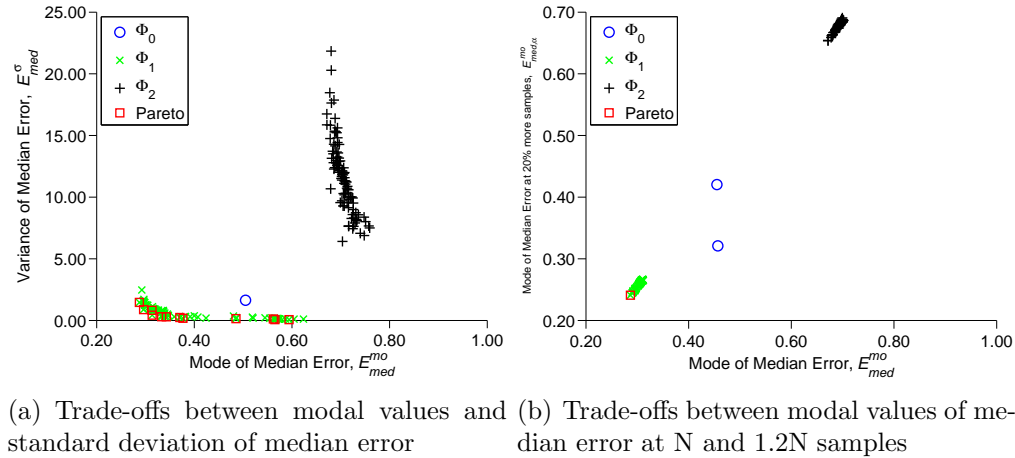


Figure 5.15: Perm test function (10-variable): Pareto models and final population of models from all Φ classes under Test II and Test III

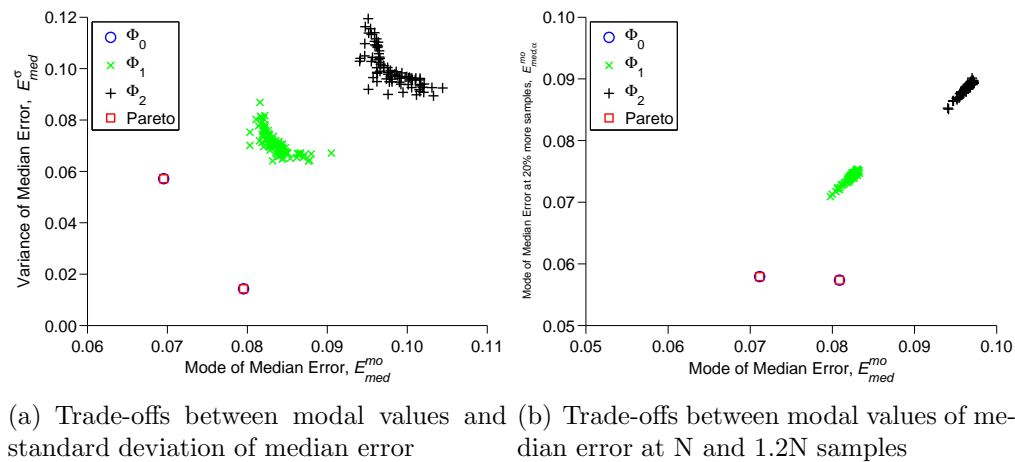


Figure 5.16: Dixon & Price test function (50-variable): Pareto models and final population of models from all Φ classes under Test II and Test III

type, and (iii) finding optimum values of the hyper-parameters in the kernel. Another key feature of *COSMOS* is the flexibility (given to the users) in using different criteria for model selection, e.g., the median error and the maximum error or the median and the variance of error. The two primary components of the *COSMOS* framework are (i) the error estimation method (PEMF) which provides the criteria functions to guide surrogate model selection, and (ii) the optimal model selector that formulates and solves multiobjective mixed integer nonlinear programming problems to search for the Pareto optimal set of surrogate models.

The PEMF method provides a model-independent approach to quantify the modal values of the median and the maximum errors in a surrogate, and has been shown to be significantly more accurate than typical leave-one-out cross validation (as illustrated in Chapter 3) – thereby facilitating effective selection of globally competitive surrogate model(s) for any given training data set.

In this chapter, *COSMOS* was applied to four analytical benchmark functions (with 2-50 variables). The results of the proposed model selection approach are compared with those based on error estimated on a large set of additional test points (for validation), resulting in reasonable agreement. The pool of candidate models tested include three major surrogate model types (Kriging, RBF, and SVR), with different kernel types for each candidate model. Widely different sets of surrogates models were selected as the optimum set in the four problems. In some of the cases, a diverse pool of model-kernel combinations were observed to form the Pareto optimal set, whereas in the other cases the Pareto front was comprised of the same model-kernel combination with different Hyper-parameter values.

CHAPTER 6

Variable-Fidelity Optimization with Adaptive Model Switching

(AMS)

How do we facilitate the applicability of population-based heuristic optimization algorithms in practical complex system design?

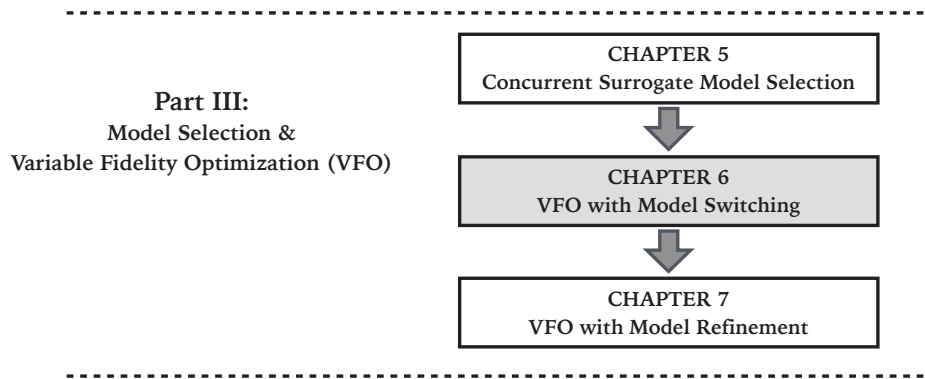


Figure 6.1: Chapter 6 orientation

This chapter presents a novel model switching technique, called “Adaptive Model Switching (AMS)”, to be implemented in population-based heuristic optimization. This technique adaptively selects different computational models (both physics-based and surrogate models) to be used during optimization, with the overall objective to identify optimal designs that have high fidelity function estimates at a reasonable computational expense. The model switching technique replaces the current model with the next higher fidelity model, when a stochastic switching criterion is met at a given iteration during the optimization process. The switch-

Portions of this chapter previously appeared as:

Mehmani, A., Chowdhury, S., Tong, W., and Messac, A., “Adaptive Switching of Variable-Fidelity Models in Population-based Optimization Algorithm,” *Engineering and Applied Sciences Optimization*, 2015, and

Mehmani, A., Chowdhury, S., and Messac, A., “Managing Variable Fidelity Models in Population-based Optimization using Adaptive Model Switching,” *The AIAA Aviation and Aeronautics Forum and Exposition*, Paper No. AIAA-2014-2436, Atlanta, Georgia, Jun 16-20, 2014.

ing criterion is based on whether the uncertainty associated with the current model output dominates the latest improvement of the relative fitness function.

The remainder of this chapter is organized as follows: Section 6.1 presents the overview of this research. Section 6.2 presents the concept and the formulation of the new Adaptive Model Switching (AMS) metrics. A description of the model error quantification methods used in this chapter and the related process, including the advancements that are made to PEMF in the AMS metric, are provided in Section 6.2.3. Section 6.3 presents the application of AMS on Shape Optimization of a Cantilever Composite Beam. The chapter summary is provided in Section 6.5.

6.1 Chapter Introduction

The primary objective of this chapter is to investigate a new adaptive model management strategy that significantly reduces the computational cost of optimization while converging to the optimum with high fidelity model evaluation. This method is designed to work with population-based optimization algorithms (e.g., GAs, PSOs), and assumes that models of different level of fidelity are available to the user. Specifically, a new stochastic model switching metric, called Adaptive Model Switching (AMS), is formulated in this chapter. The AMS technique is implemented through a powerful version of the Particle Swarm Optimization (PSO) algorithm that involves explicit diversity preservation, called *Mixed-Discrete PSO* [127]. The effectiveness of this implementation is investigated by application to two engineering design optimization problems.

6.2 Variable Fidelity Optimization with Adaptive Model Switching (AMS)

6.2.1 Major steps in Optimization with AMS

In optimization based on variable fidelity models, the important question is when and where to integrate the models with different levels of fidelity. In this chapter, the “when to integrate” question is particularly addressed. Increasing fidelity too early in the design process can be computationally expensive while wasting resources to explore undesirable regions of the design domain. On the other hand, switching to a higher fidelity model too late might mislead the search process early on to suboptimal regions of the design domain, i.e., leading to scenarios where the global optimum is outside of the region spanned by the population of candidate solutions in later iterations. In this section, a novel model management strategy called, *Adaptive Model Switching (AMS)* metric is developed to avoid both these undesirable scenarios. AMS can be perceived as a decision-making tool for the timing of model-switching or model integration. The implementation of the proposed AMS in population-based algorithm involves the following five major steps:

Step 1 Models are first ranked from the lowest fidelity to the high fidelity, based on the error associated with each model, M_i for $i = 1, \dots, n$. where model M_1 has the lowest fidelity and model M_n has the highest fidelity. Assuming the distribution of model error is known for each model, the ranking is performed using the modal values of the error distributions.

Step 2 The initial population is then generated at $t = 1$, using M_1 .

Step 3 At every iteration (t) of the heuristic optimization algorithm, the current model, M_i ,

is used to update the function values of the population, and then set $t = t + 1$. In this chapter, Particle Swarm Optimization is the chosen heuristic optimization algorithm.

Step 4 The following stopping criteria is checked after every iteration.

The optimization algorithm stops when the relative changes in the fitness function value is less than a predefined function tolerance, δ_F . To avoid termination before reaching the high fidelity model (M_n), the function tolerance must be specified to be less than the modal error of the $(n - 1)^{th}$ model (M_{n-1}).

IF the termination criteria is satisfied, the current optimum (the best global solution in case of PSO) is identified as the final optimum and the optimization process is terminated.

ELSE, Go To Step 5

Step 5 The switching metric (AMS metric) is evaluated in this step.

IF the AMS metric is satisfied, a switching event occurs, and the model is upgraded from M_i to M_{i+1} .

Go To Step 3

A flowchart of the algorithm for optimization with AMS is shown in Fig. 7.3. In practice, the AMS technique (Step 5) need not be applied at every iteration; the user can specify it to be applied after every Δt iteration. In the flowchart, AMS is shown to be applied at every iteration, for the sake of simplicity.

In the following subsection, the novel components of the AMS method (Fig. 7.3) are described. Subsequently, an overview of the Mixed-Discrete PSO algorithm, which is used for implementing the AMS method and investigating its performance, is provided.

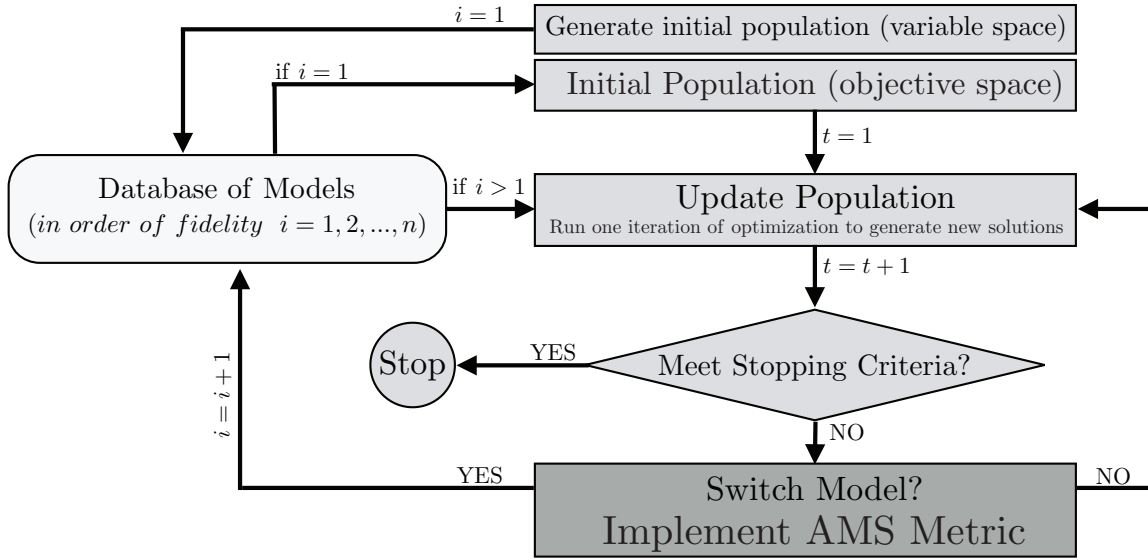


Figure 6.2: Adaptive Model Switching in population-based optimization

6.2.2 The Adaptive Model Switching (AMS) Metric

In this chapter, it is assumed that the uncertainty associated with each model (M_i ; $i = 1, \dots, n$) is known (or can be evaluated) in the form of an error distribution, \mathbb{P}_i . Under this assumption, the fitness function values evaluated using the i^{th} model can be related to the corresponding high fidelity estimation as

$$y_{HF}^i = \hat{y}_{LF}^i + \varepsilon^i \quad (6.1)$$

In Equation 6.1, \hat{y}_{LF}^i and ε^i respectively represent the response of the i^{th} low fidelity model and the stochastic error associated with it; and y_{HF}^i is the corresponding high fidelity model response. The relative improvement in the fitness function value (Δf) can be considered to follow an unknown distribution, Θ , over the population of solutions. Here, Δf in the t^{th} iteration ($t \geq 2$) can be expressed as

$$\Delta f_k^t = \begin{cases} \left| \frac{f_k^t - f_k^{t-1}}{f_k^t} \right| & \text{if } f_k^t \neq 0 \\ |f_k^t - f_k^{t-1}| & \text{if } f_k^t = 0 \end{cases} \quad (6.2)$$

where $k = 1, 2, 3, \dots, N_{pop}$

The model switching criteria is then defined based on “*whether the uncertainty associated with a model response is higher than the observed improvement in the relative fitness function of the population*”. Due to the practical unavailability of reliable local measures of model error (i.e., ε as a function of x), the model switching criteria is designed using the stochastic global measures of model error and the distribution of solution improvement. Based on prior experience or practical design requirements, the designer is likely to be cognizant of what levels of global model error, η , is acceptable for a particular low fidelity model in an optimization process. Hence, η can be perceived as a user-preference. The *critical probability*, p_{cr} for that low fidelity model with an error distribution \mathbb{P} is then defined as the probability of the model error to be less than η . This definition can be expressed as

$$p_{cr} = Pr[\varepsilon \leq \eta] = \int_0^\eta \mathbb{P}(\varepsilon') d\varepsilon' \quad (6.3)$$

The critical probability (p_{cr}) essentially indicates a critical bound in the error distribution \mathbb{P} ($0 \leq \varepsilon \leq \eta$). If the predefined cut-off value (β) of the Θ distribution lies inside this region, the current low fidelity model is considered to be no more reliable for use in the optimization process. As illustrated in Fig. 6.3, assuming that Θ and \mathbb{P} follow a log-normal distribution, $p_{cr} = Pr[\varepsilon \leq \eta^*]$, and β^* is the pre-computed cut-off value in the Θ distribution. The model with the \mathbb{P} error distribution can be used in the optimization process provided that $\eta^* \leq \beta^*$.

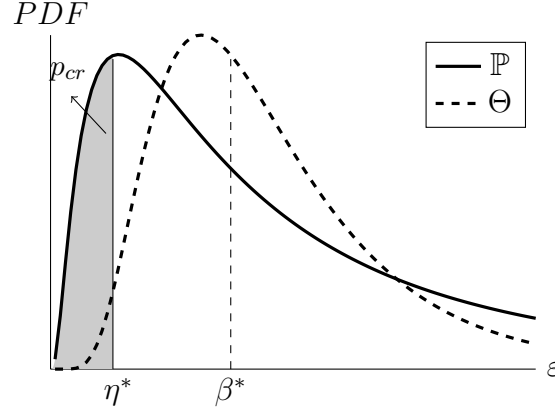


Figure 6.3: The illustration of the AMS Metric

The Adaptive Model Switching (AMS) metric is formulated as a hypothesis testing that is defined by a comparison between

- (I) the distribution of the relative fitness function improvement(Θ) over the entire population, and
- (II) the distribution of the error associated with the i^{th} model (\mathbb{P}_i) over the entire design space.

This statistical test for the i^{th} model can be stated as

$$\begin{aligned}
 H_0 &: \mathbb{Q}_{\mathbb{P}_i}(p_{cr}) \geq \mathbb{Q}_{\Theta}(1 - p_{cr}) \\
 H_1 &: \mathbb{Q}_{\mathbb{P}_i}(p_{cr}) < \mathbb{Q}_{\Theta}(1 - p_{cr}) \\
 0 &< p_{cr} < 1
 \end{aligned} \tag{6.4}$$

where \mathbb{Q} represents a quantile function of a distribution; The p -quantile, for a given distribution function, Ψ , is defined as

$$\mathbb{Q}_{\Psi}(p) = \inf\{x \in \mathbb{R} : p \leq \Psi_{(c.d.f.)}(x)\} \tag{6.5}$$

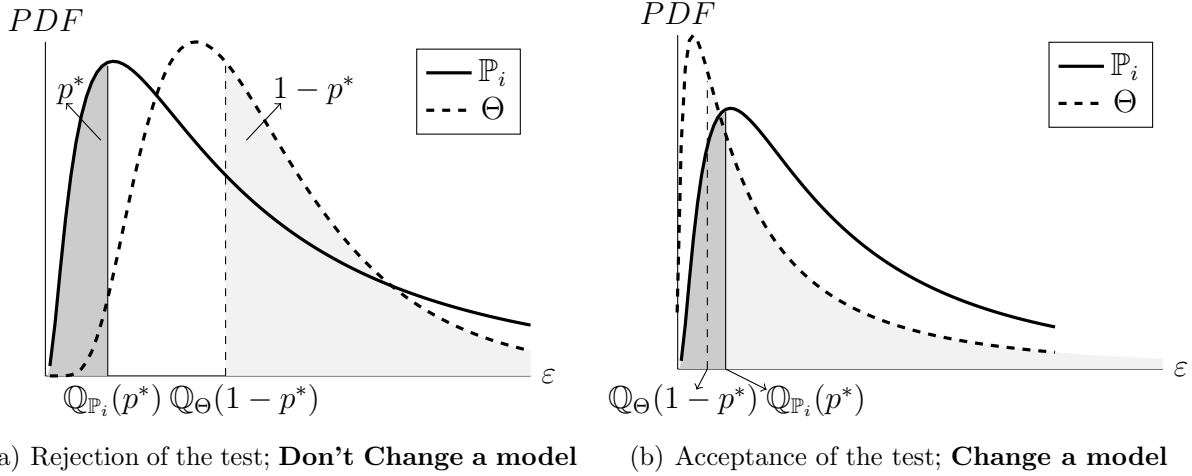


Figure 6.4: The illustration of the AMS hypothesis test (comparing the model error distribution (\mathbb{P}_i) and the distribution of fitness function improvement (Θ))

In Eq. 6.4, p_{cr} or the critical probability is an *Indicator of Conservativeness (IoC)*. The IoC is based on user preferences, and regulates the trade-off between optimal solution reliability and computational cost in the AMS-based optimization process. Generally, the higher the IoC (closer to 1), the higher the solution reliability and the greater the computational cost; under these conditions, model switching events will occur early on in the optimization process.

For the sake of illustration, assume Θ and \mathbb{P}_i follow a log-normal distribution, and $p_{cr} = p^*$. In this case, the null hypothesis will be rejected, and the optimization process will use the current model (M_i) **if** $Q_{\Theta} > Q_{\mathbb{P}_i}$, as illustrated in Fig. 6.4(a). Conversely, **if** $Q_{\Theta} < Q_{\mathbb{P}_i}$, the null hypothesis will be accepted, and the optimization process will switch to the next higher fidelity model (M_{i+1}), as shown in Fig. 6.4(b).

In this chapter, Kernel Density Estimation (KDE) is adopted to model the distribution of the relative improvement in the fitness function over consecutive Δt iterations. Since the distribution of fitness function improvement over the population (for different problems)

may not follow any particular probability model, and is also observed to be multimodal at times, KDE is a suitable choice in this context. KDE is a standard non-parametric approach to estimate the probability density function of random variables. Here, it is assumed that $\Delta f = (\Delta f_1, \Delta f_2, \Delta f_3, \dots, \Delta f_{N_{pop}})$ is an independent and identically distributed sample drawn from a distribution with an unknown density $\Theta_{\Delta f}$. The kernel density estimator can then be used to determine $\Theta_{\Delta f}$, as the average spacings of rows and columns are variable for difference rows and columns, which are given by

$$\tilde{\Theta}_{\Delta f}(x; H) = N_{pop}^{-1} \sum_{i=1}^{N_{pop}} K_H(x - x_i) \quad (6.6)$$

Here, the kernel $K(x)$ is a symmetric probability density function, H is the bandwidth matrix which is symmetric and positive-definite, and $K_H(x) = |H|^{-1/2} K(H^{-1/2}x)$. The choice of K is not as crucial as the choice of the H estimator for the accuracy of kernel density estimation [128]. In this chapter, we consider $K(x) = (2\pi)^{-d/2} \exp(-\frac{1}{2}x^T x)$, the standard normal throughout. The Mean Integrated Squared Error (MISE) method is used as a criterion for selecting the bandwidth matrix, H [129], where

$$MISE(H) = \mathbb{E} \left(\int [\tilde{\Theta}_{\Delta f}(x; H) - \Theta_{\Delta f}(x)]^2 \right) \quad (6.7)$$

6.2.3 Quantifying Model Uncertainties

The uncertainties associated with surrogate models and surrogate-based tuned low fidelity models are determined using an advanced surrogate error estimation method, called *Predictive Estimation of Model Fidelity* or *PEMF* 3. A detailed description of the PEMF method is provided in Chapter 3. In this chapter fundamental modification to the original

PEMF is made to provide the p -quantile of the surrogate error distribution. In the original PEMF, the PDFs of the median error, p_{med} is expressed as

$$p_{med} = \frac{1}{E_{med}\sigma_{med}\sqrt{2\pi}} \exp\left(\frac{(\ln(E_{med} - \mu_{med}))^2}{2\sigma_{med}^2}\right) \quad (6.8)$$

In the above equation, E_{med} represents the median of the relative errors estimated over a heuristic subset of training points at any given iteration in PEMF. The parameters, $(\mu_{med}, \sigma_{med})$ are the generic parameters of the log-normal distribution. In this variation of the PEMF method, the modal and median values of the median error distribution at any iteration, r , are then estimated as

$$\begin{aligned} E_{med}^{mo}|_r &= \exp(\mu_{med} - \sigma_{med}^2)|_r \\ E_{med}^{med}|_r &= \exp(\mu_{med})|_r \end{aligned} \quad (6.9)$$

Once the history of modal and median errors at different sample size ($< N_s$) are estimated, the variation of the modal and median values of the errors with sample density are then modeled using the multiplicative regression functions (See Eqs 3.4 and 3.5 in Chapter 3). These regression functions are then used to predict the modal values of the median and the median errors in the final surrogate, where the final surrogate is trained using all the N_s sample points. The predicted modal and the median error values, ε_{mod} and ε_{med} , are then used to define the distribution of the error in the final surrogate model, or in other words the response uncertainty of the surrogate model. The location and scale parameters of the error distribution is then given by

$$\begin{aligned}\mu_\epsilon &= \ln \varepsilon_{med} \\ \sigma_\epsilon &= \sqrt{\ln\left(\frac{\varepsilon_{med}}{\varepsilon_{mod}}\right)}\end{aligned}\tag{6.10}$$

Subsequently, the p -quantile of the error distribution associated with surrogate model is estimated by

$$\mathbb{Q}_{\mathbb{P}_{SM}}(p) = \mathbb{P}_{SM}^{-1}(p|\mu_\epsilon, \sigma_\epsilon) = \exp(\mu_\epsilon + \Phi^{-1}(p) \sigma_\epsilon)\tag{6.11}$$

In the case of physics-based low fidelity (PLF) models, the uncertainty in their output is quantified through an inverse assessment process, by comparing the physics-based low fidelity model responses with the high fidelity model responses. In this case, the relative absolute error (RAE_{PLF}) of a PLF model is estimated as

$$RAE_{PLF_i} = \begin{cases} \left| \frac{HF_i - PLF_i}{HF_i} \right| & \text{if } HF_i \neq 0 \\ |HF_i - PLF_i| & \text{if } HF_i = 0 \end{cases}\tag{6.12}$$

where $i = 1, 2, 3, \dots, N_s$ (Number of sample points)

A DoE of N_s high fidelity evaluations is used to perform the above-stated error quantification, and also to train the surrogate models and the tuned low fidelity models. The uncertainty of the low fidelity physics-based models is represented by a log-normal distribution, $\ln \mathcal{N}(\mu_{PLF}, \sigma_{PLF})$, where the p -quantile of this distribution is defined as

$$\mathbb{Q}_{\mathbb{P}_{PLF}}(p) = \mathbb{P}_{PLF}^{-1}(p|\mu_{PLF}, \sigma_{PLF}) = \exp(\mu_{PLF} + \Phi^{-1}(p) \sigma_{PLF})$$

where (6.13)

$$\mu_{PLF} = \ln\left(\frac{m_{RAR_{PLF}}^2}{\sqrt{v_{RAR_{PLF}}^2 + m_{RAR_{PLF}}^2}}\right) \quad \sigma_{PLF} = \sqrt{\ln\left(1 + \frac{v_{RAR_{PLF}}}{m_{RAR_{PLF}}^2}\right)}$$

In Eq. 6.13, $\Phi^{-1}(\cdot)$ is the inverse of the c.d.f of the standard normal distribution with zero mean and unit variance, and $m_{RAR_{PLF}}$ and $v_{RAR_{PLF}}$ are the mean and the variance of $RAE_{i=1,2,3,\dots,N_s}$, respectively.

6.2.4 Optimization Algorithm: Particle Swarm Optimization

In the proposed model management methodology, optimization is performed using an advanced implementation of the Particle Swarm Optimization (PSO). PSO was originally developed for solving continuous nonlinear optimization problems by Russel Eberhart and James Kennedy in 1995 [130]. Several advanced versions of this algorithm have been reported in the literature since then. In this article, one particular advanced implementation of the PSO algorithm called Mixed-Discrete PSO (MDPSO), which was developed by Chowdhury et al [127], is used. The advantages that the MDPSO algorithm provides over a conventional PSO algorithm include: (i) an ability to deal with both discrete and continuous design variables, and (ii) an explicit diversity preservation capability that mitigates the possibility of premature stagnation of particles. Further description of the MDPSO algorithm can be found in the paper by Chowdhury et al. [127]

6.3 Numerical Experiments with AMS

6.3.0.1 Shape Optimization of a Cantilever Composite Beam

In this optimization problem, the maximum deflection of a cantilever composite beam (shown in Fig. 6.5) is minimized. This beam is subjected to a parabolically-distributed load, $q(x) = q_0(1 - \frac{x^2}{L^2})$ [76]. In this problem, the fiber direction Young's modulus, E_L , and the composite weight density, ρ , are given by

$$E_L = E_f \nu_f + E_m (1 - \nu_f)$$

$$\rho = \rho_f \nu_f + \rho_m (1 - \nu_f)$$

where (6.14)

$$\nu_f + \nu_m = 1$$

In Eq. 6.14, E_f and E_m are the elastic modulus for graphite and epoxy resin, respectively; ρ_f and ρ_m are the weight density of the graphite fiber and epoxy resin, respectively; and ν_f and ν_m respectively represent the fiber volume fraction and the matrix volume fraction in the continuous fiber composite material

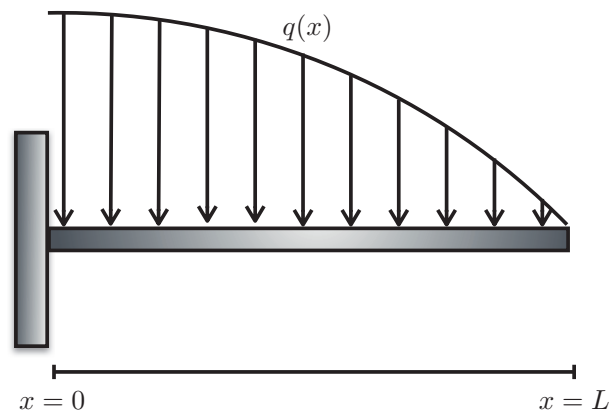


Figure 6.5: Cantilever composite beam subjected to a parabolic distributed load

The design variables include (i) the second moment of area (x_1), (ii) the depth of the beam (x_2), and (iii) the fiber volume fraction (x_3). The side constraints on the design variables and the values of the fixed design parameters are listed in Table 6.1 and Table 6.2, respectively.

Table 6.1: Design variables for the cantilever composite beam design problem

Description	Notation	Lower limit	Upper limit
Second moment of area, I [mm^4]	x_1	$3.3E4$	$20.8E4$
Depth of the beam, h [mm]	x_2	20	50
Fiber volume fraction, ν_f	x_3	0.40	0.90

Table 6.2: Specified fixed design parameters for the beam design problem

Parameter	Value
Parabolic distributed load, q_0 [N/mm]	1
Length of the beam, L [mm]	1000
Elastic modulus of graphite fiber, E_f [N/mm^2]	$2.30E5$
Elastic modulus of epoxy resin, E_m [N/mm^2]	$3.45E5$
Weight density of graphite fiber, ρ_f [N/mm^3]	$1.72E - 5$
Weight density of epoxy resin, ρ_m [N/mm^3]	$1.20E - 5$

The beam optimization problem is defined as

$$\text{Minimize : } \frac{\delta_{max}}{\delta_0}, \quad [\delta_0] = 12.93 \quad (6.15)$$

subject to

$$W/W_0 \leq 1, \quad [W_0] = 2.9E4 \quad (6.16)$$

$$\sigma_{max}/\sigma_0 \leq 1, \quad [\sigma_0] = 200 \quad (6.17)$$

$$\frac{x_2^4}{1.2E6x_1} \leq 1 \quad (6.18)$$

$$x_i^{min} \leq x_i \leq x_i^{max}, \quad i = 1, 2, 3 \quad (6.19)$$

In the optimization formulation, the inequality constraints (Eqs. 6.16, 6.17, and 6.18) are related to the allowable weight, the maximum stress, and a geometric restriction on the beam design ($depth \leq 10 \times width$). The weight and the maximum stress are given by

$$W = A\rho L = \frac{12I}{h^2} \times (12 + 5.2\nu_f)10^{-6} \times L = \frac{x_1}{x_2}(1440 + 624x_3) \quad (6.20)$$

$$\sigma_{max} = \frac{q_0 L^2 h}{8I} = \frac{1E6x_2}{8x_1} \quad (6.21)$$

The models used to estimate the maximum deflection, d_{max} , are described next.

6.3.0.2 Structural models with different levels of fidelity

To develop the high fidelity physics-based structural model (M_{HF}^B) and the low fidelity physics-based or PLF structural model (M_{PLF}^B), the Finite Element Analysis package ANSYS is used. In ANSYS, the PLF Finite Element model is constructed using 2 beam elements, while the HF Finite Element model comprises 1000 beam elements. The third model (M_{SM}^B) in this problem is a surrogate model constructed using Kriging with Gaussian correlation function. A set of 30 high fidelity function evaluations are used for this purpose. The fourth model (M_{TLF}^B) is a tuned low fidelity model constructed using the *Multiplicative form* where

$$C(x) = \frac{\delta_{max}|_{HF}}{\delta_{max}|_{PLF}} \quad (6.22)$$

The distribution of the error in the tuned low fidelity model (TLF) and the surrogate model (SM) are estimated using PEMF (Chapter 3) and are illustrated in Figs. 6.6(a) and 6.6(c), respectively. The distribution of the error in the Physics-based low fidelity model (PLF) is estimated using the inverse assessment process, by leveraging the same 30 high fidelity samples that were used to construct the TLF and SM; the PLF error distribution is

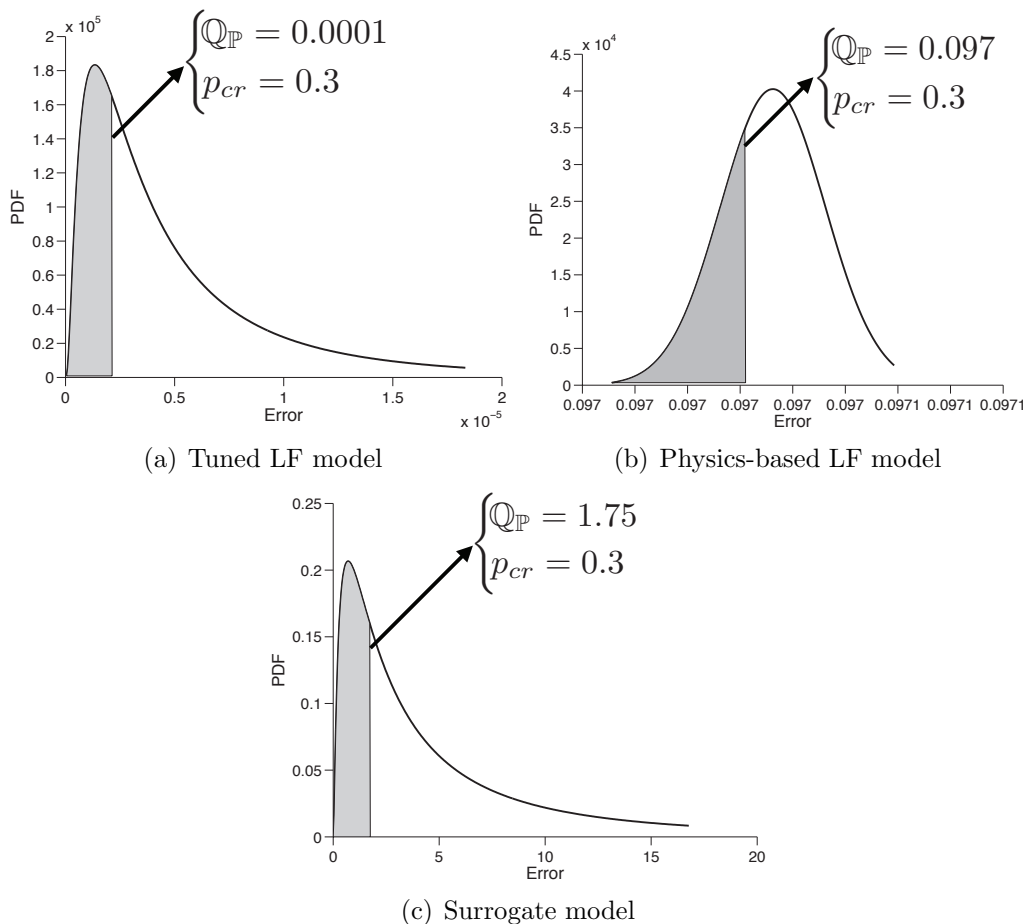


Figure 6.6: Distributions of the model errors for the cantilever beam design problem

shown in Fig. 6.6(b)

6.4 Results and Discussion

For the cantilever beam design problem, the four model types, the error distribution parameters and $Q(p_{cr})$ associated with each model, and the number of calls made by AMS to each model are listed in Table 6.3. It can be seen from Fig. 6.6 and Table 6.3 that the tuned low fidelity model provides the highest degree of accuracy and the surrogate model is the least accurate among the three low fidelity models. Hence, the initial population of particles is generated using the surrogate model in this case.

Table 6.3: Models with different levels of fidelity used in the cantilever beam optimization problem (the high fidelity model is assumed to be a true representation of the system behavior)

Model	Location parameter μ	Scale parameter σ	$\mathbb{Q}(p_{cr})$ $p_{cr} = 0.3$	Number of calls made $N_{pop} \times No. of Iter.$
Surrogate model	1.22	1.20	1.75	30×3
Physics-based low fidelity model	-2.30	0.001	0.097	30×6
Tuned low fidelity model	-12.52	0.99	0.0001	30×7
High fidelity model	-	-	-	30×4

Figures 6.7(a)-6.7(d) illustrate the distribution of the relative fitness function improvements (\mathbb{Q}_Θ) at different iterations during the optimization process. The $(1-p_{cr})$ -quantile of the \mathbb{Q}_Θ distribution, and the p_{cr} -quantile of the error distributions of the tuned low fidelity model, the surrogate model, and the physics-based low fidelity model are also shown in these figures.

The convergence history of the cantilever beam optimization performed by PSO-AMS is illustrated in Fig. 6.8. The AMS technique adaptively switches the model type three times during the optimization process at the 3rd, the 9th, and the 16th iteration, therefore resulting in an optimum design with a high fidelity function estimate. There is a substantial discontinuity in the estimated function value at the first switching event (3rd iteration), which can be attributed to the high inaccuracy of the surrogate model –the $\mathbb{Q}(p_{cr})$ value of the surrogate model (M_{SM}^B) is orders of magnitude higher than those of the other models (M_{PLF}^B and M_{TLF}^B). To avoid the termination of PSO before reaching the high fidelity model (M_{HF}^B), the relative function tolerance is set to $\delta = 1e - 5$, which is smaller than the modal error of the tuned low fidelity model (which is the most accurate low fidelity model).

In Table 6.4, the optimization results obtained by PSO-AMS is compared with the results yielded by running MDPSO solely using the surrogate model (PSO-SM), solely using the physics-based low fidelity model (PSO-PLF), solely using the tuned low fidelity model

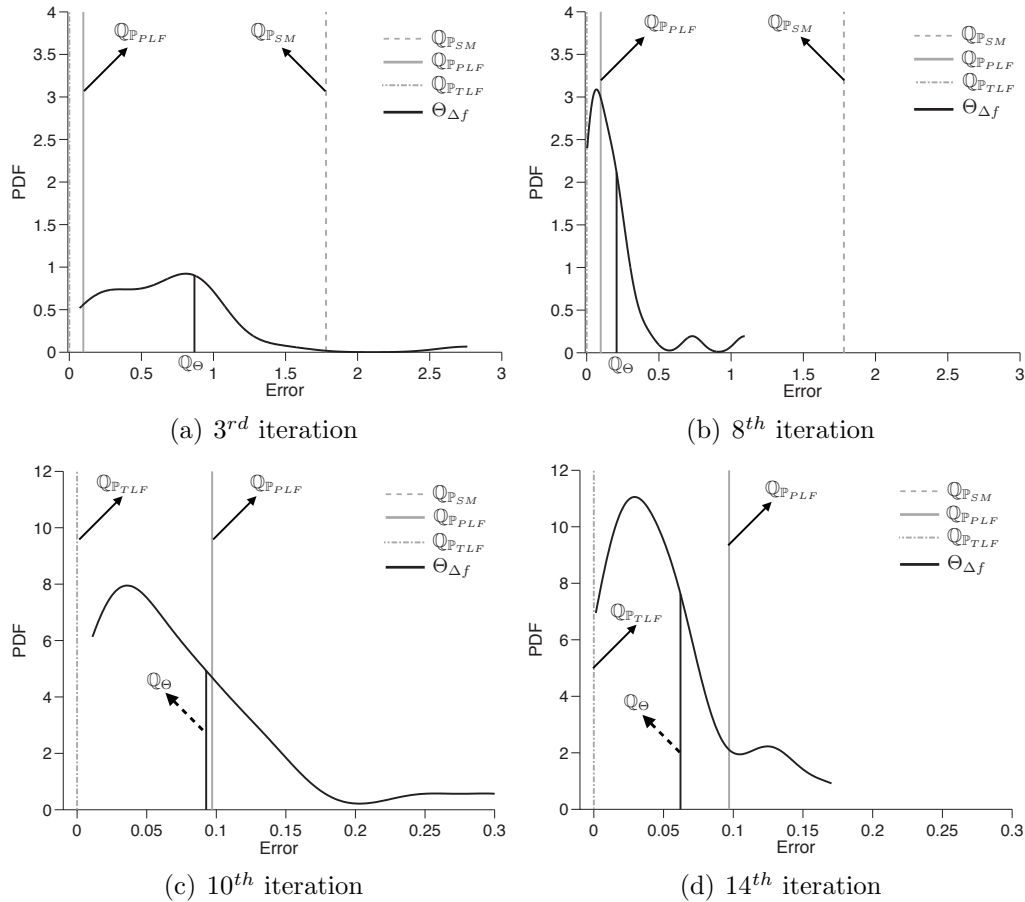


Figure 6.7: Distribution of the fitness function improvements in different iterations of the beam optimization with PSO-AMS (also showing the model error distributions)

(PSO-TLF), and solely using the high fidelity model (PSO-HF). Interestingly, the PSO-AMS, PSO-TLF, and PSO-HF arrive at the same optimum design with $f^* = 0.5435$. It is seen from Table 6.4 that PSO-AMS provides this same optimum design at a 33% lower computational expense compared to PSO-TLF and a 119% lower computational expense compared to PSO-HF (both expense differences are estimated with respect to PSO-AMS expense). It is important to note from Table 6.4 that the performance of the surrogate model-based optimization (PSO-SM) is significantly worse than that of the others. The error in the surrogate model (M_{SM}^B) at its optimum (X_{SM}^*) is more than 99%, which is expected from the predicted PEMF error of this model (Fig. 6.6(c)).

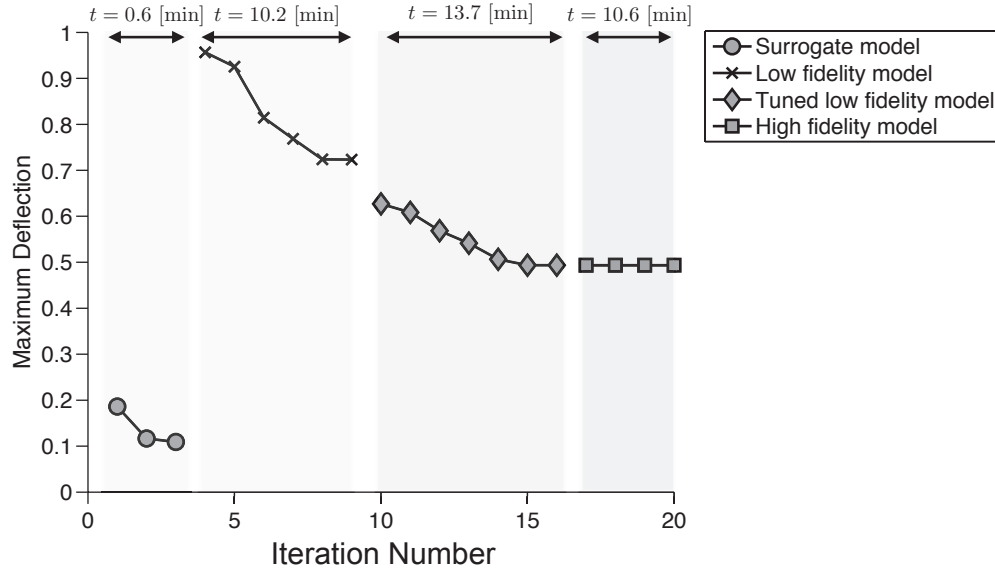


Figure 6.8: Optimization history of the cantilever beam optimization with PSO-AMS

Table 6.4: Cantilever beam design: Optimization results using different approaches

Approach	x_1^* ($e+4$)	x_2^*	x_3^*	Optimum function δ_{max}/δ_0 (f^*)	Model in last iteration	Computational time[min] over Function evaluation	HF response at optimum ($f_{HF}(x^*)$)
PSO-SM	2.82	43.28	0.71	0.0010	SM	5.7/990	0.8800
PSO-PLF	3.58	48.72	0.89	0.5011	PLF	18.4/330	0.5550
PSO-TLF	3.62	50.00	0.90	0.5435	TLF	46.8/720	0.5435
PSO-HF	3.62	50.00	0.90	0.5435	HF	76.85/870	0.5435
PSO-AMS	3.62	50.00	0.90	0.5435	HF	35.1/630	0.5435

PSO-SM: optimization performed by MDPSO solely using the surrogate model

PSO-PLF: optimization performed by MDPSO solely using the physics-based low fidelity model

PSO-TLF: optimization performed by MDPSO solely using the tuned low fidelity model

PSO-HF: optimization performed by MDPSO solely using the high fidelity model

PSO-AMS: optimization performed by MDPSO using AMS

The contributions of the four different models, in terms of computing time spent and function calls made, in the beam optimization performed by PSO-AMS are illustrated in Figures 6.9(a) and 6.9(b). It is observed that, unlike the airfoil problem, the surrogate model does not have a significant contribution in the beam optimization process in terms of function calls. Due to its high inaccuracy ($\mathbb{Q}(p_{cr}) = 1.75$), the fitness function improvement of the particles is quickly dominated by the error distribution of the surrogate model (in only 3 iterations). In this optimization process, the tuned low fidelity model (M_{TLF}^B) makes the

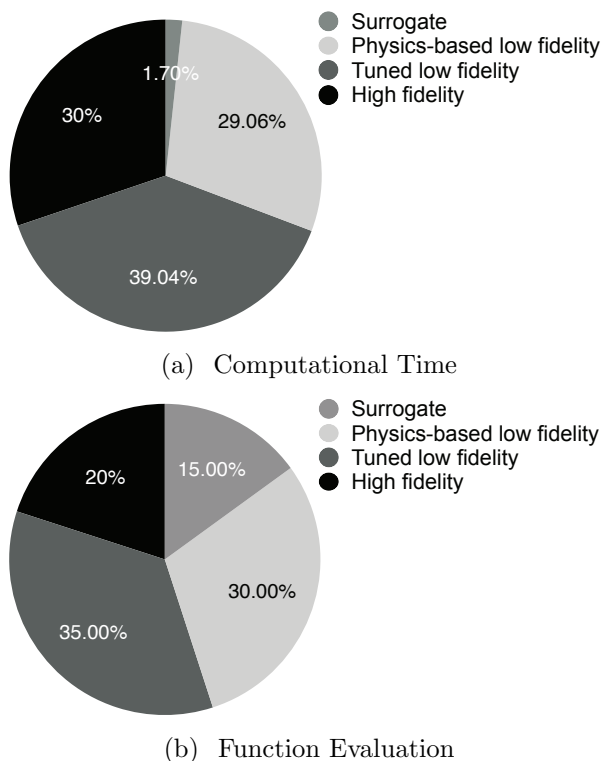


Figure 6.9: Percentage-contribution of each model in the cantilever beam optimization problem using PSO-AMS

highest contribution in terms of computing time and function calls. This case study again shows that the uncertainty in the lower fidelity models could exceed the relative function improvement across constitutive iterations way ahead of reaching convergence in practical optimization. Such likely scenarios make this variable fidelity optimization technique (AMS) a unique tool for designing complex systems, where fast low fidelity models are almost indispensable.

6.5 Chapter Summary

This chapter presented a novel model management technique that is implemented in population-based optimization algorithms to provide high fidelity optimum designs at reasonable computational expense. The optimization process is started using the model with

the highest computational efficiency, which is often the model with the lowest fidelity (which could be a physics-based low fidelity model or a surrogate model). A novel switching metric (called Adaptive Model Switching or AMS) is then used to determine when to switch to the next higher fidelity model during the optimization iterations. Assuming that the uncertainties associated with the lower fidelity models follow a probabilistic distribution (lognormal pdf is used here), the proposed model switching metric is defined as: “a probability estimate of whether the uncertainty associated with a model exceeds the improvement in the relative fitness function over the population of solutions”. In this chapter, the new adaptive model switching technique (AMS) is applied to cantilever composite beam design. A powerful version of the Particle Swarm Optimization (mixed-discrete PSO) algorithm is used to implement and investigate the performance of AMS. The results indicate that AMS along with Mixed Discrete PSO improve the efficiency of the optimization process significantly when compared to optimization solely using high fidelity models, while reaching at the same or a better optimum. The value of the optimum with AMS is also better than that accomplished using only single low fidelity models for optimization.

CHAPTER 7

Variable-Fidelity Optimization with Adaptive Model Refinement (AMR)

How do we find the optimum/optima with certainty in surrogate-based design optimization?

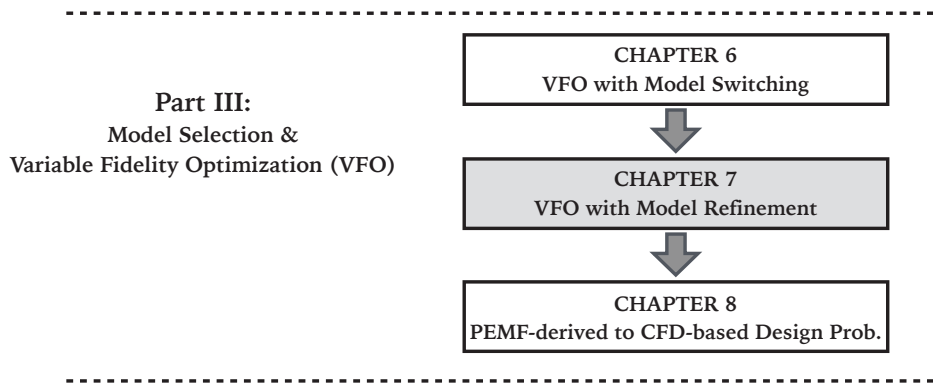


Figure 7.1: Chapter 7 orientation

In this chapter, the important challenge in using surrogate models in optimization, which is misleading the search process due to underestimation or overestimation of a system behavior and leading to suboptimal or infeasible solutions, is addressed. We develop a new model-independent surrogate-based design approach to refine the surrogate model during optimization. The proposed approach, called Adaptive Model Refinement (AMR), is designed to work particularly with population-based optimization algorithms. In AMR, reconstruction

Portions of this chapter previously appeared as:

Mehmani, A., Chowdhury, S., and Messac, A., Variable-Fidelity Optimization with In-Situ Surrogate Model Refinement, *ASME 2015 International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE)*, Paper No. DETC2015-47188, Boston, MA, August 2-5, 2015.

Mehmani, A., Tong, W., Chowdhury, S., and Messac, A., "Surrogate-based Particle Swarm Optimization for Large-scale Wind Farm Layout Design," *11th World Congress on Structural and Multidisciplinary Optimization*, Sydney, Australia, June 7-12, 2015.

(or refinement) of the model is performed by sequentially adding a batch of new samples at any given iteration (of surrogate-based optimization (SBO)) when a switching metric is met. In this approach, the model switching metric (developed in Chapter 6) is applied.

The remainder of this chapter is organized as follows: Section 7.1 presents the overview of this research; Section 7.2 presents the concept and the formulation of the new surrogate based design optimization method, called Adaptive Model Refinement (AMR). The advancements that are made to PEMF to determine the batch size (to be added) in SBO process is provided in Section 7.2.2. Section 7.3 describes the surrogate models used to demonstrate COSMOS, and the different benchmark problems to which COSMOS is applied. The numerical settings and results of case studies are also shown and discussed in this section. The chapter summary is provided in Section 7.5.

7.1 Chapter Introduction

The primary objective of this chapter is to investigate a new surrogate-based optimization approach that reduces the computational cost of optimization while converging to the optimum/optima with certainty. In its current form, this method is designed to work with population-based optimization algorithms (e.g., GAs, PSOs). The proposed SBO approach has the following characteristics:

- (i) It is independent of the type of model (e.g., Radial basis function, Polynomial response surface, Gaussian processes). Hence, it can be implemented in conjunction with different types of surrogate models instead of strategies which are defined for specific types of model (e.g., EGO works primarily for Gaussian process-based surrogate models);
- (ii) It uses a reliable surrogate model (with a desired level of fidelity) at any given iteration of SBO. Hence, it can lead the search process to the reliable optimum solution; and
- (iii) It determines the optimal batch size for the infill points for the upcoming iteration of SBO. Hence, the computational efficiency of the population-based heuristic opti-

mization algorithms can be significantly improved over the one-at-a-time sequential approaches and the approaches which the batch size is predefined by the user.

7.2 Surrogate-based optimization with Adaptive Model Refinement

7.2.1 Major steps in Surrogate-based Optimization with AMR

In this section, we develop a method called, *Adaptive Model Refinement (AMR)*. The major components of the surrogate-based optimization using AMR method include:

- (i) The model switch metric (Chapter 6): This metric is perceived as a decision-making tool for the timing of surrogate model refinement. Performing model refinement (by adding infill points) too early in the optimization process can be computationally expensive while wasting resources to explore undesirable regions of the design domain. On the other hand, updating surrogate model too late might mislead the search process early on to suboptimal regions of the design domain, i.e., leading to scenarios where the global optimum is outside of the region spanned by the population of candidate solutions in later iterations.
- (ii) The Predictive Estimation of Model Fidelity (Chapter 3): In AMR method, the PEMF method with certain implementation is used to quantify the surrogate model fidelity, and identify the optimal batch size (e.g., size of infill points). The inputs and outputs of PEMF (in the AMR method) are illustrated in Fig. 7.2 and can be expressed as:

$$[\mathbb{P}_{(\mu_\varepsilon, \sigma_\varepsilon)}, \Gamma^{Infill}] = f_{PEMF}(X, \varepsilon^*) \quad (7.1)$$

where the vector X represents the sample data (input and output) used for training the surrogate model; and ε^* is the desired fidelity in the model refinement process. In Eq. 7.1, $\mathbb{P}_{(\mu_\varepsilon, \sigma_\varepsilon)}$, and Γ^{Infill} respectively represent the distribution of the error in the surrogate model, and the batch size for the infill points to achieve a desired level of fidelity in the AMR method. A brief description of the KDE and PEMF methods are provided in the following sections.



Figure 7.2: The inputs and outputs of the PEMF method

- (iii) Optimization algorithm: Mixed-Discrete PSO (MDPSO): In the proposed surrogate-based optimization methodology, optimization is performed using an advanced implementation of the Particle Swarm Optimization (PSO). In this chapter, we use one particular advanced implementation of the PSO algorithm called Mixed-Discrete PSO (MDPSO), which was developed by Chowdhury et al [127].

The surrogate-based optimization using AMR approach is implemented by observation the following major steps:

Step 1 A set of initial sampling points are generated in the design space using design of experiment methods (Latin Hypercube (LH) sampling [98] is applied to sample the whole design space in the first iteration). An initial surrogate model is then constructed using the initial set of sample points.

Step 2 The initial population is generated at $t = 1$, using the initial surrogate model.

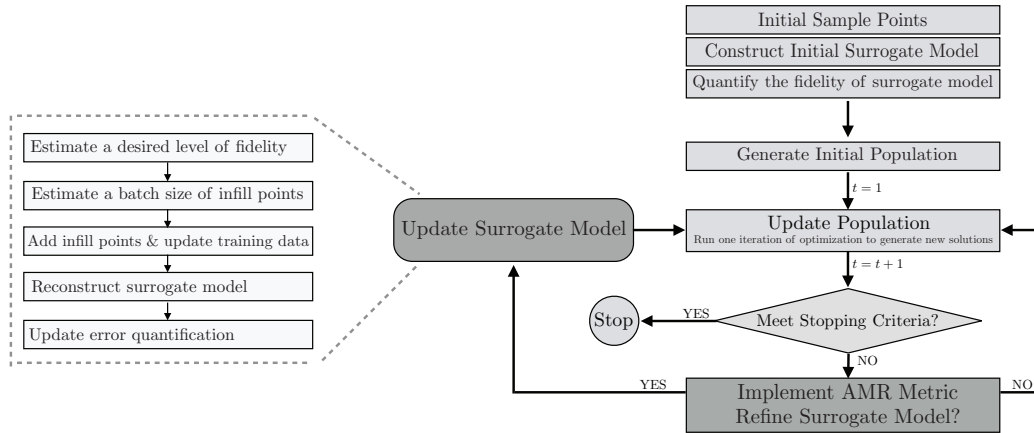


Figure 7.3: Adaptive Model Refinement in Surrogate-based Optimization

Step 3 At every iteration (t) of the heuristic optimization algorithm, the current surrogate model is used to update the function values of the population, and then set $t = t + 1$. Particle Swarm Optimization is the chosen heuristic optimization algorithm.

Step 4 The stopping criterion is checked. The following two different methods can be used as the stopping criteria: (i) the difference between optimum values of five consecutive iterations is less than a threshold value, (ii) the maximum allowed number of evaluations of function is reached. In this chapter, the optimization algorithm stops when the relative changes in the fitness function values in five consecutive iterations is less than a predefined function tolerance, δ_F . If the termination criterion is satisfied, the current optimum (the best global solution in case of PSO) is identified as the final optimum and the optimization process is terminated. Otherwise, go to **Step 5**.

Step 5 The model switching criteria (developed in Chapter 6) is evaluated in this step. If the switching metric is satisfied, a model refinement occurs, and the surrogate model is updated. Go to **Step 3**

A flowchart of the AMR method is shown in Fig. 7.3. In the flowchart, the metric is shown to be applied at every iteration, for the sake of simplicity. In the following subsections, we describe the novel component of the AMR method (Fig. 7.3); the model switching criteria was explained in Chapter 6. Next, we provide an overview of the AMR method in estimating the optimal batch size for the infill points for the upcoming iteration of SBO.

7.2.2 The Adaptive Model Refinement (AMR)

In the previous section, the concepts of the AMR method is defined. Based on the AMR method, the model refinement will be performed at the t^* -th iteration of SBO, when the model switching metric is met, under the condition that

$$Q_{\mathbb{P}}^{\text{SM}^{\text{CURR}}} \geq Q_{\Theta}^{t=t^*} \quad (7.2)$$

where SM^{CURR} represents the current surrogate model in the optimization process. Model refinement is performed to efficiently improve the fidelity of the current surrogate model to achieve the “desired fidelity” for the upcoming iterations of SBO. In this chapter, the desired fidelity, ε_{mod}^* , is determined using the history of the fitness function improvement in the optimization process, which is given by:

$$\varepsilon_{mod}^* = \left| 1 - \frac{Q_{\Theta}^{t=t^*} - Q_{\Theta}^{t=t^*-\tau}}{Q_{\Theta}^{t=t^*}} \right| \times \varepsilon_{mod}^{\text{CURR}} \quad (7.3)$$

In Equation 7.3, $\varepsilon_{mod}^{\text{CURR}}$ is the predicted modal error value associated with the current surrogate model; and $\tau \in \mathbb{Z}_{<t^*}$ is a user-defined parameter which regulates the occurrences of the “surrogate model refinement” in the proposed SBO approach. Numerical experiments exploring different values for the τ indicated that the $3 \leq \tau \leq 5$ can be the suitable choice.

The novel and unique strategy of tracking the variation of error with an increasing density of training points in the PEMF method 3 is then applied to determine the batch size for the samples to be added to the current sample points (X^{CURR}). To do this end, the desired batch size (Γ^{Infill}) is estimated using the inverse of regression functions used to represent the variation of error with sample density (the Type 1 and Type 2 VESD functions given in Equations 3.4 and 3.5, respectively), as shown below:

$$\Gamma^{Infill} = \begin{cases} \lceil \frac{\ln(\varepsilon_{mod}^*) - \ln(a^{CURR})}{b^{CURR}} \rceil - N_s^{CURR} & \text{Type 1} \\ \lceil \exp(\frac{\ln(\varepsilon_{mod}^*) - \ln(a^{CURR})}{b^{CURR}}) \rceil - N_s^{CURR} & \text{Type 2} \end{cases} \quad (7.4)$$

where a^{CURR} and b^{CURR} are regression coefficients of the VESD function in the current surrogate model, determined using the least square method; and X^{CURR} represents the size of the current sample points.

The location of the new infill points ($X^{Infill} | N(X^{Infill}) = \Gamma^{Infill}$) in the input space is determined based on a hypercube (F) enclosing promising current candidate designs in the optimization process. The lower bound (L_j^F) and the upper bound (U_j^F) of the j^{th} dimension of the current hypercube (F) are estimated by

$$\begin{aligned} L_j^F &= \text{Min}\{x_j^{min}, X_j^{min}\} \\ U_j^F &= \text{Max}\{x_j^{max}, X_j^{max}\} \end{aligned} \quad (7.5)$$

In Equation 7.5, x_j^{min} and x_j^{max} are respectively the lower and the upper bounds of the entire set of the current candidate solutions in the j^{th} dimension of the design space; and X_j^{min} and X_j^{max} are the lower and upper bounds of the j^{th} dimension of the design space, respectively.

The distance-based criterion is then applied to select the optimum setting for the new infill points ($X^{Infill} \subset F$) based upon the current sample points (X^{CURR}). The objective of this criterion is to minimize the correlation between the current and the new points and yield a design which is distributed uniformly, as given by

$$\mathbf{Max}_{X^{Infill} \subset F} \mathbf{Min}_{\mathbf{x}^s, \mathbf{x}^t \in X^{UPDATED}} \mathbb{D}(\mathbf{x}^s, \mathbf{x}^t) \quad (7.6)$$

where \mathbb{D} denotes the Euclidean distance; and $X^{UPDATED}$ ($X^{UPDATED} = \{X^{CURR}, X^{Infill}\}$, and $N(X^{UPDATED}) = N_s^{CURR} + \Gamma^{Infill}$) is the updated training set which is used to refine (reconstruct) the current surrogate model. An appealing feature of this distance based criterion is that it is computationally efficient and reasonably easy to implement in a batch sequential manner. However, the proposed AMR method can also use other sequential sampling criteria. The updated surrogate models will be used in the next iteration of the PSO in the AMR method. The PEMF error measures used in the proposed SBO approach is discussed next.

7.3 Numerical Experiments with AMR

The effectiveness of the proposed AMR metric in surrogate based optimization is explored using Kriging, RBF, and E-RBF surrogate models for the following three analytical test functions: (i) the 5-variable Rosenbrock function, (ii) the 6-variable Hartmann function, and (iii) the 10-variable Levy function. The equations of these three test functions are given below:

Rosenbrock Function (5 variables):

$$f(x) = \sum_{i=1}^{N/2} [100(x_{2i-1}^2 - x_{2i})^2 + (x_{2i-1} - 1)^2] \quad (7.7)$$

where $x = (x_1 \ x_2 \ \dots \ x_n)$ $x_i \in [-5 \ 10]$

$n = 5$

Hartmann Function (6 variables):

$$f(x) = - \sum_{i=1}^4 c_i \exp \left\{ - \sum_{j=1}^n A_{ij} (x_j - P_{ij})^2 \right\} \quad (7.8)$$

where $x = (x_1 \ x_2 \ \dots \ x_n)$ $x_i \in [0 \ 1]$

In Hartmann-6, $n = 6$; the constants c , A , and P , are a 1×4 vector, a 4×6 matrix, and a 4×6 matrix, respectively.

Levy Function (10 variables)

$$f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] \\ + (w_d - 1)^2 [1 + \sin^2(2\pi w_d)] \quad (7.9)$$

where $w_i = 1 + \frac{x_i - 1}{4}$ $i = 1 \ 2 \ \dots \ n$

$x = (x_1 \ x_2 \ \dots \ x_n)$ $x_i \in [-10 \ 10]$

$n = 10$

These three test problems are not computationally expensive, but they have char-

acteristics (e.g., having several local minima) typically encountered with computationally expensive practical engineering global optimization problems.

7.4 Results and Discussion

In this chapter, the initial number of sample points (initial investment) for training the surrogates in each application is $N(\{X^{Initial}\}) = 5 \times d$, where d is a problem dimension. To implement *PEMF* in the *AMR* method, the size of training points at each iteration (as a function of iteration) is determined by

$$n^r = (r - (N^{itr} + 1)) \times \lceil \ell \times N_s \rceil \quad (7.10)$$

where ℓ and N^{itr} are the normalized incremental value and the number of iteration. These parameters are pre-specified to $\ell = 0.1$ and $N^{itr} = 4$. In this section, the AMR metric is applied at every iteration $\delta = 1$, the Indicator of Conservativeness (IoC) is pre-specified to $p_{cr} = 0.3$, the population size of PSO is pre-specified to $N_{pop} = 100$, and the PSO algorithm converges by satisfying the predefined function tolerance, $\delta f = 1e - 5$.

Figures 7.4(a)-7.4(c) represent the improvement of the model fidelity through sequential model refinement process of the AMR method for the three benchmark problems in conjunction with different surrogate models (Kriging, RBF, and ERBF). These figures illustrate that the errors decrease with sequentially adding batch of infill points in SBO, thus showing that the batch size of the infill points and the location of these points determined by the AMR method resulted in lower surrogate model error in the refined (updated) surrogate model. It should be noted that the novel use of the modal value in the PEMF method, promotes a decrease of the error measure with infill points, as opposed to the untraceable

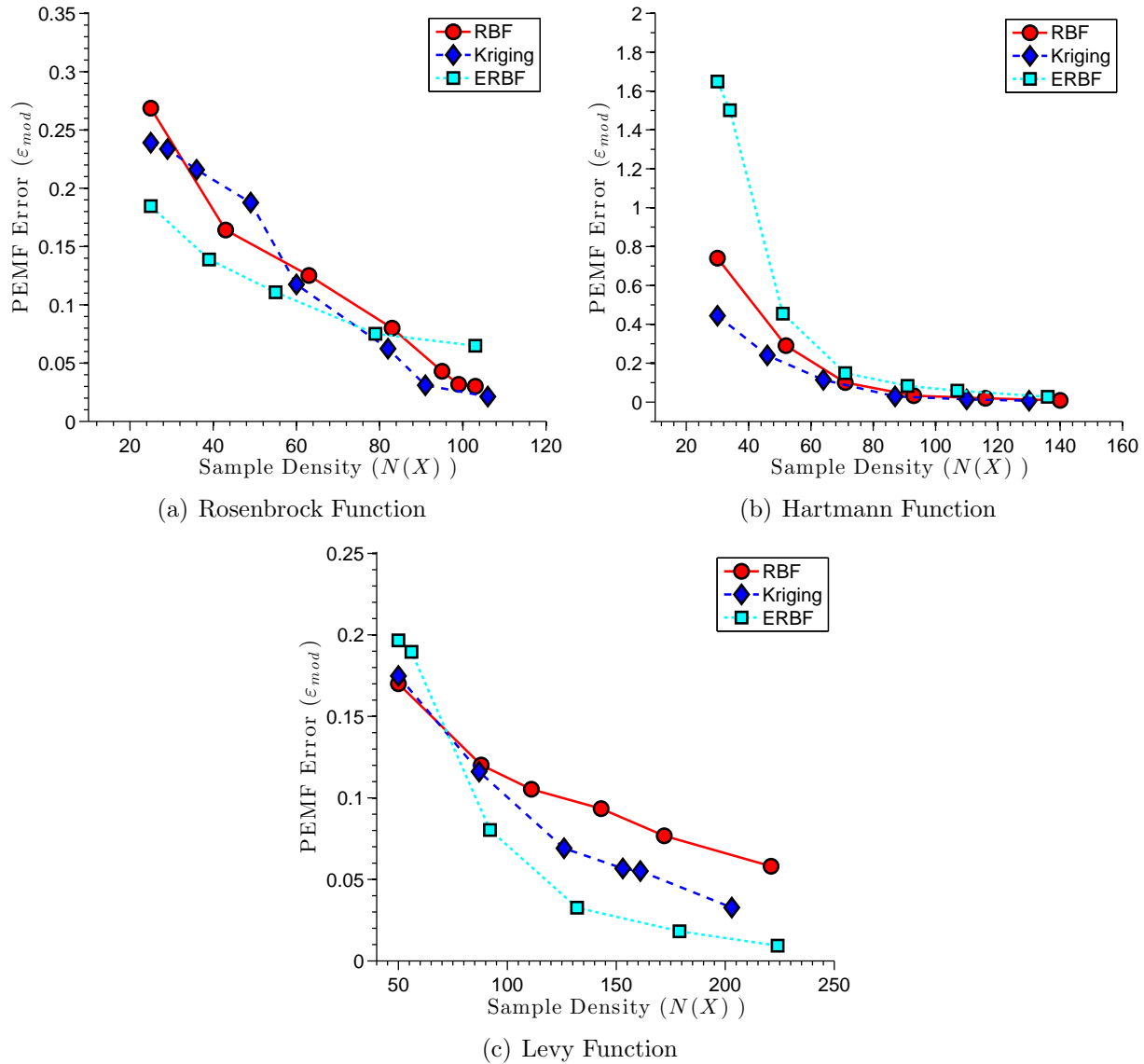


Figure 7.4: The improvement of the surrogate model fidelity through sequential model refinement process with AMR

noisy variation that is often characteristic of mean or RMS error measures [121]. It is also important to note that AMR provides additional helpful insights into the performance of the surrogates which can be used in the context of *Adaptive Model Selection*. For example, in the Hartmann function, it is observed (from Fig. 7.4(b)) that although the predicted error of the initial ERBF model is relatively higher than that of Kriging and RBF (using initial sample points), ERBF is significantly more sensitive to the infill points than that of the RBF

and Kriging surrogates.

Figures 7.5(a)-7.5(h) illustrate the distribution of the fitness function improvement at different iterations during the SBO process of the Rozenbrok function using AMR-RBF. In these figures, \mathbb{Q} represents $(1 - p_{cr})$ -quantile of the Θ distribution. The error distributions of the surrogate models constructed in the model refinement process (the 1st surrogate model to the 5th surrogate model), which are determined through the AMR method, are also shown in these figures. In this case, model refinement through AMR occurs at the 8th, the 14th, the 24th, the 29th, the 32th, and the 36th iterations.

The convergence history of the benchmark optimization problems for different surrogate models are illustrated in Figs. 7.6(a)- 7.8(c). These figures also indicate: (i) the active surrogate model at each iteration (using different symbols/colors), (ii) the size of data set ($N(X^{UPDATED}) = N_s^{CURR} + \Gamma^{Infill}$) used to refine (update) the active surrogate model in the AMR method, and (iii) the iteration that the model refinement is performed (the AMR metric is met). In these problems, the optimization progresses converges by satisfying the predefined function tolerance, $\delta f = 1e - 5$.

Next, we investigate how the AMR method performs better than simply using the single stage sampling strategy for SBO. The results yielded by the AMR method are therefore compared with the results yielded by running MDPSO using the surrogate model constructed in the one-step method. The optimum results thus obtained in each case are reported in Table 7.1. The final columns of this table shows the actual function estimate at the optimum design obtained under each optimization run. It is observed that the AMR method provides the better optimum values in most of the cases.

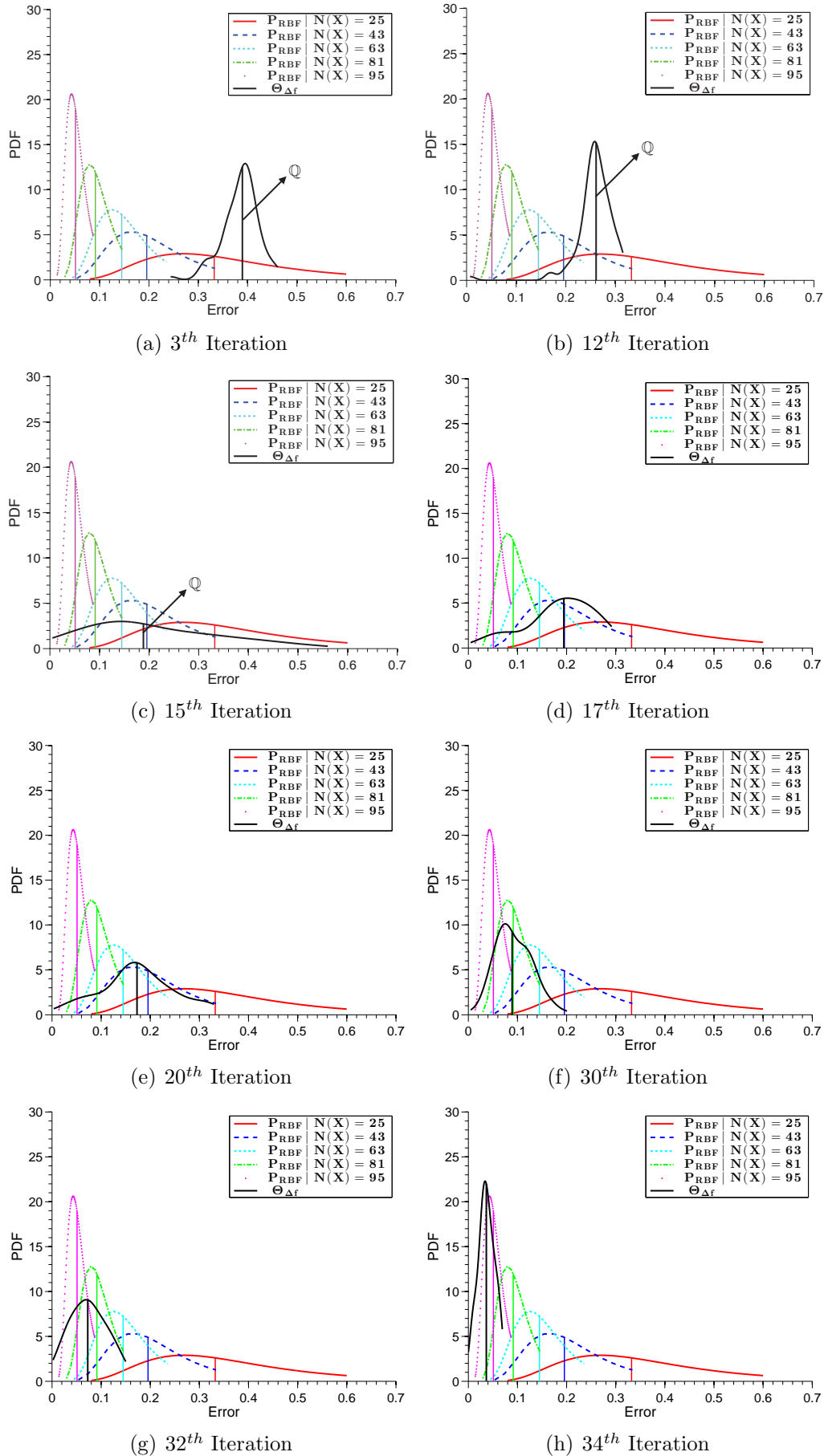


Figure 7.5: Distribution of the fitness function improvements in different iterations of the global optimization problem with AMR-RBF (also show-

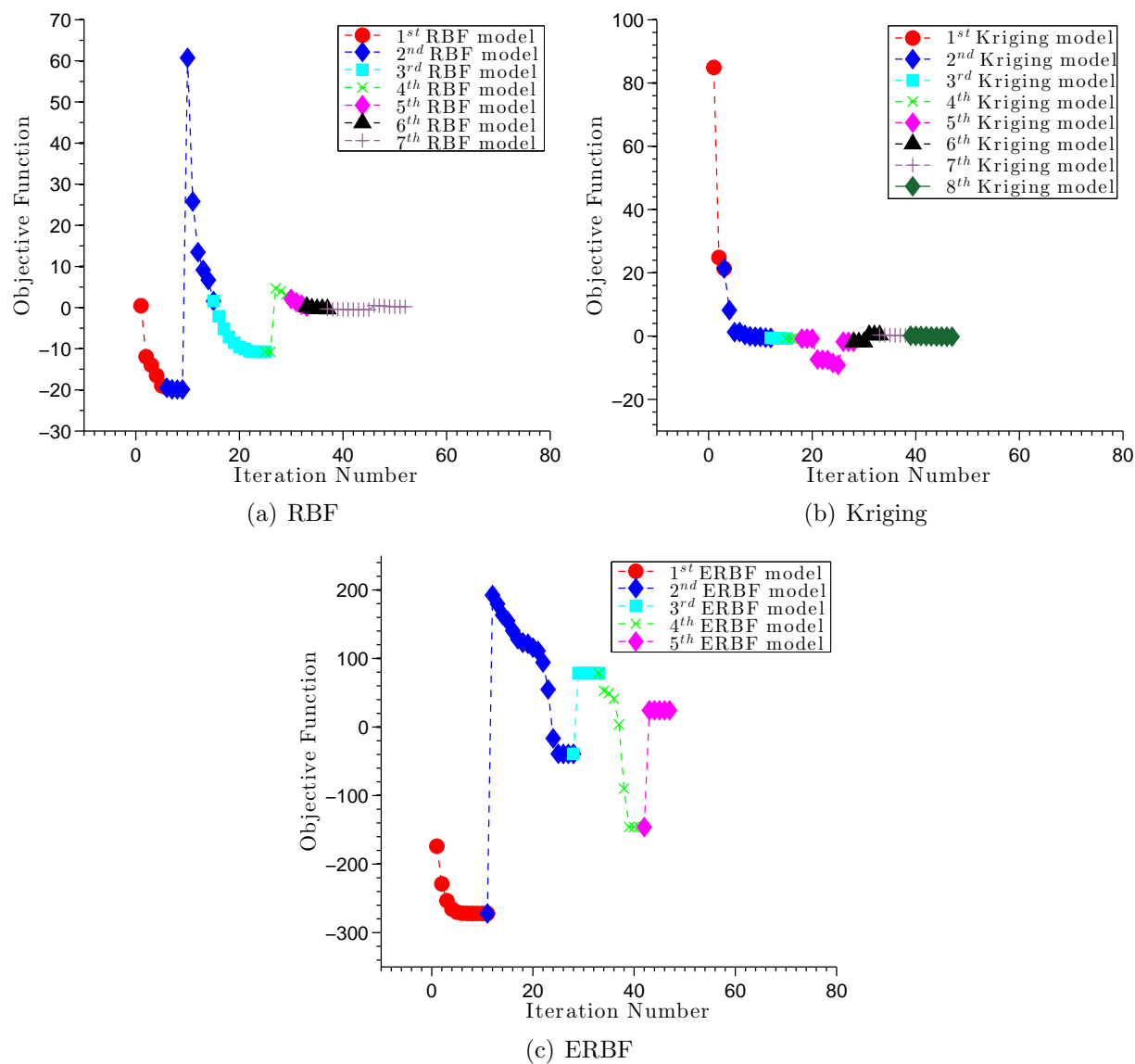


Figure 7.6: Rosenbrock function (5-variable): Optimization history of the global optimization problem

7.5 Chapter Summary

This chapter presented a novel surrogate-based optimization (SBO) method that is implemented in population-based optimization algorithms to provide high fidelity optimum designs at reasonable computational expense. The proposed method, called Adaptive Model Refinement (AMR), is designed to refine the surrogate model by sequentially adding a batch of new samples at any given iteration of SBO when a switching model metric is met. The op-

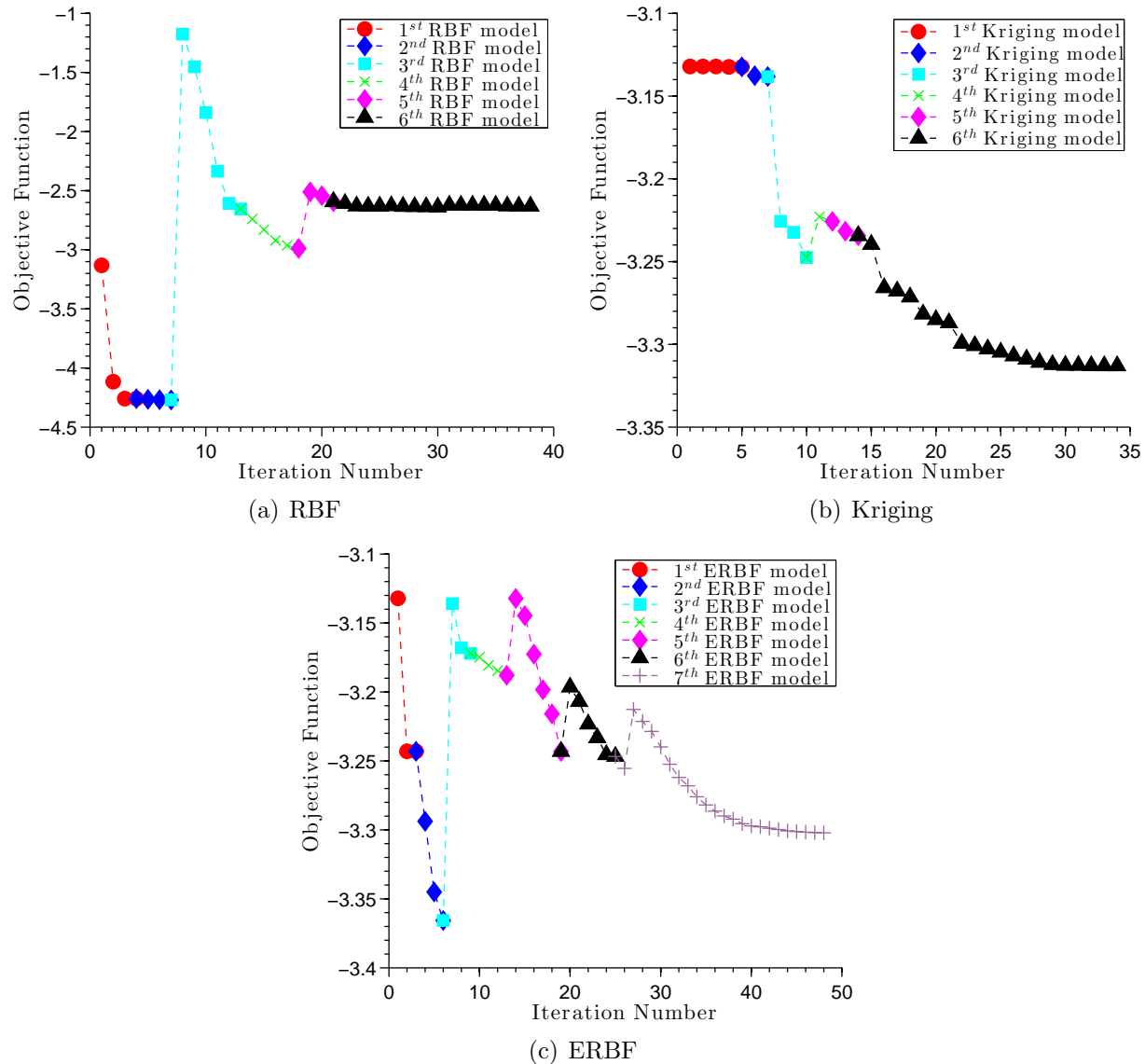


Figure 7.7: Hartmann function (6-variable): Optimization history of the global optimization problem

imization process is started using the initial surrogate model constructed over an initial set of uniform sample points. A novel model-independent metric is then used to determine when to refine the current surrogate model during the optimization iterations. The uncertainty in the current surrogate model output and the batch size of the new samples is determined by Predictive Estimation of Model Fidelity (with certain modifications). Different surrogate models (e.g., Kriging, RBF, and E-RBF) are used in conjunction with Particle Swarm Op-

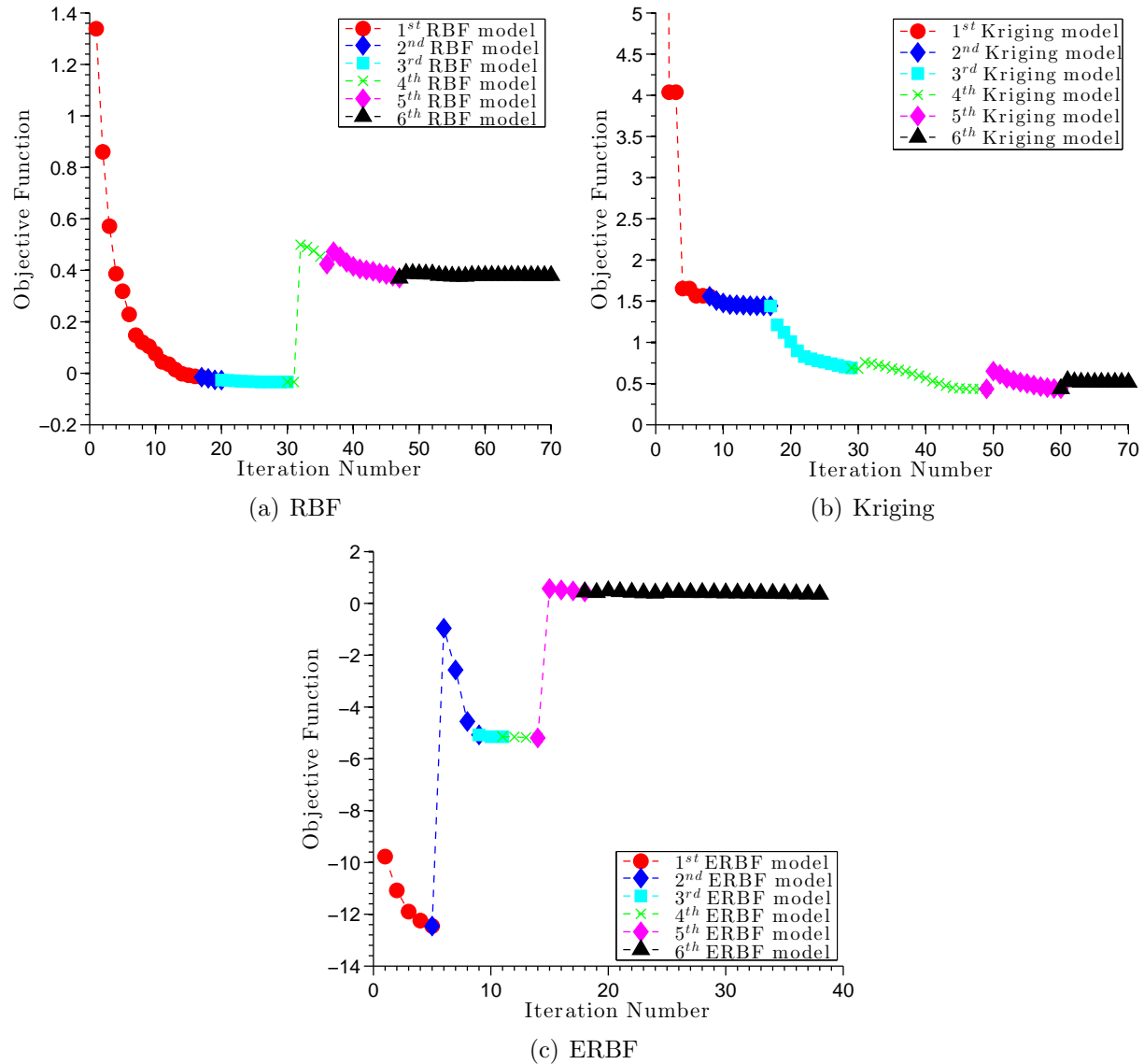


Figure 7.8: Levy function (10-variable): Optimization history of the global optimization problem

timization to investigate the performance of the new AMR method for SBO. To this end, three benchmark optimization problems are solved. The performance of the AMR method is also compared with that of the single stage sampling. The results indicate that the AMR method improves the efficiency and the accuracy of SBO over the single stage method.

Table 7.1: Optimization results using different optimization approaches

Function/Problem	Surrogate	d^{AMR}	d^{SS}	G^{AMR}	G^{SS}	G^{ANA}
Rosenbrock	Kriging	2.2859	3.2559	0.0902	1.0348e+03	
	RBF	2.2119	4.0290	0.0114	453.2797	0.0000
	E-RBF	2.6833	1.6983	23.2000	336.9315	
Hartmann	Kriging	0.0039	0.1608	-3.3220	-2.6727	
	RBF	0.2519	0.4441	-2.6279	-0.7588	-3.3223
	E-RBF	0.0521	0.5680	-3.2622	-0.1684	
Levy	Kriging	2.1783	1.8403	0.5011	2.0017	
	RBF	1.5224	2.5657	0.3532	1.3473	0.0000
	E-RBF	2.2209	4.0980	0.2110	1.8871	

d^{AMR} : the distances between the optimal design found using the AMR method and the actual optimum

d^{SS} : the distances between the optimal design found using the single stage method and the actual optimum

G^{ANA} : the (analytically found) actual optimum objective value

G^{AMR} : the the actual function estimate at the optimum design obtained under the AMR method

G^{SS} : the the actual function estimate at the optimum design obtained under the single stage method

PART IV

Practical Application:

Error Quantification, Model Selection,

Variable Fidelity Optimization

CHAPTER 8

Application of PEMF-derived Methods to CFD-based Problems

What is the effectiveness of PEMF-derived method in CFD-based design problems?

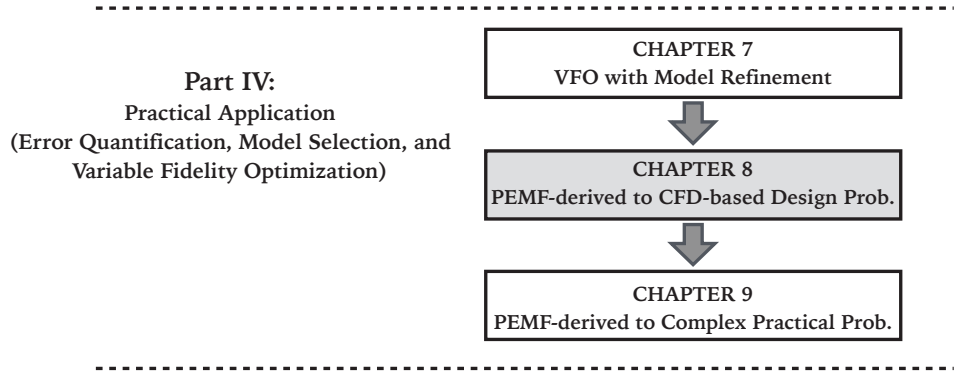


Figure 8.1: Chapter 8 orientation

8.1 Chapter Introduction

This chapter presents the application of the frameworks and approaches developed in Chapters 5, 6, and 7 of this dissertation in different CFD-based design problems. Each of these problems is presented in a stand-alone fashion so that they can be read independently. Inevitably, this involves some repetition of the parts already covered in earlier chapters of this dissertation.

The remainder of this chapter is organized as follows: Section 8.2 presents two different CFD-based problems to illustrate important aspects of the new model selection framework (COSMOS) outlined in Chapter 5. These problems include: (i) airfoil design problem, and (ii) the three-pane window heat transfer problem. In Section 8.3 the proposed model switching metric in variable fidelity optimization discussed in Chapter 6 is applied to a CFD-based design problem through Particle Swarm Optimization, which is aerodynamic shape optimization of airfoil. The chapter summary is provided in Section 9.7.

8.2 Numerical Case Studies with COSMOS

In this section, the effectiveness of the new model selection framework (**COSMOS**) developed in chapter 5 is investigated by considering a pool of models comprised of the following three popular model types: (i) Kriging, (ii) RBF, and (iii) SVR. The different forms of the kernel/basis/correlation functions currently considered in COSMOS are given in Chapter 5. COSMOS offers five different criteria for selection of optimal surrogates. Any combination of these criteria can be chosen by the user. In the current implementation, three of the most practically pertinent pairwise combinations ($E_{med}^{mo} - E_{max}^{mo}$, $E_{med}^{mo} - E_{med}^{\sigma^2}$, and $E_{med}^{mo} - E_{med,\alpha}^{mo}$) are allowed to be selected. Using more than two criteria could prove to be taxing on the complex multiobjective optimal model selection process, and is also an unlikely scenario in practice.

8.2.1 Airfoil Design Problem:

In this problem, the lift-to-drag ratio (C_L/C_D) of a Wortmann FX60.126 series airfoil [131] is represented as a function of four design variables, which include the angle of incidence (ranging from 0 to 10) and the three shape variables (each ranging from -0.01 to 0.01). As illustrated in Fig. 8.2, the three shape variables represent the distances (i) between the middle of the suction side and the horizontal axis (x_1), (ii) between the middle of pressure side and the horizontal axis (x_2), and (iii) between the trailing edge and the horizontal axis (x_3). These three shape variables allow a modification of the un-deformed airfoil profile. With respect to the initial airfoil design, two cubic splines are added to the suction and the pressure sides. Each of these splines is characterized by 3 points, defined on the leading edge, the middle span, and the trailing edge. To develop the high fidelity aerodynamics model of this airfoil, the commercial Finite Volume Method tool, FLUENT, is used, which

solves the Reynolds-averaged Navier-Stokes (RANS) equations. The chord length of the airfoil is specified to be 1.0m, and the incoming wind velocity is specified to be 25.0m/s. The structured CFD mesh is constructed using 9,838 quadrangular cells and 10,322 node points.

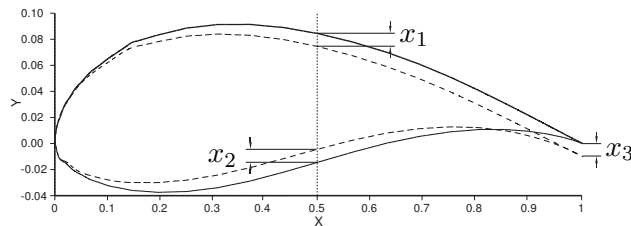


Figure 8.2: Design variables governing the shape geometry of the Airfoil

8.2.2 Three-Pane Window Heat Transfer Problem:

The performance of a Three-pane window varies with climatic conditions [132]. The heat transfer through the window is a function of environmental conditions and the window material and dimensions. A simplified schematic and the geometrical dimensions of the Three-pane window is shown in Fig. 8.3. The heat transfer simulation model of the side channels and the air gap is created using the computational fluid dynamics (CFD) software Fluent. The model simulates the steady-state heat transfer process. In this problem, the middle tinted pane is made of a generic bronze glass, and the other two panes are made of clear glass. To reduce the computational expense of the window heat transfer model, a surrogate model is used to represent the heat flux through the inner pane, \dot{Q}_w . The inputs for the surrogate model are (i) the atmospheric temperature, (ii) the wind speed, and (iii) the solar radiation.

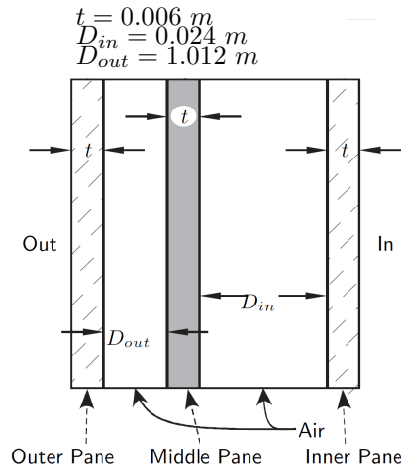


Figure 8.3: Schematic of the three-pane window [132]

Table 8.1: Three-pane window geometry

Property	Symbol	Value	Units
Height of the glass	h_g	1.000	m
Width of the glass	w_g	0.500	m
Inner gap depth	D_{in}	0.024	m
Outer gap depth	D_{out}	1.012	m
Pane thickness	t	0.006	m

8.2.3 Results and Discussion

The sample size used for training the surrogates and the settings specified for optimal model selection in each application are summarized in Table 8.2. The numerical settings for the implementation of *PEMF* in the *COSMOS framework* are also provided in Table 8.2, which lists (i) the number of iterations, and (iv) the size of training points at each iteration (as a function of iteration). In this section, the Elitist Non-dominated Sorting Genetic Algorithm

Table 8.2: Problem properties and optimization settings

Problem	Problem Properties		Optimization Settings		PEMF Settings	
	Dimension	Sample Size	Population Size	Maximum No. of Generations	No. of iterations, N^{it}	No. of training points at each iteration, n^t
Airfoil design	4	30	30	60	4	$19+2t$
Three-pane window	3	30	30	60	4	$19+2t$

(NSGA-II) [124] is applied to perform the hyper-parameter optimization in the *Cascaded* technique, and to solve the mixed integer nonlinear programming problem formulated in the *One-Step* technique.

Table 8.3: The set of Pareto Surrogate Models given by (i) Cascaded technique, and (ii) One-Step technique

Problem	Cascaded technique	One-step technique
Airfoil Design	- <i>RBF (Cubic)</i> - RBF (Gaussian) - RBF (Multi-quadric)	- <i>RBF (Cubic)</i> - RBF (Gaussian) - RBF (Multi-quadric)
Three-pane window	- Kriging (Gaussian) - <i>RBF (Cubic)</i>	- Kriging (Gaussian) - <i>RBF (Cubic)</i>

The final optimal solutions, including the best trade-offs between the median and the maximum errors (in the Φ_0 , Φ_1 , and Φ_2 classes) for these two CFD-based test problems are illustrated in Figs. 8.4, 8.5, and 8.6-8.7. Each figure shows the results from a particular test for a particular problem. In each of these figures, the 'O', the 'X', and the '+' symbols respectively represent the Pareto solutions of model selection under Φ_0 , Φ_1 , and Φ_2 classes (i.e., different hyper-parameter classes). The final Pareto optimal solutions in these figures are encircled by square symbols, and are derived by applying a Pareto Filter.

Three different model selection tests are performed using COSMOS, each involving a unique pair of selection criteria, as listed below:

TEST I: Apply COSMOS to (i) minimize the modal value of the median error E_{med}^{mo} , and (ii) minimize the modal value of the maximum error E_{max}^{mo} ;

TEST II: Apply COSMOS to (i) minimize the modal value of the median error E_{med}^{mo} , and (ii) minimize the standard deviation of the median error $E_{med}^{\sigma^2}$;

TEST III: Apply COSMOS to (i) minimize the modal value of the median error E_{med}^{mo} , and (ii) minimize the expected modal value of the median error at 20% more number of sample points $E_{med,\alpha}^{mo}$, $\alpha = 0.2$;

TEST I: ($E_{med}^{mo} - E_{max}^{mo}$)

The Pareto optimal surrogate models obtained by the *Cascaded* and the *One-Step* techniques under Test I are listed in Table 8.3. For each of the test problems, the Pareto optimal solutions with the lowest values of the median error and the maximum error are respectively written in **boldface** and *italic*.

For the Airfoil design problem, it is observed from Table 5.5 that in both the *Cascaded* and the *One-Step* techniques, the *RBF with the Multiquadric and Gaussian basis functions* are respectively the best models in terms of the median and the maximum errors. For the Airfoil design problem, the optimal model selection yielded by *COSMOS* perfectly matches that given by the Actual error (Table 5.5), although the Pareto solutions occur at different regions of the objective space. In this case, again a single model type (*RBF*) with different kernels (*Cubic*, *Gaussian*, and *Multiquadric*) constitutes the set of Pareto models. It is also observed from Fig. 8.4(b) that the Pareto solution from the Φ_0 class (*RBF-Cubic*), appearing as the lower anchor point offers an attractive trade-off at a small compromise in the median error.

In the Three-pane window problem, the same model-kernel combination (*Kriging-Gaussian*) provides all the best trade-off solutions under the *One-Step* technique. In contrast, under the *Cascaded* technique, the lowest median error is accomplished by the *RBF-Cubic* model. This disagreement can be attributed to the very small difference between the median errors estimated for the *Kriging-Gaussian* and the *RBF-Cubic* models for this problem.

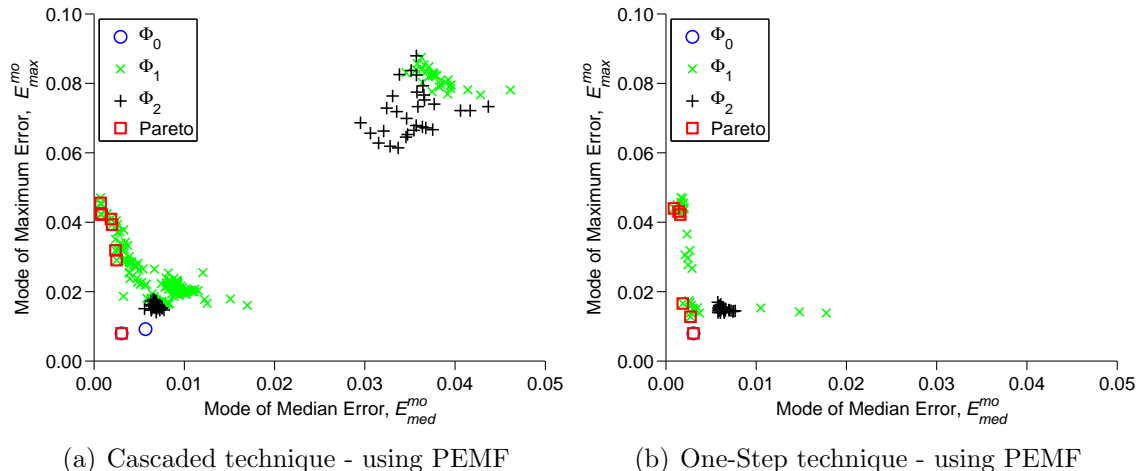


Figure 8.4: Trade-offs between modal values of median and maximum error - Airfoil Design: Pareto models and final population of models from all Φ classes

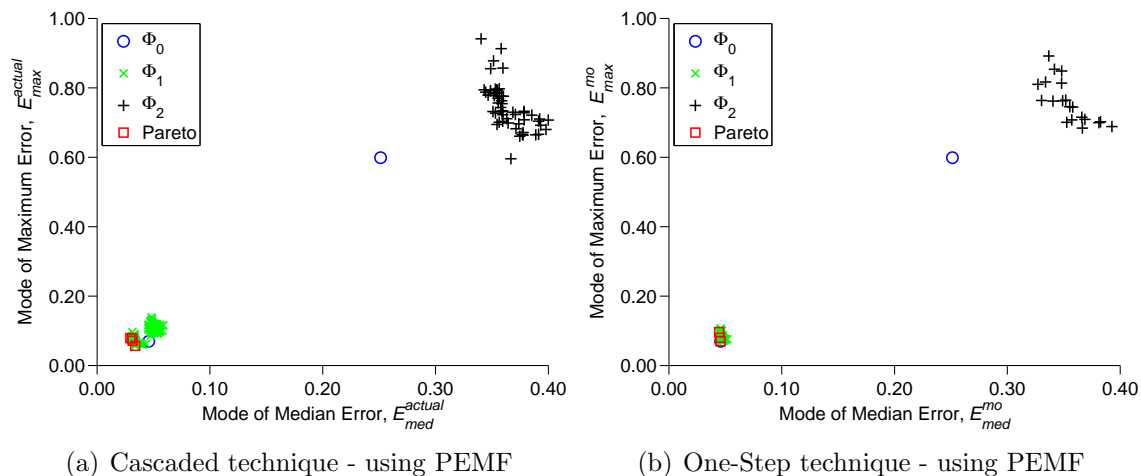


Figure 8.5: Trade-offs between modal values of median and maximum error - Window model: Pareto models and final population of models from all Φ classes

Next the performance of the COSMOS method is investigated to show that how it performs better than simply selecting the kernel functions and the hyper parameter values recommended in the literature. The median and the maximum errors of the models selected using COSMOS are therefore compared with the errors of (i) the Kriging model with the Gaussian correlation function (where the correlation parameter is obtained using the MLE

Table 8.4: Comparison of the performance of the COSMOS with the current paradigm in engineering problems

Problem	Median Error					Maximum Error				
	Kriging ¹	RBF ²	SVR ³	COSMOS		Kriging ¹	RBF ²	SVR ³	COSMOS	
				Model	Error				Model	Error
Airfoil Design	0.007	0.005	0.05	RBF Multiquadric $\sigma = 0.274$	0.0015	0.017	0.041	0.08	RBF Cubic	0.008
Window model	0.06	0.24	0.44	Kriging Gaussian $\theta = 0.1$	0.044	0.09	0.59	0.51	RBF Cubic	0.069

¹ *Kriging-Gaussian*. Correlation parameter is obtained using a MLE technique [35]

² *RBF-Multiquadrics*. Shape parameter is set to $\sigma = 0.9$ [28]

³ *SVR-Linear*. Penalty parameter is set to $C = 50$ [122]

technique [35], (ii) the RBF with a multiquadrics basis function (where the shape parameter is set to $\sigma = 0.9$ [28], and (iii) support vector regression (SVR) with a linear kernel function (where the penalty parameter is set to $C = 50$ [122]). The median and the maximum errors of the models in each cases for these two practical problems are illustrated in Table. 8.5.

It is observed that, for the Air foil design problem, COSMOS offers models which have 70% lower median error and 20% lower maximum error, respectively, than the next best models (obtained by RBF and Kriging). In the Window model, the models selected using COSMOS have 26% and 23% lower median and maximum error than the next best model (obtained by Kriging).

TEST II ($E_{med}^{mo} - E_{med}^{\sigma^2}$) and TEST III ($E_{med}^{mo} - E_{med,\alpha}^{mo}$)

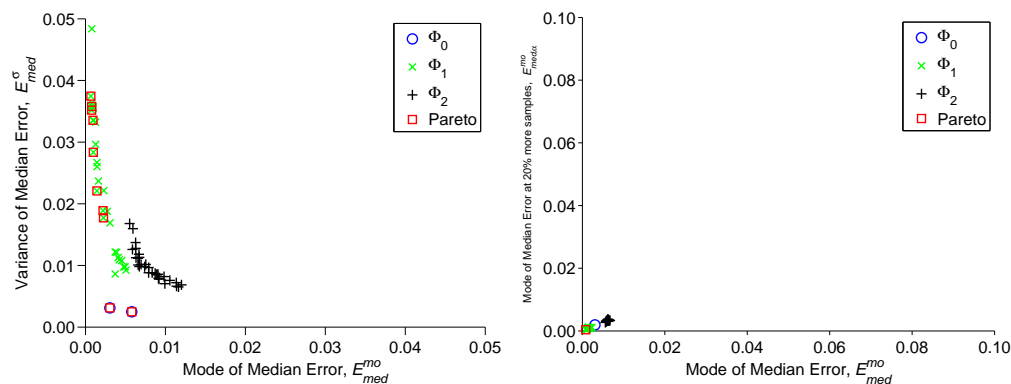
The surrogate models in the Pareto Optimal set for the Airfoil design problem and Three-pane window problem (under Test II and Test III) are listed in Table 8.5. The Pareto optimal solutions as well as the solutions in the final population for these CFD-based problems are shown in Figs. 8.6 to 8.7. Again, each figure shows the results from Test II and Test III for a particular problem. In each of these figures, the blue 'O', the green 'X', and the black '+' symbols respectively represent the final results of model selection under Φ_0 ,

Φ_1 , and Φ_2 classes. The final Pareto optimal solutions in these figures are encircled by red square symbols.

Table 8.5: The set of Pareto Surrogate Models given by COSMOS

Problem	Test II, $\min[E_{med}^{mo}, E_{med}^{\sigma^2}]$	Test II, $\min[E_{med}^{mo}, E_{med,\alpha}^{mo}]$
Airfoil Design	RBF (Linear), RBF (Cubic), RBF (Multiquadric)	RBF (Multiquadric)
Window model	Kriging (Spherical), Kriging (Gaussian)	Kriging (Gaussian)

For the Airfoil design problem, RBF-Linear, RBF-Cubic, and RBF-Multiquadric constituted the set of Pareto models in Test II (Table 8.5 and Fig. 8.6(a)); and RBF-Multiquadric was selected as the single optimum surrogate model in Test III (Table 8.5 and Fig. 8.6(b)). In the Three-pane window problem; in Test II where the variance of the median error is considered, Kriging-Spherical and Kriging-Gaussian are included in Pareto optimal set (from Fig. 8.7(a)). In contrast, Kriging-Gaussian is the sole optimum model in Test III for the Three-pane window problem (from Table 8.5 and Fig. 8.7(b) and).



(a) Trade-offs between modal values and (b) Trade-offs between modal values of median error at N and $1.2N$ samples

Figure 8.6: Airfoil Design: Pareto models and final population of models from all Φ classes

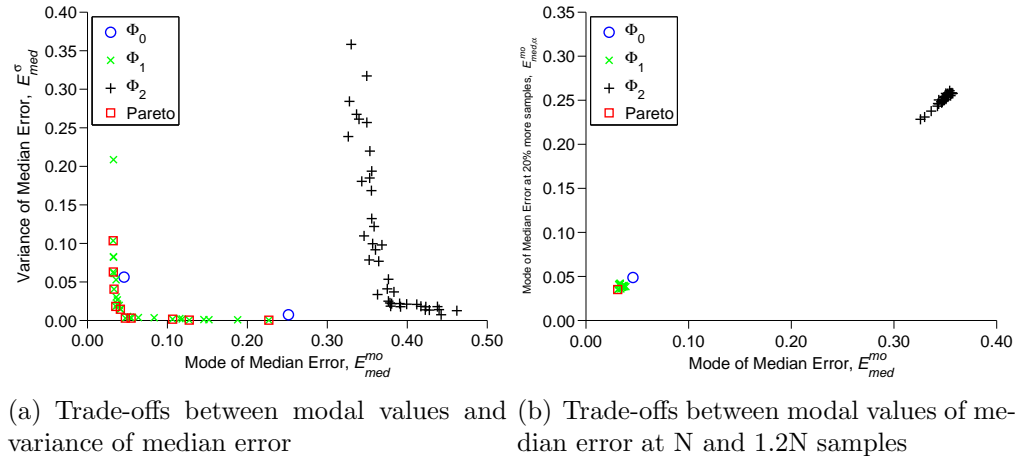


Figure 8.7: Window model: Pareto models and final population of models from all Φ classes

8.3 Numerical Case Study with Adaptive Model Switching

In this section, the proposed AMS-based optimization developed in chapter 6 is applied to the Airfoil design problem through Particle Swarm Optimization. This application illustrates: (i) the computational advantage of this method over purely high fidelity model-based optimization, and (ii) the accuracy advantage of this method over purely low fidelity model-based optimization.

8.3.1 Aerodynamic shape optimization of 2D airfoil

This section describes a 2D airfoil design problem where the ratio of the coefficients of lift and drag (C_L/C_D) of the Wortmann FX60.126 2D airfoil [131] is to be maximized. The lift-to-drag ratio (C_L/C_D) is expressed as a function of four design variables, which include the angle of incidence (ranging from 0 to 10) and the three normalized shape variables (each ranging from -0.01 to 0.01). As illustrated in Fig. 8.2, the three shape variables define the distances (i) between the middle of the suction side and the horizontal axis (x_1), (ii) between the middle of pressure side and the horizontal axis (x_2), and (iii) between the trailing edge and

the horizontal axis (x_3). These three shape variables allow a modification of the un-deformed airfoil profile. With respect to the initial airfoil design, two cubic splines are added to the suction and the pressure sides. Each of these splines is characterized by 3 points, defined on the leading edge, the middle span, and the trailing edge. The chord length of the airfoil is equal to 1 m . The design constraints are the side constraints on the design variables which are listed in Table 8.6.

Table 8.6: Design variables in airfoil optimization problem

Description	Notation	Lower limit	Upper limit
Distance between the middle of suction side and horizontal axis	x_1	-0.01	0.01
Distance between the middle of pressure side and horizontal axis	x_2	-0.01	0.01
Distance between the trailing edge and horizontal axis	x_3	-0.01	0.01
Incidence angle	x_4	0°	10°

8.3.2 Aerodynamic models with different level of fidelity

To develop a high fidelity aerodynamic model for determining C_L and C_D (M_{HF}^{A}), the commercial Finite Volume Method package, FLUENT, is used. The Reynolds-averaged Navier-Stokes (RANS) formulation is used along with a Reynolds model to represent the turbulence. The CFD mesh is constructed using quadrangular cells[131], characterized by 9,838 quadrangular cells and 10,322 grid points (Fig. 8.8(a)).

The low fidelity physics-based model ($M_{\text{PLF}}^{\text{A}}$) is constructed based on the assumptions that the fluid is steady, incompressible, and irrotational. In this model, the Navier-Stokes equations are solved using the Finite Element method. Triangular T3 elements used for demonstration, as shown in Fig. 8.8(b). The incoming velocity in the analysis is set to 25 m/s . The computational time of the High and Low fidelity physics-based models are approximately 300 seconds and 30 seconds, respectively (i.e., an order of magnitude apart). The pressure field around the airfoil for the low and high fidelity aerodynamic models at a

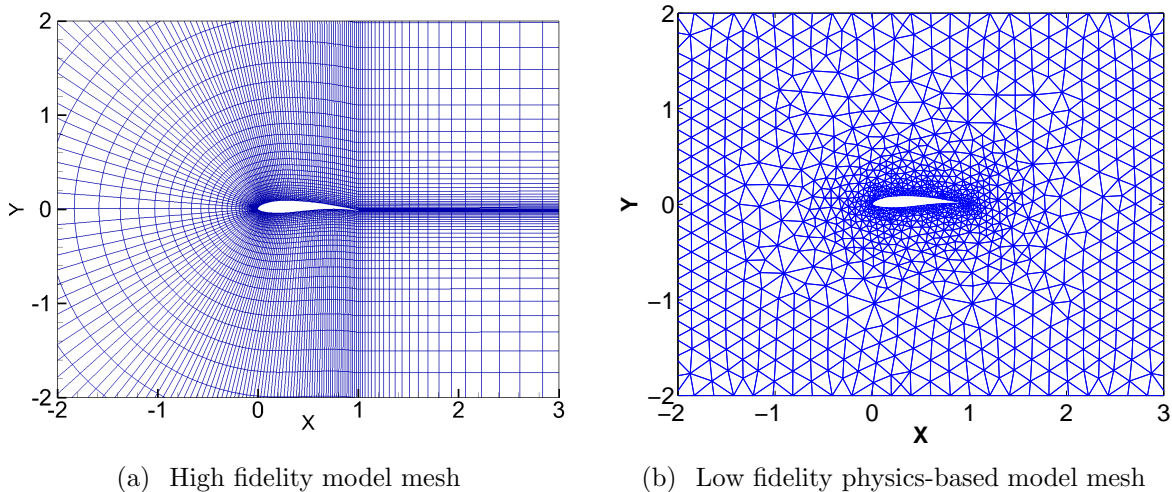


Figure 8.8: Fine and coarse mesh for CFD of airfoil [131]

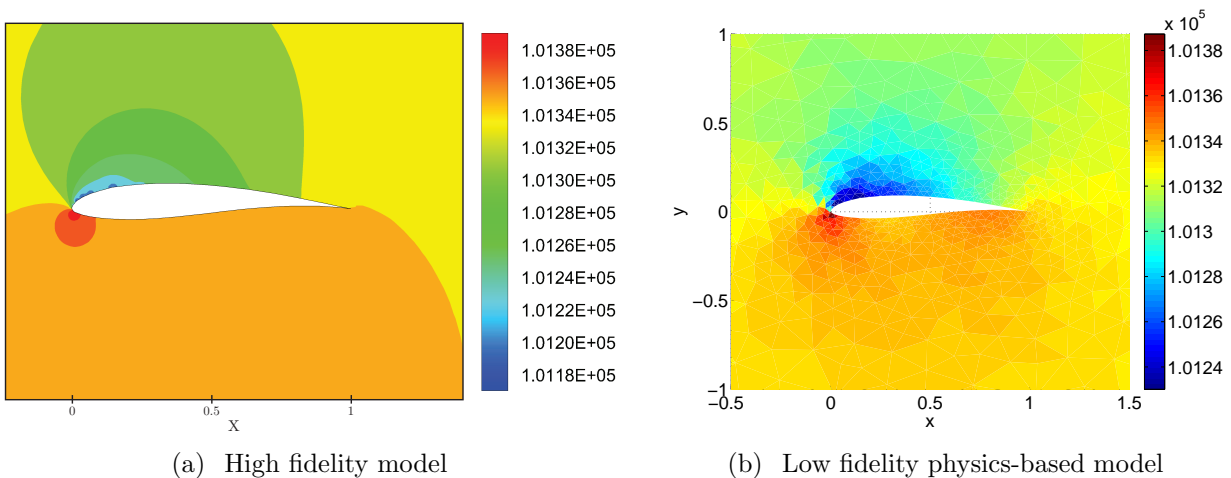


Figure 8.9: Pressure field around the airfoil at a baseline design

baseline design ($x_1 = 0$, $x_2 = 0$, $x_3 = 0$, and $x_4 = 5^\circ$) are illustrated in Fig. 8.9.

The third model is a surrogate model (M_{SM}^A) constructed using a DoE of high fidelity evaluation involving 30 sample points. The fourth model is a tuned low fidelity model (M_{TLF}^A). In this article, the tuned low fidelity model is constructed using the *Multiplicative approach*, as given by

$$\tilde{F}(x, a) = f(x) \times C(x) \quad (8.1)$$

where \tilde{F} is a tuned low fidelity model; $f(x)$ is a low fidelity model; $C(x)$ is an explicit tuning surrogate constructed using the high fidelity samples, as shown below:

$$C(x) = \frac{\frac{C_L}{C_D}|_{\text{HF}}}{\frac{C_L}{C_D}|_{\text{PLF}}} \quad (8.2)$$

where C_L and C_D are respectively the lift and drag coefficients.

The surrogate model (M_{SM}^{A}) and the surrogate component of the tuned low fidelity model ($M_{\text{TLF}}^{\text{A}}$) are both constructed using Kriging with a Gaussian correlation function [8, 9]. Kriging is an interpolating method that is widely used for representing irregular data. Under the Kriging approach, the zero-order polynomial function is used as a regression model. In this article the Optimal Latin Hypercube is adopted to determine the locations of the sample points. The PEMF method is then applied to estimate the error in the surrogate models constructed using the high fidelity responses, and the low fidelity model. To estimate the error in the physics-based low fidelity FEA model, the inverse assessment process defined in Section 6.2.3, is applied. Figures 8.10(a)-8.10(c) illustrate the distributions of the error in the tuned low fidelity model, the surrogate model, and the physics-based low fidelity model. It is observed from Fig. 8.10 that the accuracy of the physics-based low fidelity model is less than that of the surrogate model. It is also readily evident that the computational cost of the physics-based low fidelity model is more than that of the surrogate model. Therefore, in this problem, the physics-based low fidelity model is not included as a model choice in the variable fidelity optimization.

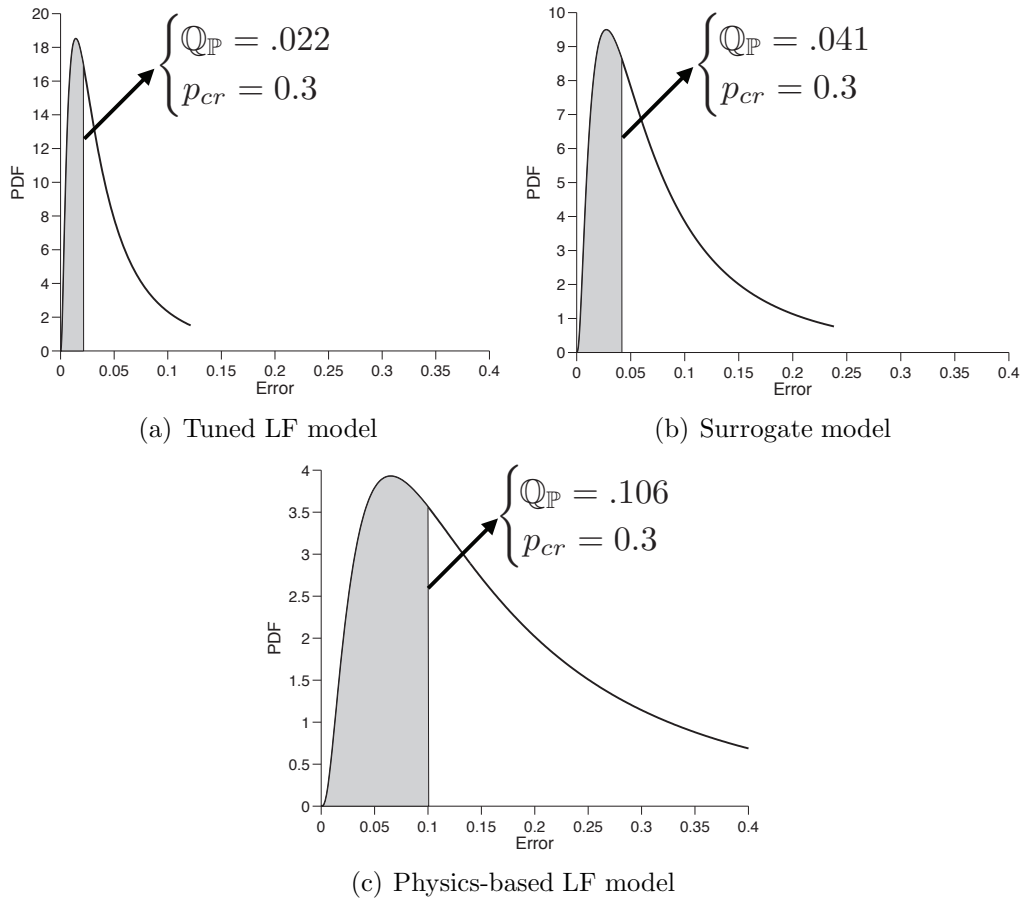


Figure 8.10: Distribution of the model errors: aerodynamic C_L/C_D ratio of the 2D airfoil

8.3.3 Results and Discussion

In the airfoil optimization problem, the initial population of particles is generated using the fastest model, which is the surrogate model. The AMS technique adaptively switches the model type twice during optimization (over a total of 22 iterations), resulting in an optimum design with a high fidelity function estimate.

The model types, the error distribution parameters associated with each model, and the number of calls made to each model in this optimization are listed in Table 8.7. The total number of calls made to each model is equal to the product of the particle population and the number of iterations during which that particular model is used for system evaluation.

In this problem, the AMS technique is applied at every iteration $\delta = 1$.

Table 8.7: Models with different levels of fidelity used in the airfoil optimization problem (the high fidelity model is assumed to be a true representation of the system behavior)

Model	Location parameter μ	Scale parameter σ	$\mathbb{Q}(p_{cr})$ $p_{cr} = 0.3$	Number of calls made $N_{pop} \times No. of Iter.$
Surrogate model	-2.6793	0.9628	0.0414	30×13
Tuned low fidelity model	-3.3197	0.9547	0.0219	30×6
High fidelity model	-	-	-	30×3

Figure 8.11(a)-8.11(f) illustrate the distribution of the fitness function improvement at different iterations during the optimization process. In these figures, \mathbb{Q} represents $(1 - p_{cr})$ -quantile of the Θ distribution. The error distributions of the surrogate model and the tuned low fidelity model, which are determined apriori, are also shown in these figures. Through AMS, model switching, from surrogate model to tuned low fidelity model and from tuned low fidelity model to high fidelity model, occur at the 14th and the 20th iteration, respectively.

The convergence history of the airfoil optimization is illustrated in Fig. 8.12. This figure also indicates which model is active at each iteration. It is observed that, from the first iteration till the 13th iteration the surrogate model (M_{SM}^A) is active, before switching to the tuned low fidelity model that remains active till the 19th iteration. Interestingly, most of the objective function improvement occurs under the tuned low fidelity model (more than 10% increase in the CL/CD ratio). The final switching event, from the tuned low fidelity to the high fidelity model, occurs at the 19th iteration. The optimization progresses with the high fidelity model for another 3 iterations before reaching convergence. In this case, the algorithm converges by satisfying the predefined function tolerance, $\delta f = 1e - 5$.

Next, the performance of the AMS method is investigated to show how it performs better than simply using the individual low fidelity models for optimization or running the

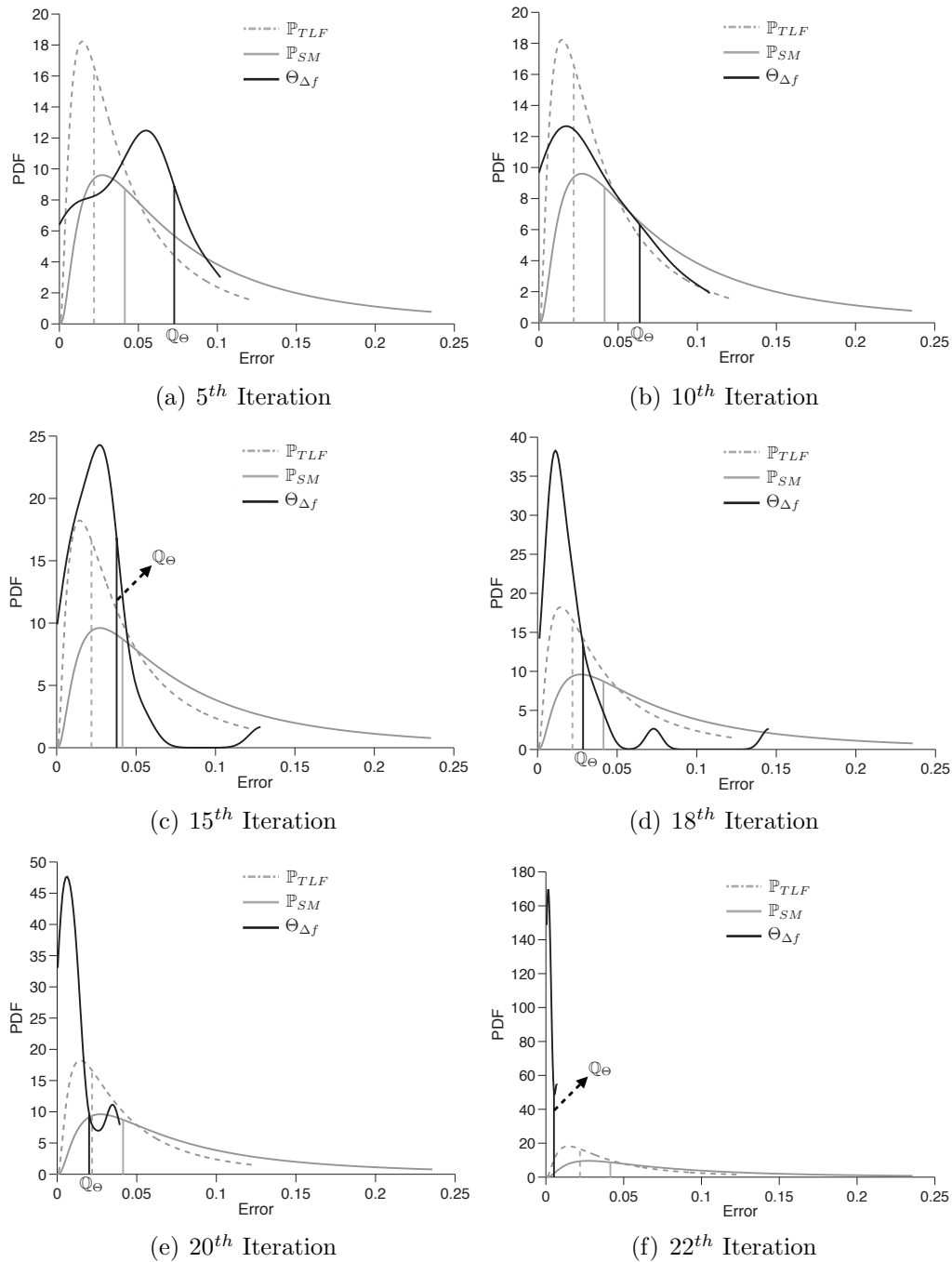


Figure 8.11: Distribution of the fitness function improvements in different iterations of the airfoil optimization with PSO-AMS (also showing the model error distributions)

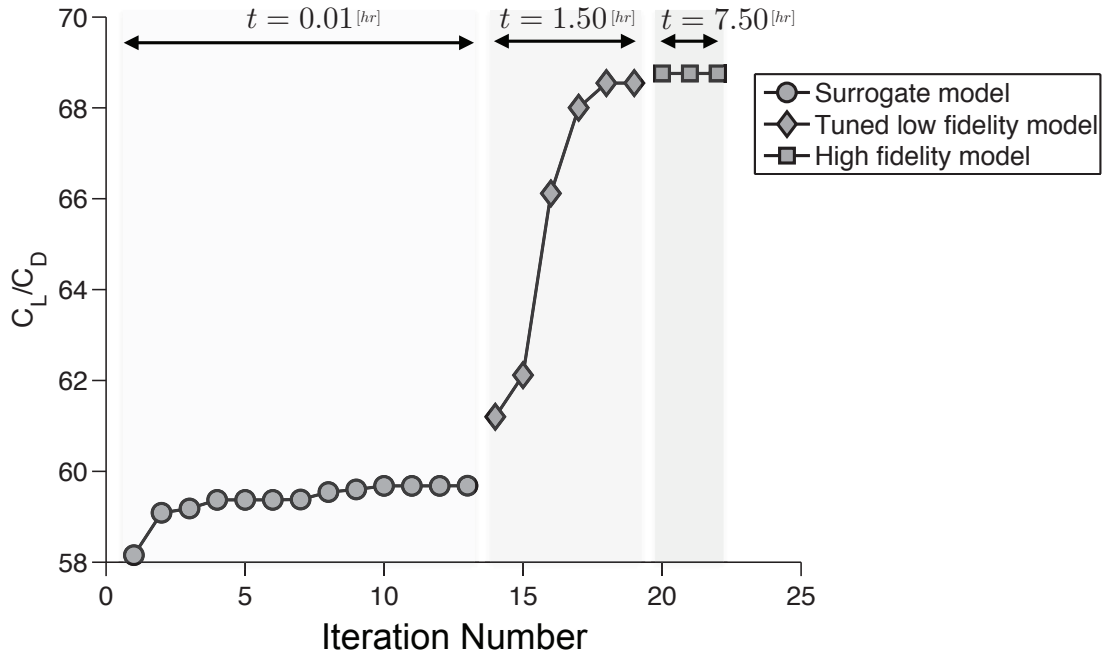


Figure 8.12: Optimization history of the airfoil design problem

optimization solely on the high fidelity model. The results yielded by the PSO-AMS are therefore compared with the results yielded by running MDPSO solely using the surrogate model (PSO-SM), solely using the tuned low fidelity model (PSO-TLF), and solely using the high fidelity model (PSO-HF). The optimum results thus obtained, the computational cost, and the total number of function evaluations in each case are reported in Table 8.8. The final column of this table shows the high fidelity function estimate at the optimum design obtained under each optimization run (e.g., $y_{HF}^*(x_{SM}^*)$ and $y_{HF}^*(x_{TLF}^*)$). It is observed that the PSO-AMS not only requires 185% less computing time compared to PSO-HF, it also provides the best optimum value that is 5% better than the next best value (where the 2nd best is obtained by PSO-TLF). It is also observed that, in the PSO-TLF approach, the optimum is located in the region where the TLF model has more than 8% error. This optimum is in the vicinity of the high fidelity optimum yielded by the AMS method. The optimization performed solely using the tuned low fidelity model (PSO-TLF) incurs a higher

computational cost in comparison with that performed using the AMS method, which is attributed to the high number of function evaluations invested to satisfy the termination criterion ($\delta f = 1e - 5$) in the former.

Table 8.8: 2D Airfoil Design: Optimization results using different optimization approaches

Approach	x_1^*	x_2^*	x_3^*	x_4^*	Optimum function C_L/C_D (f^*)	Model in last iteration	Computational time[hr] over Function evaluation	HF response at optimum ($f_{HF}(x^*)$)
PSO-SM	0.0003	0.0003	0.0003	4.8300	59.69	SM	0.275/990	59.43
PSO-TLF	-0.0028	-0.0058	-0.0014	2.7911	70.54	TLF	9.96/1380	65.20
PSO-HF	7.14E-5	-0.0021	-0.0018	4.8273	59.57	HF	25.7/360	59.57
PSO-AMS	-0.0020	-0.0004	-0.0009	2.8313	68.75	HF	9.01/660	68.75

PSO-SM: optimization performed by MDPSO solely using the surrogate model

PSO-TLF: optimization performed by MDPSO solely using the tuned low fidelity model

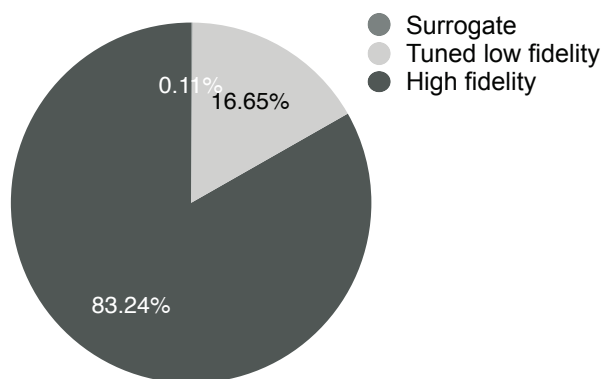
PSO-HF: optimization performed by MDPSO solely using the high fidelity model

PSO-AMS: optimization performed by MDPSO using AMS

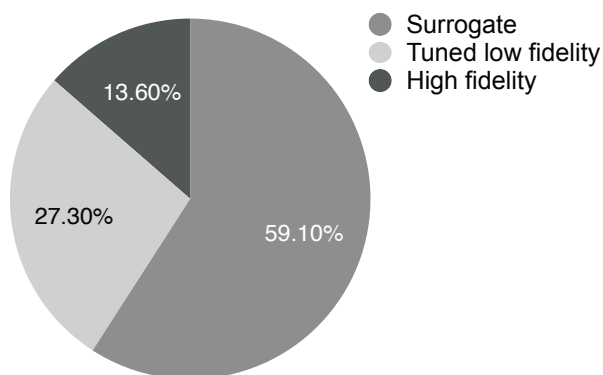
Figures 8.13(a) and 8.13(b) illustrate the contribution, in terms of computing time and function evaluations, of the three different models in the airfoil design optimization performed by PSO-AMS. These figures show that the overall computational cost is highly sensitive to the number of high fidelity model evaluations, which is expected. It is also observed that the surrogate model dominates the optimization process in terms of function calls, while the computational expense of this model is significantly lower than that of the tuned low fidelity and the high fidelity models. This observation supports the hypothesis that a probabilistic AMS technique can provide a significantly better balance between accuracy of the optimum and computational efficiency, compared to purely low fidelity or purely high fidelity optimizations.

8.4 Chapter Summary

This chapter presents the application of the frameworks and approaches developed in Chapters 5, 6, and 7 of this dissertation in different CFD-based design problems.



(a) Computational Time



(b) Function Evaluation

Figure 8.13: Percentage-contribution of each model in the airfoil optimization problem performed by PSO-AMS

In the first section of this chapter, COSMOS framework was applied to two practical CFD-based design problems (Airfoil design and Three-pane window heat transfer problems). The pool of candidate models tested include three major surrogate model types (Kriging, RBF, and SVR), with different kernel types for each candidate model. Widely different sets of surrogates models were selected as the optimum set in the these problems. In some of the cases, a diverse pool of model-kernel combinations were observed to form the Pareto optimal set, whereas in the other cases the Pareto front was comprised of the same model-kernel combination with different Hyper-parameter values. Other observations that emphasize the need for model-independent and automated surrogate model selection (COSMOS), in order to truly bring surrogate modeling into mainstream engineering design, are as follows: (i)

There is often a significant spread of the trade-offs between the median and the maximum errors; and (ii) In some cases due to the nature of the Pareto front (elbow front), a single or few optimal models can be considered as the most attractive trade-offs from a practical perspective.

In the second part of this chapter, the new adaptive model switching technique (AMS) developed in chapter 6 is applied to 2D Airfoil design problem. A powerful version of the Particle Swarm Optimization (mixed-discrete PSO) algorithm is also used to implement and investigate the performance of AMS. The results indicate that AMS along with Mixed Discrete PSO improve the efficiency of the optimization process significantly when compared to optimization solely using high fidelity models with up to 185% reduction in computing time, while reaching at the same or a better optimum. The value of the optimum with AMS is also better than that accomplished using only single low fidelity models for optimization.

CHAPTER 9

Application of PEMF-derived Methods to a Complex Practical Problem

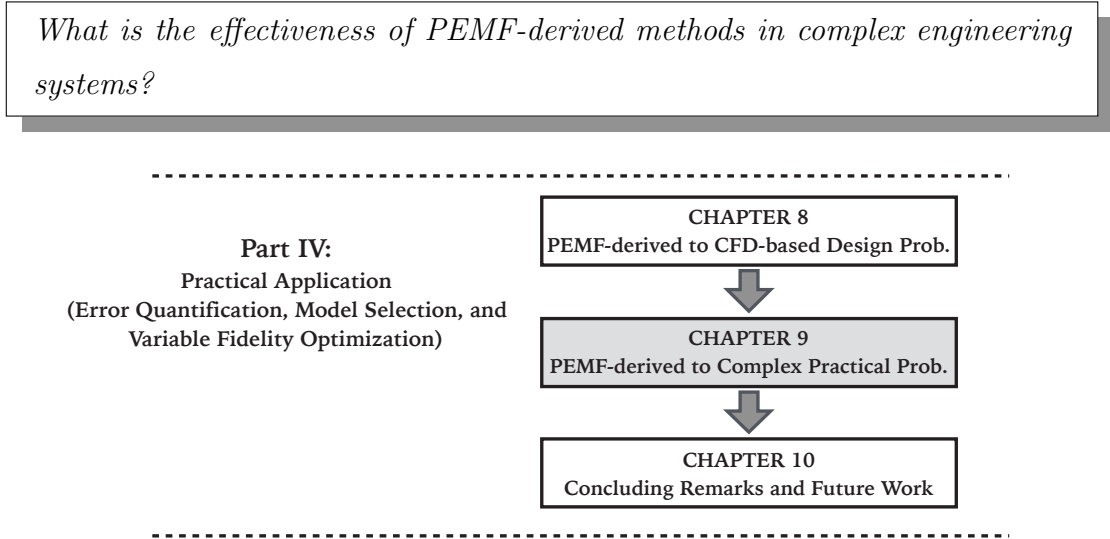


Figure 9.1: Chapter 9 orientation

9.1 Chapter Introduction

This chapter presents the application of the frameworks and approaches developed in Chapters 5, 6, and 7 of this dissertation in a complex engineering system. This problem is in the area of energy sustainability, and is presented in a stand-alone fashion so that it can be read independently. Inevitably, this involves some repetition of the parts already covered in earlier chapters of this dissertation.

The remainder of this chapter is organized as follows: Section 9.2 presents the explanation and the formulation of the Very Large Utility-scale Wind Farm. In Section 9.5, the COSMOS framework is applied to select the best surrogate model to quantify the average energy production of the very large 500-turbine wind farms. In Section 9.6, the AMR approach is used to maximize to maximize the average annual energy production of the farm through layout optimization. The chapter summary is provided in Section 9.7.

9.2 A Very Large Utility-scale Wind Farm

Wind farm layout optimization (WFLO) is the process of optimizing the location of turbines in a wind farm site, with the possible objective of maximizing the energy production or minimizing the average cost of energy. Conventional WFLO methods not only limit themselves to prescribing the site boundaries, they are also generally applicable to designing only small-to-medium scale wind farms (<100 turbines). Large-scale wind farms entail greater wake-induced turbine interactions, thereby increasing the computational complexity and expense by orders of magnitude. In this chapter, we further advance the Unrestricted WFLO framework by designing the layout of large-scale wind farms with 500 turbines (where energy production is maximized). First, the high-dimensional layout optimization problem (involving $2N$ variables for a N turbine wind farm) is reduced to a 6-variable problem through a novel mapping strategy, which allows for both global siting (overall land configuration) and local exploration (turbine micrositing). Secondly, a surrogate model is used to substitute the expensive analytical WF energy production model; the high computational expense of the latter is attributed to the factorial increase in the number of calls to the wake model for evaluating every candidate wind farm layout that involves a large number of turbines. The powerful *Concurrent Surrogate Model Selection (COSMOS)* framework (from Chapter 5) is applied to identify the best surrogate model to represent the wind farm energy production as a function of the reduced variable vector. To accomplish a reliable optimum solution, the surrogate-based optimization (SBO) is performed by implementing the Adaptive Model Refinement (AMR) technique within Particle Swarm Optimization (PSO) (developed in Chapter 7). In AMR, both local exploitation and global exploration aspects are considered within a single optimization run of PSO, unlike other SBO methods that often either require multiple (potentially misleading) optimizations or are model-dependent. By using the AMR

approach in conjunction with PSO and COSMOS, the computational cost of designing very large wind farms is reduced by a remarkable factor of 26, while preserving the reliability of this WFLO within 0.05% of the WFLO performed using the original energy production model.

9.3 Mapping of the Layout for a Large Scale Wind Farm

In this section, we present a mapping approach to quantify the layout of a very large wind farm using six mapping factors:

- (i) the maximum allowable *streamwise* and *spanwise* spacings between neighboring turbines (s_{max} and r_{max}),
- (ii) the average spacings of rows and columns (d_r and d_c),
- (iii) the normalized local radial displacement,
- (iv) the turbine rotor diameter (D),
- (v) the number of turbines (N_{turb}),
- (vi) the farm site orientation (ϕ), and
- (vii) the maximum number of rows and/or the maximum side length of the wind farm.

The maximum allowable *streamwise* and *spanwise* spacings between neighboring turbines are determined based on the size of turbine rotor diameter. The average spacings of rows and

columns are variable for difference rows and columns, and given by

$$\begin{aligned} d_r &= (r_{max} - r_{min}) \times \left\{ A\pi \frac{j}{N_{row}} - \lfloor \frac{1}{2} + (A\pi \frac{j}{N_{row}}) \rfloor + 1 \right\} + r_{min} \\ d_c &= (s_{max} - s_{min}) \times \left\{ B\pi \frac{i}{N_{column}} - \lfloor \frac{1}{2} + (B\pi \frac{i}{N_{column}}) \rfloor + 1 \right\} + s_{min} \end{aligned} \quad (9.1)$$

where s_{min} and r_{min} are the minimum *streamwise* and *spanwise*, respectively; A and B are design factors in mapping function; i and j are respectively the row and column indexes; N_{row} and N_{column} are number of rows and column, respectively. By this definition, non-uniform spacings between rows and columns are allowed.

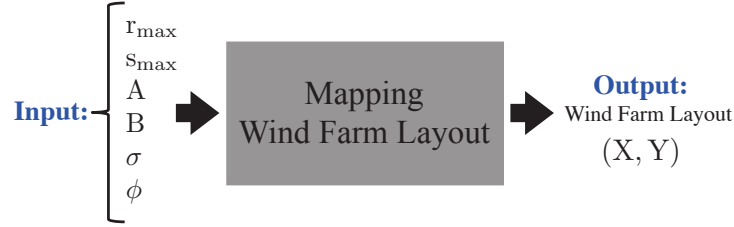
In this section, the normalized local radial displacement is expressed as the Gaussian distribution with zero mean ($\mu = 0$) and the variable standard deviation (σ) [133]. The actual radial displacement is obtained by multiplying the minimum *streamwise/spanwise* between turbines. This allows the turbines not to be restricted to be in the grid form. The side length of the wind farm can provided based on the average land usage of US commercial wind farms. Therefore, the coordinates of the turbine at the I^{th} row and the J^{th} column are given by

$$\begin{aligned} X_I &= \sum_{j=1}^J \left[(r_{max} - r_{min}) \times \left\{ A\pi \frac{j}{N_{row}} - \lfloor \frac{1}{2} + (A\pi \frac{j}{N_{row}}) \rfloor + 1 \right\} + r_{min} \right] + r_{min} \hat{r}(0, \sigma) \\ Y_J &= \sum_{i=1}^I \left[(s_{max} - s_{min}) \times \left\{ B\pi \frac{i}{N_{column}} - \lfloor \frac{1}{2} + (B\pi \frac{i}{N_{column}}) \rfloor + 1 \right\} + s_{min} \right] + s_{min} \hat{r}(0, \sigma) \end{aligned} \quad (9.2)$$

The upper and lower bounds of the mapping design factors are listed in Table 9.1. Figure 9.2 shows the input design factors and output of the layout mapping, provided in this chapter.

Table 9.1: Upper and lower bounds of mapping design factors

Design factors	Lower bound	Upper bound
r_{max}	5D	15D
s_{max}	5D	15D
A	-20	20
B	-20	20
σ	0	1

**Figure 9.2: Input and output structure of the Wind Farm Layout Mapping**

9.4 Energy Production Model

In this section, first, the wind farm power generation model is adopted from the Unrestricted Wind Farm Layout Optimization (UWFLO) framework [134] to estimate the total power extracted by a wind farm. This power generation model quantifies the wind farm power output as a function of the turbine features, the location of turbines, and the incoming wind conditions [134]. A *generalized power curve* is used to evaluate the power output of each turbine. This *generalized power curve* is scaled back to represent the approximated power response of a particular commercial turbine, using the corresponding manufacturer specifications. For Turbine- i , the power generation, P_i , can be evaluated using the following equations:

$$\frac{P_i}{P_r} = \begin{cases} P_n \left(\frac{U_i - U_{in}}{U_r - U_{in}} \right), & \text{if } U_{in} < U_i < U_r \\ 1, & \text{if } U_r < U_i < U_{out} \\ 0, & \text{if } U_{out} < U_i \text{ or } U_i < U_{in} \end{cases} \quad (9.3)$$

where U_i is the velocity immediately in front of Turbine- i . Estimation of U_i accounts for wake merging scenarios and the possibility of partial wake-rotor overlap. U_{in} , U_{out} , and U_r are respectively the turbine cut-in, cut-out, and rated speeds, reported by the turbine manufacturer. The function P_n represents a polynomial fit for the *generalized power curve*, generated using the power curve data reported for the “GE 1.5 MW xle” turbine [135].

This power generation model also allows for a variable induction factor. According to the 1-D flow assumption [136], the induction factor a and the power coefficient, C_P , can be related by

$$C_P = 4a(1 - a)^2 \quad (9.4)$$

where the power coefficient itself can be expressed as a function of incoming wind speed and turbine characteristics, as given by

$$C_P = \frac{P_i}{P_0} = \frac{P_i}{\frac{1}{8}\rho\pi D_i^2 U_\infty^3} \quad (9.5)$$

In Eq.(9.5), P_0 represents the power available from the wind; and U_∞ is the incoming wind speed at the hub height.

Subsequent solution of the non-linear equation, Eq.(9.4), gives the induction factor for each turbine based on the estimated approaching wind conditions. Thereafter, the overall power output of a N -turbine wind farm, P_{farm} , can then be given by

$$P_{farm} = \sum_{i=1}^N P_i \quad (9.6)$$

The wake effects are integrated in this model, particularly in the process of determining the effective wind speed immediately in front of any turbine (U_i). Given the predicted wake

growth, the location of turbines, and the turbine features, an influence matrix is created to determine whether a turbine is influenced by the wake of other upstream turbines for a given wind direction. As a result, the velocity immediately in front of each turbine is dynamically evaluated using a wake model, and in the same order in which the turbines encounter the wind coming from a particular direction. The Katic model [137] is used here to account for the wake merging and partial wake-rotor overlap. If Turbine- i is in the influence of multiple wakes created by K upstream turbines, the corresponding velocity deficit, v_i , is given by

$$v_i = \sqrt{\sum_{k=1}^K \frac{A_{ki}}{A_i} (u_f^{ki})^2} \quad (9.7)$$

where u_f^{ki} represents the velocity deficit in the wake (created by Turbine- k) at the location of Turbine- i ; and A_{ki} is the effective influence area of the wake (created by Turbine- k) on Turbine- i . If Turbine- i is completely in the wake of Turbine- k , $A_{ki} = A_i$; otherwise, A_{ki} denotes the overlapping area between the wake of Turbine- k and Turbine- i , estimated by standard geometrical intersection formula.

The wake model used in this section is adopted from Frandsen et al. [138]. In this model, the growth of the wake at a distance s behind any Turbine- j and the velocity deficit are respectively formulated as

$$D_{wake,j} = D (\beta + \alpha s)^{1/2} \quad (9.8)$$

and

$$u_f = \frac{U_\infty}{2} \left(1 \pm \sqrt{1 - 2 \frac{A}{A_{wake}} C_T} \right) \quad (9.9)$$

where α defines the initial wake speed deficit that must be determined experimentally; C_T is the turbine thrust coefficient; A_{wake} is the effective influence area of the wake with respect to the wake width at the current location; and the wake expansion parameter β is given by

$$\beta = \frac{1 + \sqrt{1 - C_T}}{2\sqrt{1 - C_T}} = \left(\frac{D_{eff}}{D} \right)^2, \quad s = x/D \quad (9.10)$$

For the “ \pm ” sign in Eq.(9.9), the “+” applies to cases in which the induction factor $a \leq 0.5$; while the “-” applies to $a > 0.5$. It is also noted that the Eq.(9.10) uses an effective rotor diameter, D_{eff} , to account for the near wake approximation, which is given by

$$D_{eff} = D \sqrt{\frac{1 + \sqrt{1 - C_T}}{2\sqrt{1 - C_T}}} \quad (9.11)$$

Next, the annual energy production (AEP) is estimated by integrating the power generation function over the estimated annual wind distribution. Assuming the farm operates 365×24 hours), the annual energy production can be approximate by

$$E_{farm} = 365 \times \sum_{i=1}^{n_p} P_{farm}(U^i, \theta^i) p(U^i, \theta^i) \times \left(\frac{U_{max} \times 360}{n_p} \right) \quad (9.12)$$

where n_p is the number of segmentation (in numerical integration); U^i and θ^i represent the speed and the direction of the i^{th} segment in incoming wind condition. In this section, the probability of wind speed and direction $p(U^i, \theta^i)$, is estimated by Multivariate and Multimodel Wind Distribution (MMWD) model. This model uses a KDE joint distribution to represent multimodal distributed wind data. Using AEP estimated in Eq.9.12, the average annual energy production can be expressed as

$$\bar{E}_{farm} = \frac{E_{farm}}{365 \times 24}. \quad (9.13)$$

Figure 9.3 shows the input and output structure of the average annual energy production model.



Figure 9.3: Input and output structure of the average annual energy production model

Below are other assumptions applied to the numerical experiments:

1. The GE-1.5MW-XLE turbine is chosen as the specified turbine-type in this problem.

The features of this turbine are listed in Table 9.2.

Table 9.2: Features of the *GE-1.5MW-XLE* turbine [135]

Turbine feature	Value
Rated power (P_{r0})	1.5MW
Rated wind speed (U_{r0})	11.5m/s
Cut-in wind speed (U_{in0})	3.5m/s
Cut-out wind speed (U_{out0})	20.0m/s
Rotor-diameter (D)	82.5m
Hub-height (H)	80.0m

2. The minimum *streamwise* (s_{min}) and *spanwise* (r_{min}) are set to the same value: $4D$; and
3. The wind data this problem is obtained from the North Dakota Agricultural Weather Network (NDAWN). The local wind distribution is shown in Fig. 9.5, and the onshore

wind farm scenario is assumed, and the ambient turbulence (10%) is constant over the entire farm site.

The wind data used in this problem is obtained from the North Dakota Agricultural Weather Network (NDAWN) [118]. We use the daily averaged data for wind speed and direction, measured at the Baker station between the years 2000 and 2009. Fig. 9.4 shows the Baker station, and further details are provided in Table 9.3.

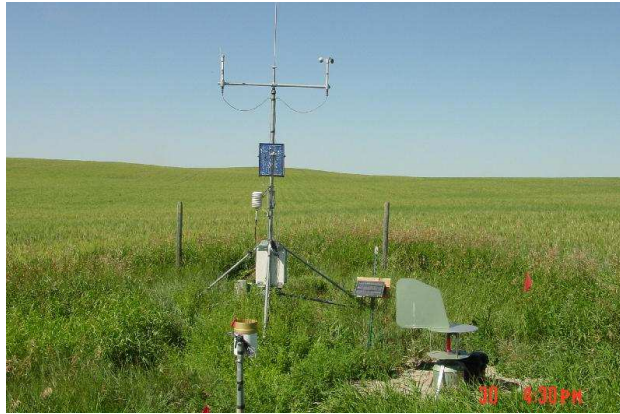


Figure 9.4: Baker station setup [118]

Table 9.3: Details of the NDAWN station at Baker, ND [118]

Parameter	Value
Location	Baker, ND
Period of Record	01/01/2000 to 12/31/2009
Latitude	48.167
Longitude	-99.648
Elevation	512m
Measurement height	3m

The wind speed data is recorded at a height of $3m$. To determine the wind speed at the pertinent heights, the log-profile is used. For a recorded wind speed U_m at a height z_m , the log profile can be expressed as

$$\frac{U}{U_m} = \frac{\ln \frac{z}{z_0}}{\ln \frac{z_m}{z_0}} \quad (9.14)$$

where z_0 is the average roughness length in the farm region, and U is the wind speed at a height z . The estimated annual distribution of wind speed and direction is illustrated by a Windrose diagram in Fig. 9.5. In the Windrose diagram, each of the sixteen sectors represent the respective probability of wind blowing from that direction.

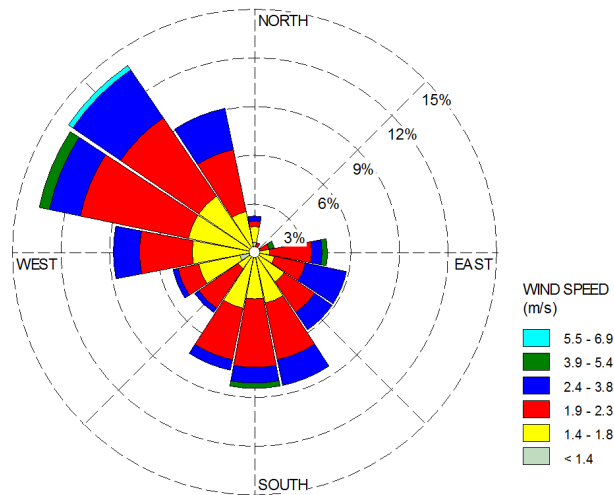


Figure 9.5: Wind rose diagram for the site at Baker

9.5 Surrogate Model Selection using COSMOS

In this section the *Concurrent Surrogate Model Selection (COSMOS)* framework (developed in Chapter 5) is applied to select the best surrogate model to represent the *average annual energy production of a large-scale wind farm* as a function of the mapping factors illustrated in Table 9.1 and the farm site orientation (ϕ). The input and output structure of the surrogate models is illustrated in Fig.9.6. In this problem, the number of sample points for training the surrogate models considered in the COSMOS framework is $N(\{X\}) = 200$. To implement the PEMF in the COSMOS, the number of training points at each iteration is determined by

$$n^r = (r - (N^{itr} + 1)) \times [\ell \times N_s] \quad (9.15)$$

where ℓ and N^{itr} are the normalized incremental value and the number of iteration. These parameters are pre-specified to $\ell = 0.1$ and $N^{itr} = 4$.

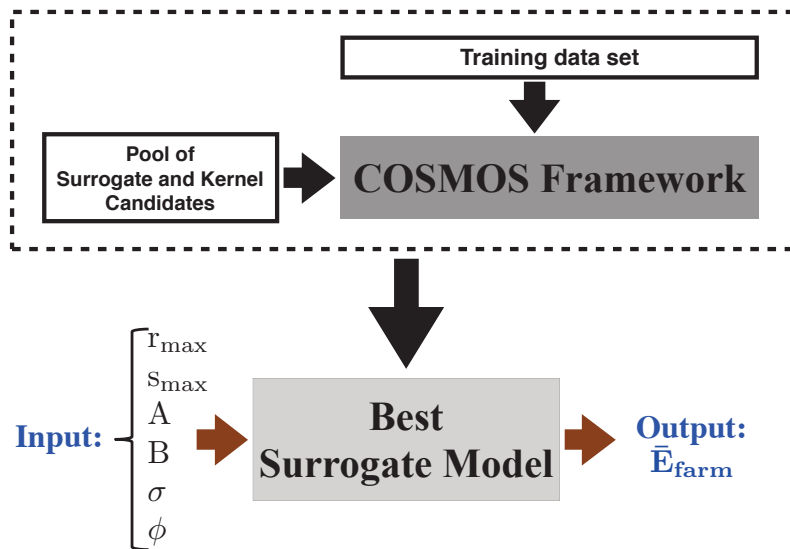


Figure 9.6: Model selection for the average annual energy production of a large-scale wind farm

The different forms of the kernel/basis/correlation functions currently considered in COSMOS are given in Chapter 5. COSMOS offers five different criteria for selection of optimal surrogates. Any combination of these criteria can be chosen by the user. In the current implementation, three of the most practically pertinent pairwise combinations are selected, as listed below:

TEST I: (i) minimize the modal value of the median error E_{med}^{mo} , and (ii) minimize the modal value of the maximum error E_{max}^{mo} ;

TEST II: (i) minimize the modal value of the median error E_{med}^{mo} , and (ii) minimize the standard deviation of the median error $E_{med}^{\sigma^2}$;

TEST III: (i) minimize the modal value of the median error E_{med}^{mo} , and (ii) minimize the expected modal value of the median error at 20% more number of sample points

$$E_{med,\alpha}^{mo}, \alpha = 0.2;$$

9.5.1 Results and Discussion

The surrogate models in the Pareto Optimal set for the average annual energy production of a large-scale wind farm under Test I, Test II, and Test III are listed in Table 9.4.

Table 9.4: The set of Pareto Surrogate Models given by COSMOS for the wind farm problem

TEST	Pareto Surrogate Model
Test I: $\min[E_{med}^{mo}, E_{max}^{mo}]$	Kriging-Linear, SVR-Sigmoid
Test II: $\min[E_{med}^{mo}, E_{med}^{\sigma}]$	Kriging-Linear, Kriging-Exponential, SVR-Sigmoid
Test III: $\min[E_{med}^{mo}, E_{med,\alpha}^{mo}]$	SVR-Sigmoid

The final optimal solutions, including the best trade-offs between the median and the maximum errors (in the Φ_0 , Φ_1 , and Φ_2 classes) for this problem is illustrated in Fig. 9.7. Each sub-figure shows the results from a particular test for the wind farm problem. In each of these figures, the 'O', the 'X', and the '+' symbols respectively represent the Pareto solutions of model selection under Φ_0 , Φ_1 , and Φ_2 classes (i.e., different hyper-parameter classes). The final Pareto optimal solutions in these figures are encircled by square symbols, and are derived by applying a Pareto Filter.

In this problem, the selected surrogate model will be used as an initial model in the surrogate-based optimization process. In the surrogate-based optimization process, the fidelity of surrogate model iteratively should be improved by adding infill points in the optimization process. Therefore, Kriging model with Linear correlation function could be the

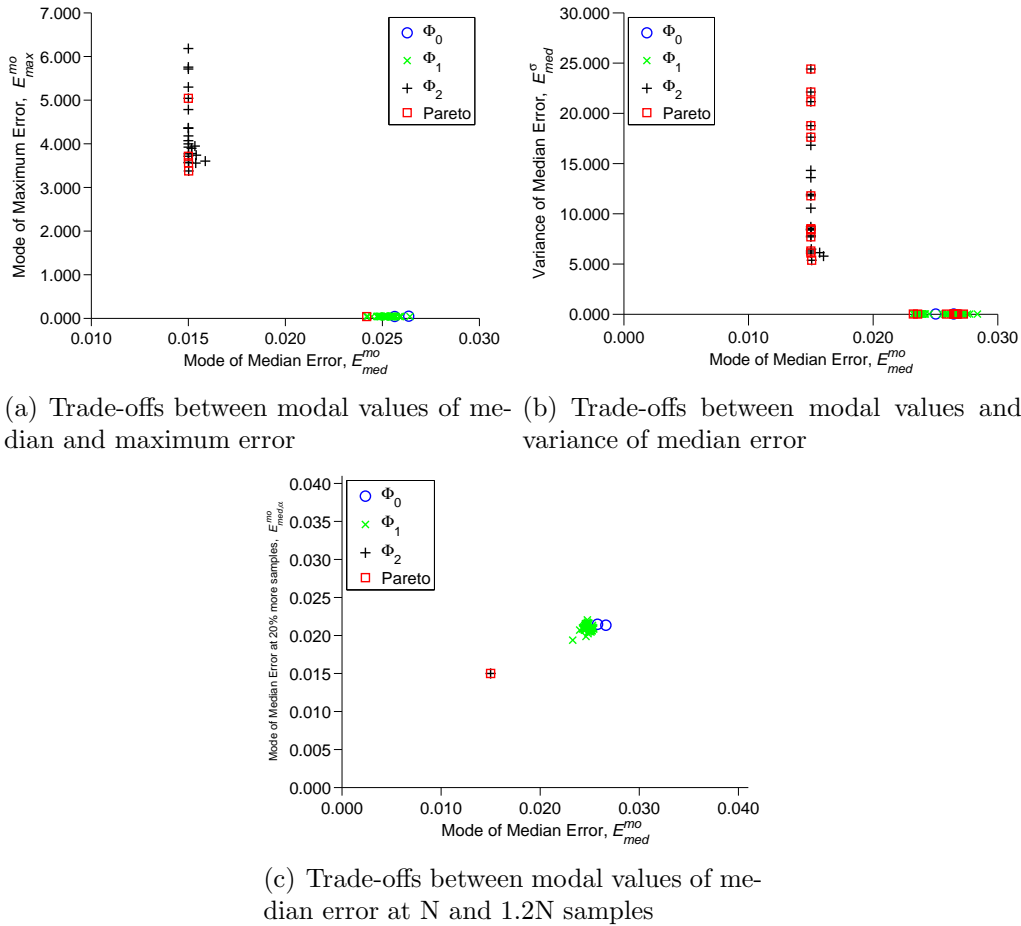


Figure 9.7: Pareto models and final population of models from all Φ classes

best choice. The SVR model with the Sigmoid kernel function is not selected because of the the high variance error of this model.

9.6 Variable Fidelity Optimization using AMR

In this section, the Adaptive Model Refinement (AMR) approach (developed in Chapter 7) is applied to a wind farm layout design problem. The objective is to maximize the average annual energy production of the farm through layout optimization. In this problem, the farm layout optimization is started using the best surrogate model selected in Sec. 9.5 (Kriging model with Linear correlation function). To reach a reliable optimum solution, a

refinement of the surrogate is then performed by sequentially adding batches of new samples using the Adaptive Model refinement approach.

$$\begin{aligned} \max_V : \quad & f = \bar{E}_{farm} \\ & V = \{v_1, v_2, \dots, v_6\} \\ \text{subject to} \\ & g_1(v) \leq 0 \\ & g_2(v) \leq 0 \\ & V^{Lower} \leq v_i \leq V^{Upper} \end{aligned}$$

The upper and lower bounds of design variables ($v_i = 1, 2, \dots, 6$) are listed in Table 9.5. These variables include the mapping factors (that defines the coordinates of the wind turbines in the farm), and the farm site orientation (ϕ). In Eq. 9.16, $g_1(v) = \|A_{MW} - 45(\text{hectare}/MW)\|$ is the land area constraints. This constraint is defined based on the average land usage of US commercial wind farms in 2009. In this equation, $g_2(v) = \sum_{i,j=1, i \neq j}^{N=500} \max\{[4D - d_{ij}]\}$ is the minimum inter-turbine spacing constraints.

Table 9.5: Upper and lower bounds of design variables

Design variables	Lower bound	Upper bound	Baseline
r_{max}	5D	15D	10D
s_{max}	5D	15D	10D
A	-20	20	0.1
B	-20	20	0.1
σ	0	1	0
ϕ	0	90	0

Table 9.5 also contains a baseline set of values, roughly representative of a layout.

These baseline values and the upper and lower bounds were used to generate a filled contour plot of the average annual energy production (see Figure 9.8) by varying the inputs pairwise and keeping the remaining variables at the baseline value.

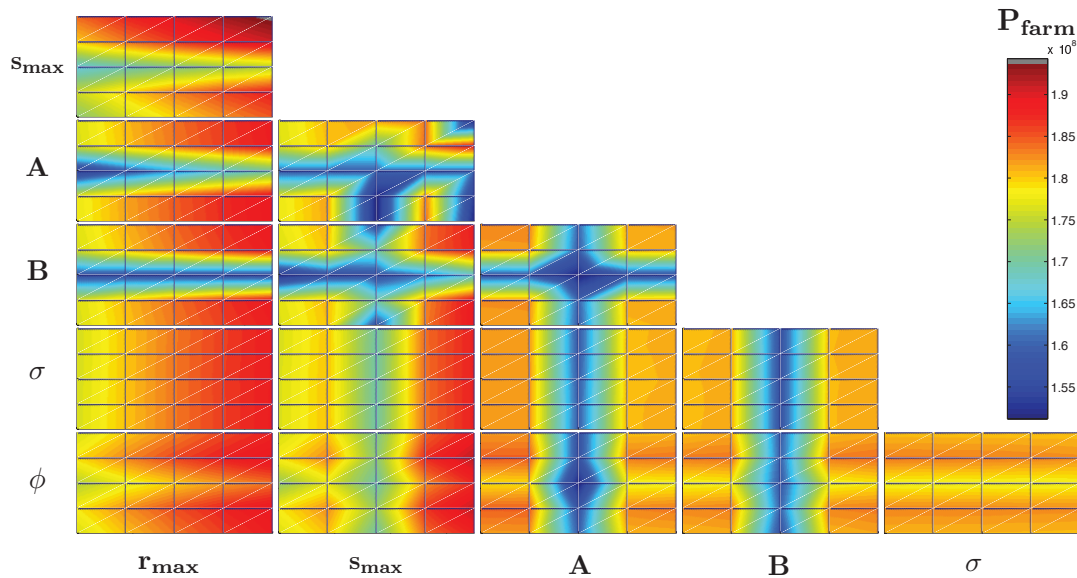


Figure 9.8: Wind farm problem: Each tile shows a contour of the wind farm average annual energy production versus two of the six mapping factors (variables), with the remaining four variables held at the baseline value

As expected, in this problem, it is observed from Fig. 9.8 that all the design variables are active and they are involved in interactions with other variables. It is also observed that s_{\max} and r_{\max} are clearly very active in power energy of the wind farm.

9.6.1 Results and Discussion

In this problem, the AMR metric is applied at every iteration $\delta = 1$, the Indicator of Conservativeness (IoC) is pre-specified to $p_{cr} = 0.3$. The IoC regulates the trade-off between optimal solution reliability and computational cost in SBO. Generally, the higher the IoC (closer to 1), the higher the solution reliability and the greater the computational cost; under these conditions, model switching events will occur early on in the optimization process. The

population size of PSO is pre-specified to $N_{pop} = 300$, and the PSO algorithm converges by satisfying the predefined function tolerance, $\delta f = 1e - 6$. In this problem the model refinement will be implemented if the current size of data set is less than $N(\{X\}) = 500$. In this optimization problem, the initial population of particles is generated using the Kriging model with Linear correlation function. The AMR approach adaptively improve the fidelity of the surrogate model ten times during the optimization process (over a total of 110 iterations), resulting in an optimum design with a reasonable level of fidelity.

Figure 4.19 represents the improvement of the model fidelity for the wind farm problem through the sequential model refinement process using the AMR method. This figure illustrates that the error decreases with sequential addition of batches of infill points in SBO, thus showing that the batch size of the infill points and the location of these points determined by the AMR method resulted in lower surrogate model error in the refined (updated) surrogate model. It should also be noted that the novel use of the modal value in the PEMF method (developed in Chapter 3), promotes a decrease of the error measure with infill points, as opposed to the untraceable noisy variation that is often characteristic of mean or RMS error measures [121].

The convergence history of the optimization problem is illustrated in Fig. 9.10. These figures also indicate: (i) the active surrogate model at each iteration (using different colors), (ii) the size of data set ($N(\{X\}) = N^{Current}(\{X\}) + \Gamma^{Infill}$) used to refine (update) the active surrogate model in the AMR approach, and (iii) the iteration that the model refinement is performed (the AMR metric is met). It is observed that, from the first iteration till the 8th iteration the initial surrogate model with $N(\{X\}) = 200$ is active, before refining the model using additional 19 sample data through the AMR approach. The final switching event, from the surrogate model constructed using $N(\{X\}) = 496$ training points to the surrogate model

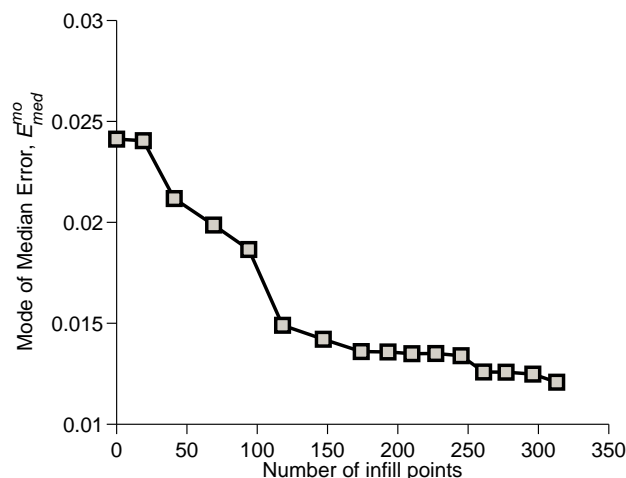


Figure 9.9: The improvement of the surrogate model fidelity through sequential model refinement process with AMR

constructed using $N(\{X\}) = 513$, occurs at the 96th iteration. The optimization progresses using the surrogate model constructed using $N(\{X\}) = 513$ training points for another 14 iterations before reaching convergence. In this case, the algorithm converges by satisfying the predefined function tolerance, $\delta f = 1e - 6$.

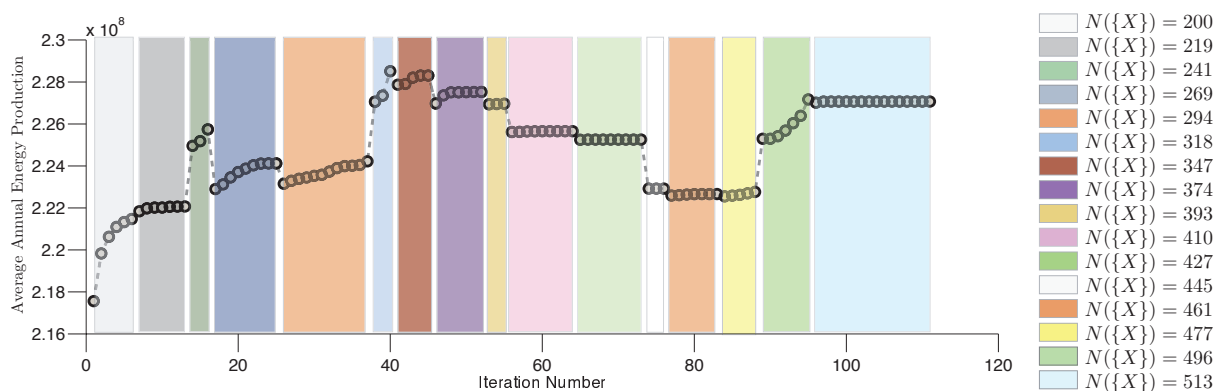


Figure 9.10: Convergence history of the surrogate-based optimization using AMR

Next, we investigate how the AMR method performs better than simply using the single stage sampling strategy for SBO. The result yielded by the AMR method are therefore compared with the results yielded by running MDPSO using the surrogate model constructed in the one-step method (using all 500 sample points). The optimum results are thus obtained,

and the total number of function evaluations in each case are reported in Table 9.6. The final column of this table shows the actual function estimate for the optimum design obtained under each optimization run. It is observed that the PSO-AMR not only requires 26 times less computing time compared to PSO-HF, it also provides the optimum value that is 14.9 MWh better than simply using the single stage sampling strategy for SBO. It is also observed that, in the PSO-AMR approach, the optimum is located in the region where the SM model has 0.63% error.

Table 9.6: Optimization results using different optimization approaches

Approach	v_1^* $\times D$	v_2^* $\times D$	v_3^*	v_4^*	v_5^*	v_6^* (degree)	Optimum function \bar{E}_{farm} ($f^* \times 10^8$)	Model in last iteration	No. of HF function evaluations	HF response at optimum ($f_{HF}(v^*) \times 10^8$)	A_{MW}
PSO-SM	14.03	13.40	15.68	4.72	0.05	70.73	2.3013	SM	500	2.1248	29.7085
PSO-HF	13.70	13.54	16.27	4.92	0.08	69.69	2.2760	HF	300×45	2.2760	43.80
PSO-AMR	12.65	12.57	9.99	-2.53	0.03	72.17	2.2892	SM	513	2.2748	44.74

PSO-SM: optimization performed by MDPSO solely using the surrogate model with all 500 sample points

PSO-HF: optimization performed by MDPSO solely using the high fidelity model

PSO-AMR: optimization performed by MDPSO using AMR approach (surrogate-based optimization using AMR)

Figure 9.11 presents an illustration of the optimal layout. It can be observed that the spacing between rows and columns is nonuniform. It is also observed that there is a relatively higher degree of freedom in location of turbines, owing to the allowed radial displacement. The number of rows and column, and the land area per MW installed (LAMI) are also illustrated in this figure.

9.7 Chapter Summary

This chapter presented a new approach for layout optimization of very large scale wind farms based on the methods developed in Chapters 5, 6, and 7 of this dissertation. A mapping of the wind farm layout to the detailed turbine coordinates was created, allowing the wind farm layout to be represented by a set of 6 parameters. As a result, the design variable space for optimizing the layout of a large scale wind farm is significantly reduced.

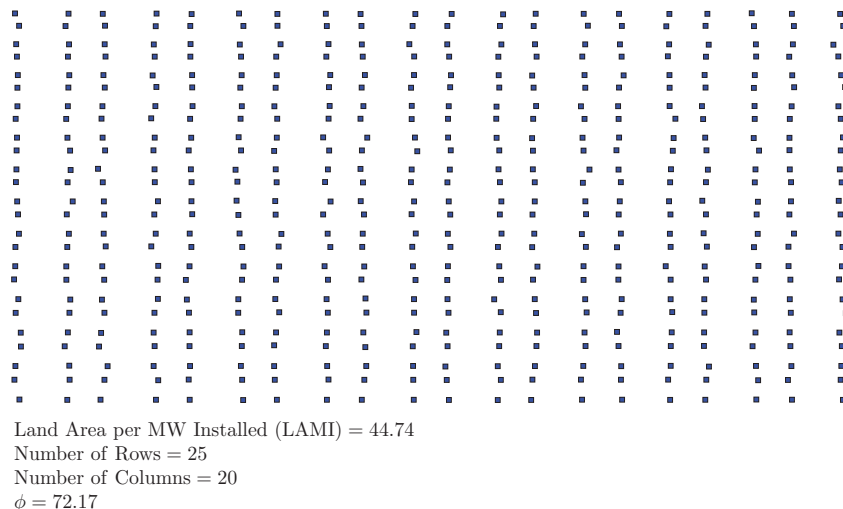


Figure 9.11: Illustration of the optimized wind farm layout

The COSMOS framework was then applied to select the globally-best surrogate model to represent the energy production of the wind farm as a function of the reduced set of layout variables. Surrogate-based optimization was then performed using the Adaptive Model Refinement approach, implemented through Particle Swarm Optimization. The objective of wind farm layout optimization was to maximize the average annual energy production of a 500-turbine wind farm. The results indicated that AMR along with Mixed Discrete PSO improved the efficiency of the optimization process by a factor of 26 when compared to optimization using the standard energy production model, while retaining an accuracy of within 0.05% of the latter.

PART V

Concluding Remarks and Future Work

CHAPTER 10

Concluding Remarks and Future Work

What are the important contributions of this dissertation? Is there potential for future improvement?

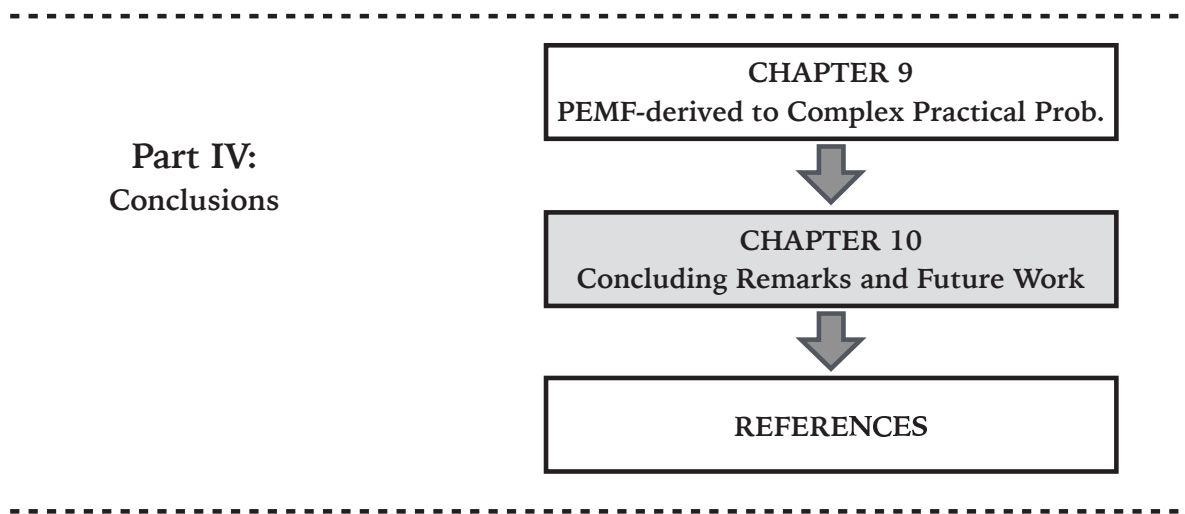


Figure 10.1: Chapter 10 orientation

In this chapter, we conclude with the observations made from the research performed to date. In addition, we also present an overview of potential improvements to the algorithms and methods developed.

10.1 Summary of Contributions in the Dissertation

Below is a summary of the important contributions of this dissertation:

1. **Reliable ‘model-independent’ methodology for quantifying the fidelity of the ‘actual surrogate’:**

In this dissertation, a model-independent methodology for quantifying the fidelity of the ‘actual surrogate’ without requiring additional system evaluations was developed. The developed error measurement, Predictive Estimation of Model Fidelity (PEMF), is derived from the hypothesis that the accuracy of approximation models is related to the amount of data resources leveraged to train the model. The PEMF method quantifies the level of surrogate fidelity in any given region of the design domain, which can then be applied for informed decision-making based on the surrogates. The smart use of the modal value in this error measure makes it significantly less susceptible to outliers and the Design of Experiments. This new paradigm in surrogate modeling was shown to be highly reliable and robust in comparison to cross-validation (which is still the current paradigm).

2. Tracking the variation of the surrogate model error with an increasing density of training points:

We proposed a novel method which provides maximum pertinent information regarding the behavior of the model in comparison to the current paradigm. The PEMF method does this by tracking the trend exhibited by the error with an increasing number of sample points (or tracking the progress in model fidelity as additional knowledge becomes available) irrespective of the model type. We also proposed a criterion, called Monotonic Trend Criterion (MTC), to statistically test the feasibility of the monotonic decrease of the model fidelity with increasing sample density in any given region.

3. A new model selection framework to identify the globally-competitive surrogate model:

In this dissertation, a new model selection framework for identifying the best surrogate model (or set of models) from a diverse pool of candidates was developed. This framework, called Concurrent Surrogate Model Selection (COSMOS), selects the globally-optimal surrogates by simultaneously operating at the following three levels: (i) selecting the optimal model type, (ii) selecting the optimal kernel/basis function type, and (iii) finding the optimum values of the hyper-parameters in the kernel.

4. Flexibility in using different criteria in surrogate model selection:

The model selection framework proposed in this dissertation, COSMOS, gives users the unique flexibility to use different criteria for model selection. These criteria include:

- (i) the (predicted) most probable values of the median error and the maximum error in the surrogate which stochastically account for the sensitivity of the surrogate accuracy to the choice (number and location) of training points;
- (ii) the variance in the median and the maximum error of the surrogate model (which reflect the robustness of the surrogate model); and
- (iii) the (expected) most probable value of the median error on a hypothetically-larger sample set.

5. Variable fidelity optimization using Adaptive Model Switching

In this dissertation, we developed a novel model management technique to be implemented in population-based optimization algorithms to provide high fidelity optimal

designs at reasonable computational expense. This model management technique, called the Adaptive Model Switching Metric (AMS), is perceived as a decision-making tool for the timing of model-switching or model integration during the optimization process using models with different levels of fidelity. This criterion is defined based on “whether the uncertainty associated with a model response is higher than the observed improvement in the relative fitness function of the population”. The AMS criterion along with Mixed Discrete PSO improved the efficiency of the optimization process significantly.

6. Surrogate-based optimization using Adaptive Model Refinement

In this dissertation, we developed a new model-independent surrogate-based optimization (SBO) approach to be used in population-based optimization algorithms. The proposed method, called Adaptive Model Refinement (AMR), is designed to refine the surrogate model by sequentially adding a batch of new samples at any given iteration of SBO when a switching metric is met. In this approach, the Adaptive Model Switching (AMS) metric and Predictive Estimation of Model Fidelity (with some important advancements) are used to determine the timing of model refinement and the batch size for the samples to be added, respectively.

10.2 Limitations and Future Work

Throughout the dissertation, we have mentioned a number of issues that should be investigated in the future. In this section, we further elaborate on potential future research.

10.2.1 Predictive Estimation of Model Fidelity (PEMF)

In Chapter 3 of this dissertation, we developed the PEMF method for quantifying the fidelity of the surrogate model. It can be said about any ongoing research that there will always remain certain aspects that need further investigation and careful handling. The same is true about the developmental work presented in this document. Below, we identify some aspects of the PEMF approach that need further development.

- (i) In the PEMF method, there are two factors that need to be chosen carefully: (i) the number of iterations and (ii) the initial ratio of the numbers of intermediate training points to test points. Chapter 3 provided some preliminary guidelines regarding the choice of appropriate values for these factors. Future research efforts could consider dynamically setting these factors during the surrogate model construction, and adaptively determine these two factors as functions of the dimension and the strength of the sample data.
- (ii) In this dissertation, we introduced a novel concept for quantifying the level of surrogate errors in any given region of the design domain. There remains an opportunity to develop strategies for choosing the appropriate sample size (or the boundaries of the region) in which the Predictive Estimation of Model Fidelity (PEMF) is feasible.

10.2.2 Concurrent Surrogate Model Selection (COSMOS)

In Chapter 5 of this dissertation, we developed the COSMOS framework to search for a globally-competitive surrogate model. However, some aspects of the COSMOS framework still need further development as discussed below.

- (i) In this dissertation, the COSMOS framework was used to select the best surrogate

model for the entire design domain. Future research efforts could consider determining the best-suited surrogate models for different regions of the design domain.

- (ii) The COSMOS framework was implemented to select the best models among surrogate models, which were constructed using uniform sample points. Future research should pursue testing of COSMOS on nonuniform sample sets (e.g., samples following specified probability distributions).
- (iii) Future work may also include implementing the COSMOS framework for *network weights* estimation in Artificial Neural Network (ANN). Hyper-parameter optimization in the COSMOS framework could avoid the over-fitting issue (due to the sum-squared error) in ANN. Apart from providing the best value for the network weights, the predictive ability of ANN models can be readily compared with that of other models (regression and interpolation models e.g., Kriging) using COSMOS, where ANN will serve as a member of the pool of candidate models available for selection.

10.2.3 Variable Fidelity Optimization

In Chapters 6 and 7 of this dissertation, we developed two metrics for variable fidelity optimization and surrogate-based optimization. These two metrics facilitate the applicability of population-based heuristic optimization algorithms in practical complex system design. Below we present some aspects of these metrics that need further development.

- (i) In this dissertation, the proposed model management and surrogate-based design optimization approaches are design for single-objective population-based heuristic optimization algorithms. However, complex design problems are almost always characterized by the presence of multiple objectives. In the future, modified versions of these

metrics for multi-objective population-based heuristic optimization algorithms should be developed.

- (ii) In the developed model switching metric in this dissertation, the critical probability (or Indicator of Conservativeness) that regulates the trade-off between optimal solution reliability and the total computational cost of the optimization process, is predefined by the user (based on prior experience or practical design requirements). A more intuitive definition of the Indicator of Conservativeness as a function of a system complexity, computational resources, and a desired robustness would further establish the wide potential of these metrics for optimizing complex practical systems.
- (iii) Future research efforts should seek to consider the possibility of *switching back to the lower fidelity model* in the variable fidelity optimization using the developed Adaptive Model Switching metric, for the sake of computational efficiency. In this scenario, a backward switching event (switching to a lower fidelity model) will be allowed if the averaged quantile value of the distribution of the fitness function improvement in κ subsequent iterations ($\frac{\sum_{t=1}^{\kappa} Q_{\theta}^t}{\kappa}$) is more than that of the prior model uncertainty ($Q_{\mathbb{P}_{i-1}}$) in the variable fidelity optimization process. Such scenarios are not uncommon in heuristic and stochastic optimization processes, where the slope of the objective function convergence can both increase and decrease dramatically at an iteration. However, allowing backward switching can change the ranking among the candidate solutions in the population, thereby running the risk of leading the population astray in search of optimum. Careful investigation is thus needed to consider the trade-offs between the risk and gain (in computational efficiency) that backward switching is associated with.
- (iv) Physics-based models are fundamentally constructed based on physical understand-

ing of the real systems (unlike surrogate models). In these models, input parameters (e.g., dimension and material property) have their own physical meaning and are often determined experimentally. Therefore, future research in the AMS metric should also pursue implementing a method (e.g., Bayesian Model Averaging) to quantify model uncertainty associated with the physics-based model while considering a prior uncertainty in input parameters.

10.2.4 Conservative surrogate model

In future research, the global and regional maximum error estimated using PEMF can be used to construct a more reliable conservative surrogate model. This model can be used to conservatively approximate the response of the complex system in reliability-based optimization, and assure the obtained optimum/optima satisfy the design constraints and criteria.

10.2.5 Hybrid surrogate model

In future research, the regional error estimated using PEMF can be used to appropriately combine the favorable attributes of characteristically different surrogate modeling methods, and construct a more accurate hybrid surrogate model.

10.2.6 Large-scale applications

- (i) Future research should also consider the implementation of the PEMF-based approaches (which have been developed in this dissertation) to solve complex large-scale problems in different engineering disciplines, such as computational fluid dynamics and vehicle crash simulations.

(ii) In large-scale complex system optimization problems (e.g., aerodynamics shape optimization of helicopter rotor blades), the selection of the highest fidelity model from two or more available Physics-based simulation models (e.g., LES, DNS, and RANS) is generally challenging. To solve these challenging problems, the uncertainty in the output of different models can be defined as the variation in the output given by different alternatives for the same model, where the alternatives are in the same class of computational expense and fidelity. Future research efforts should consider this uncertainty in informed modeling (such as the Adaptive Model Switching metric, the COSMOS framework, and the Adaptive Model Refinement approach). This future work is necessary to enhance the applicability of the developed methods and frameworks in this dissertation.

We do not claim the developed methods to be entirely perfect; indeed, we have identified potential areas of further improvement and some possible limitations. However, even in their present form, the methods developed in this dissertation can lead to better decisions and ultimately better systems through the power of surrogate modeling and optimization.

PART VI

REFERENCES

REFERENCES

- [1] Box, G. E. P. and Draper, N. R., *Empirical model-building and response surfaces*, John Wiley and Sons, 1987.
- [2] Messac, A., *Optimization in Practice with MATLAB: For Engineering Students and Professionals*, Cambridge University Press, 2015.
- [3] Myers, R. and Montgomery, D., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley-Interscience; 2nd edition, 2002.
- [4] Ye, K. Q., Li, W., and Sudjianto, A., “Algorithmic Construction of Optimal Symmetric Latin Hypercube Designs,” *Journal of Statistical Planning and Inference*, Vol. 90, 2000, pp. 145–159.
- [5] Owen, A. B., “Orthogonal Arrays for Computer Experiments, Integration and Visualization,” *Statistica Sinica*, Vol. 2, 1992, pp. 439–452.
- [6] Zhang, J., Chowdhury, S., Messac, A., Zhang, J., and Castillo, L., “Surrogate Modeling of Complex Systems Using Adaptive Hybrid Functions,” *ASME 2011 International Design Engineering Technical Conferences (IDETC)*, Washington, DC, August 2011.
- [7] Jin, R., Chen, W., and Simpson, T. W., “Comparative Studies of Metamodeling Techniques Under Multiple Modeling Criteria,” *AIAA*, , No. 4801, 2000.
- [8] Simpson, T., Korte, J., Mauery, T., and Mistree, F., “Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization,” *AIAA Journal*, Vol. 39, No. 12, 2001, pp. 2233–2241.

- [9] Forrester, A. and Keane, A., “Recent Advances in Surrogate-based Optimization,” *Progress in Aerospace Sciences*, Vol. 45, No. 1-3, 2009, pp. 50–79.
- [10] Hardy, R. L., “Multiquadric Equations of Topography and Other Irregular Surfaces,” *Journal of Geophysical Research*, Vol. 76, 1971, pp. 1905–1915.
- [11] Yegnanarayana, B., *Artificial Neural Networks*, PHI Learning Pvt. Ltd., 2004.
- [12] Gunn, S. R., “Support vector machines for classification and regression,” Tech. rep., ISIS - 14, NASA Langley Research Center, Hampton, VA, 1998.
- [13] Queipo, N., Haftka, R., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P., “Surrogate-based Analysis and Optimization,” *Progress in Aerospace Sciences*, Vol. 41, No. 1, 2005, pp. 1–28.
- [14] Zhang, J., Chowdhury, S., and Messac, A., “An Adaptive Hybrid Surrogate Model,” *Structural and Multidisciplinary Optimization*, Vol. 46, No. 2, 2012, pp. 223–238.
- [15] Lehmensiek, R., Meyer, P., and Muller, M., “Adaptive sampling applied to multivariate, multiple output rational interpolation models with application to microwave circuits,” *International Journal of RF and Microwave Computer-Aided Engineering*, Vol. 12, No. 4, 2002, pp. 332–340.
- [16] Sugiyama, M., “Active Learning in Approximately Linear Regression Based on Conditional Expectation of Generalization Error,” *Journal of Machine Learning Research*, Vol. 7, 2006, pp. 141–166.
- [17] Jones, D., Schonlau, M., and Welch, W., “Efficient Global Optimization of Expensive

- Black-box Functions,” *Journal of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492.
- [18] Mehmani, A., Zhang, J., Chowdhury, S., and Messac, A., “Surrogate-based Design Optimization with Adaptive Sequential Sampling,” *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Hawaii, USA, April 2012.
- [19] Simpson, T., Booker, A., Ghosh, D., Giunta, A., Koch, P., and Yang, R.-J., “Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion,” *Structural and Multidisciplinary Optimization*, Vol. 27, No. 5, 2004, pp. 302–313.
- [20] Wang, G. and Shan, S., “Review of Metamodeling Techniques in Support of Engineering Design Optimization,” *Journal of Mechanical Design*, Vol. 129, No. 4, 2007, pp. 370–380.
- [21] Simpson, T. W., Toropov, V., Balabanov, V., and Viana, F. A. C., “Design and Analysis of Computer Experiments in Multidisciplinary Design Optimization: A Review of How Far We Have Come or Not,” *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA, Victoria, Canada, September 10-12 2008.
- [22] Myers, R. H. and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley, New York, 2nd ed., 2002.
- [23] Giunta, A. and Watson, L., “A Comparison of Approximation Modeling Techniques: Polynomial Versus Interpolating Models,” Tech. Rep. AIAA-98-4758, 1998.
- [24] Cressie, N., *Statistics for Spatial Data*, Wiley, New York, 1993.

- [25] Jin, R., Chen, W., and Simpson, T., “Comparative Studies of Metamodelling Techniques under Multiple Modelling Criteria,” *Structural and Multidisciplinary Optimization*, Vol. 23, No. 1, 2001, pp. 1–13.
- [26] Cherrie, J. B., Beatson, R. K., and Newsam, G. N., “Fast Evaluation of Radial Basis Functions: Methods for Generalized Multiquadrics in R^n ,” *SIAM Journal of Scientific Computing*, Vol. 23, No. 5, 2002, pp. 1549–1571.
- [27] Hussain, M. F., Barton, R. R., and Joshi, S. B., “Metamodeling: Radial Basis Functions, Versus Polynomials,” *European Journal of Operational Research*, Vol. 138, No. 1, 2002, pp. 142–154.
- [28] Mullur, A. and Messac, A., “Extended Radial Basis Functions: More Flexible and Effective Metamodeling,” *AIAA Journal*, Vol. 43, No. 6, 2005, pp. 1306–1315.
- [29] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*, Wiley, New York, 2nd ed., 2000.
- [30] Clarke, S. M., Griebisch, J. H., and Simpson, T. W., “Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses,” *Journal of Mechanical Design*, Vol. 127, No. 6, 2005, pp. 1077–1087.
- [31] Vapnik, V., *The Nature of Statistical Learning Theory*, Springer, New York, 1995.
- [32] Basudhar, A. and Missoum, S., “Adaptive Explicit Decision Functions for Probabilistic Design and Optimization Using Support Vector Machines,” *Computers and Structures*, Vol. 86, No. 19-20, 2008, pp. 1904–1917.
- [33] Simpson, T. W., Peplinski, J. D., Koch, P. N., and Allen, J. K., “Metamodels for

- Computer-based Engineering Design: Survey and Recommendations,” *Engineering with Computers*, Vol. 17, No. 2, 2001, pp. 129–150.
- [34] Mongillo, M., “Choosing basis functions and shape parameters for radial basis function methods,” *SIAM Undergraduate Research Online*, 2011.
- [35] Martin, J. D. and Simpson, T. W., “Use of kriging models to approximate deterministic computer models,” *AIAA journal*, Vol. 43, No. 4, 2005, pp. 853–863.
- [36] Mullur, A. A., *A New Response Surface Paradigm for Computationally Efficient Multi-objective Optimization*, Ph.D. thesis, Mechanical Engineering, Rensselaer Polytechnic Institute, Troy, NY, December 2005.
- [37] Bunge, M., *Foundations of Physics*, Springer-Verlag, New York, 1967.
- [38] Park, I. S., *Quantification of Multiples Types of Uncertainty in Physics-based Simulation*, Ph.D. thesis, Wright State University, 2012.
- [39] Yang, Y., “Regression with Multiple Candidate Models: Selecting or Mixing?” *Statistica Sinica*, Vol. 13, 2003, pp. 783–809.
- [40] Lee, S. H. and Chen, W., “A Comparative Study of Uncertainty Propagation Methods for Black-box-type Problems,” *Structural and Multidisciplinary Optimization*, Vol. 37, 2009, pp. 239–253.
- [41] Metropolis, N. and Ulam, S., “The Monte Carlo Method,” *Journal of the American Statistical Association*, Vol. 44, 1949, pp. 335–341.
- [42] Haldar, A. and Mahadevan, S., *First-order and second-order reliability methods*, Springer-Verlag, New York, 1995.

- [43] Meckesheimer, M., Barton, R., Simpson, T., and Booker, A., “Computationally inexpensive Metamodel Assessment Strategies,” *AIAA*, Vol. 40, No. 10, 2002, pp. 2053–2060.
- [44] Goel, T., Haftka, R., Shyy, W., and Queipo, N., “Ensemble of Surrogates,” *Structural and Multidisciplinary Optimization*, Vol. 33, No. 3, 2007, pp. 199–216.
- [45] Bischl, B., Mersmann, O., and Trautmann, H., “Resampling methods in model validation,” *Proc. WEMACS*, Vol. 14, 2010.
- [46] Wu, C. F. J., “Jackknife, bootstrap and other resampling methods in regression analysis,” *the Annals of Statistics*, 1986, pp. 1261–1295.
- [47] Romero, D. A., Marin, V. E., and Amon, C. H., “Error Metrics and the Sequential Refinement of Kriging Metamodels,” *Journal of Mechanical Design*, Vol. 1, No. 137, 2015, pp. 011402.
- [48] Larson, S. C., “The shrinkage of the coefficient of multiple correlation,” *Journal of Educational Psychology*, Vol. 45, No. 1, 1931, pp. 031102.
- [49] Stone, M., “Cross-validatory choice and assessment of statistical predictions,” *Journal of the Royal Statistical Society*, Vol. 36, No. 2, 1974, pp. 111–147.
- [50] Salkind, N. J., *Encyclopedia of Research Design*, SAGE Publications, 2010.
- [51] Viana, F., Haftka, R., and Steffen, V., “Multiple Surrogates: How Cross-validation Errors Can Help Us to Obtain the Best Predictor,” *Structural and Multidisciplinary Optimization*, Vol. 39, No. 4, 2009, pp. 439–457.

- [52] Goel, T., Haftka, R. T., and Shyy, W., “Comparing error estimation measures for polynomial and Kriging approximation of noise-free functions,” *Structural and Multidisciplinary Optimization*, Vol. 38, No. 5, 2009, pp. 429–442.
- [53] Goel, T. and Stander, N., “Comparing three error criteria for selecting radial basis function network topology,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 27, 2009, pp. 2137–2150.
- [54] Acar, E. and Rais-Rohani, M., “Ensemble of metamodels with optimized weight factors,” *Structural and Multidisciplinary Optimization*, Vol. 37, No. 3, 2009, pp. 279–294.
- [55] Aute, V., Saleh, K., Abdelaziz, O., Azarm, S., and Radermacher, R., “Cross-validation based single response adaptive design of experiments for Kriging metamodeling of deterministic computer simulations,” *Structural and Multidisciplinary Optimization*, Vol. 48, No. 3, 2013, pp. 581–605.
- [56] Viana, F. A. C., Pecheny, V., and Haftka, R. T., “Using cross validation to design conservative surrogates,” *AIAA JOURNAL*, Vol. 48, No. 10, 2010, pp. 2286–2298.
- [57] Zhang, J., Chowdhury, S., Mehmani, A., and Messac, A., “Characterizing Uncertainty Attributable to Surrogate Models,” *Journal of Mechanical Design*, Vol. 3, No. 136, 2014, pp. 031004.
- [58] Nguyen, H. M., Couckuyt, I., Knockaert, L., Dhaene, T., Gorissen, D., and Saeys, Y., “An Alternative Approach to Avoid Overfitting for Surrogate Models,” *Proceedings of the 2011 Winter Simulation Conference*, 2011, pp. 2765–2776.
- [59] Deschrijver, D. and Dhaene, T., “An Alternative Approach to Avoid Overfitting for

- Surrogate Models,” *Signal Propagation on Interconnects, 2005. Proceedings. 9th IEEE Workshop*, 2005, pp. 111–114.
- [60] Gorissen, D., Dhaene, T., and Turck, F. D., “Evolutionary Model Type Selection for Global Surrogate Modeling,” *Journal of Machine Learning Research*, Vol. 10, 2009, pp. 2039–2078.
- [61] Solomatine, D. and Ostfeld, A., “Data-driven modelling: some past experiences and new approaches,” *Journal of hydroinformatics*, Vol. 10, No. 1, 2008, pp. 3–22.
- [62] Holena, M. and Demut, R., “Assessing the suitability of surrogate models in evolutionary optimization.” *In Information Technologies*, 2011, pp. 31–38.
- [63] Gano, S. E., Kim, H., and Brown, D. E., “Comparison of three surrogate modeling techniques: Datascape, kriging, and second order regression,” *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA-2006-7048, Portsmouth, Virginia*, 2006.
- [64] Chen, P. W., Wang, J. Y., and Lee, H. M., “Model selection of SVMs using GA approach,” *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, Vol. 3, IEEE, 2004, pp. 2035–2040.
- [65] Viana, F. A. C., Haftka, R. T., and Jr., V. S., “Multiple surrogates: how cross-validation errors can help us to obtain the best predictor,” *Structural and Multidisciplinary Optimization*, Vol. 39, 2009, pp. 439–457.
- [66] Bozdogan, H., “Akaike’s information criterion and recent developments in information complexity,” *Journal of Mathematical Psychology*, Vol. 44, 2000, pp. 62–91.

- [67] Claeskens, G. and Hjort, N. L., *Model selection and model averaging*, Cambridge Books.
- [68] Acar, E., “Optimizing the shape parameters of radial basis functions: An application to automobile crashworthiness,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Vol. 224, No. 12, 2010, pp. 1541–1553.
- [69] Rippa, S., “An algorithm for selecting a good value for the parameter c in radial basis function interpolation,” *Advances in Computational Mathematics*, Vol. 11, No. 2-3, 1999, pp. 193–210.
- [70] Mongillo, M., “Choosing basis functions and shape parameters for radial basis function methods,” *SIAM Undergraduate Research Online*, accepted 2011.
- [71] Choi, K., Young, B., and Yang, R., “Moving Least Square Method for Reliability-based Design Optimization,” 4th World Congress of Structural and Multidisciplinary Optimization, Dalian, China, June 2001, pp. 4–8.
- [72] Toropov, V. V., Schramm, U., Sahai, A., Jones, R. D., and Zeguer, T., “Design Optimization and Stochastic Analysis based on the Moving Least Squares Method,” 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, 30 May - 03 June 2005.
- [73] Clarke, S. M., Griebisch, J. H., and Simpson, T. W., “Analysis of support vector regression for approximation of complex engineering analyses,” *Journal of mechanical design*, 2005.
- [74] Barthelemy, J.-F. and Haftka, R., “Approximation Concepts for Optimum Structural Design. In A Review,” *Structural Optimization*, Vol. 5, No. 3, 1993, pp. 129–144.

- [75] Haftka, R. T., “Combining global and local approximations,” *AIAA journal*, Vol. 29, No. 9, 1991, pp. 1523–1525.
- [76] Zadeh, P. M., Toropov, V. V., and Wood, A. S., “Metamodel-based collaborative optimization framework,” *Structural and Multidisciplinary Optimization*, Vol. 38, No. 2, 2009, pp. 103–115.
- [77] Zadeh, P. M. and Mehmani, A., “Multidisciplinary Design Optimization Using Variable Fidelity Modeling: Application to a Wing Based on High Fidelity Models,” Third International Conference on Multidisciplinary Design Optimziation, Paris, France, June 2010.
- [78] Keane, A. J. and Nair, P. B., *Computational Approaches for Aerospace Design: The Pursuit of Excellence*, Wiley, New York, 2005.
- [79] Booker, A. J., Dennis, J. E., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., “A rigorous framework for optimization of expensive functions by surrogates,” *Structural optimization*, Vol. 17, No. 1, 1999, pp. 1–13.
- [80] Alexandrov, N., Lewis, R., Gumbert, C., Green, L., and Newman, P., “Optimization With Variable-Fidelity Models Applied To Wing Design,” Tech. rep., ICASE, Institute for Computer Applications in Science and Engineering. NASA Langley Research Center, Hampton, Virginia, 1999.
- [81] Rodriguez, J., Perez, V., Padmanabhan, D., and Renaud, J., “Sequential approximate optimization using variable fidelity response surface approximations,” *Structural and Multidisciplinary Optimization*, Vol. 22, 2001, pp. 2434.

- [82] Marduel, X., Tribes, C., and Trepanier, J.-Y., “Variable-fidelity optimization: Efficiency and robustness,” *Optimization and Engineering*, Vol. 7, No. 4, 2006, pp. 479–500.
- [83] Robinson, T. D., Eldred, M. S., Willcox, K. E., and Haimes, R., “Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping,” *AIAA Journal*, Vol. 46, No. 11, 2008, pp. 2814–2822.
- [84] Alexandrov, N. M., Dennis, J. E., Lewis, R. M., and Torczon, V., “A trust-region framework for managing the use of approximation models in optimization,” *Structural Optimization*, Vol. 15, No. 1, 1998, pp. 16–23.
- [85] Toropov, V. V. and Alvarez, L. F., “Development of mars–multipoint approximation method based on the response surface fitting,” *AIAA Journal*, 1998.
- [86] Jin, Y., Olhofer, M., and Sendhoff, B., “A framework for evolutionary optimization with approximate fitness functions,” *Evolutionary Computation, IEEE Transactions on*, Vol. 6, No. 5, 2002, pp. 481–494.
- [87] Graning, L., Jin, Y., and Sendhoff, B., “Individual-based management of meta-models for evolutionary optimization with application to three-dimensional blade optimization,” *Evolutionary Computation in Dynamic and Uncertain Environments*, 2007, pp. 225–250.
- [88] Jin, Y., “A comprehensive survey of fitness approximation in evolutionary computation,” *Soft computing*, Vol. 9, No. 1, 2005, pp. 3–12.
- [89] Ulmer, H., Streichert, F., and Zell, A., “Evolution strategies with controlled model

- assistance,” *Evolutionary Computation, 2004. CEC2004. Congress on*, Vol. 2, IEEE, 2004, pp. 1569–1576.
- [90] Jin, Y. and Sendhoff, B., “Reducing fitness evaluations using clustering techniques and neural network ensembles,” *Genetic and Evolutionary Computation, GECCO 2004*, 2004, pp. 688–699.
- [91] Forrester, A. I. J., Sobester, A., and Keane, A. J., *Engineering Design via Surrogate Modelling: A Practical Guide*, Wiley, New York, 2008.
- [92] Williams, B., Loepky, J. L., Moore, L. M., and Macklem, M. S., “Batch sequential design to achieve predictive maturity with calibrated computer models,” *Reliability Engineering and System Safety*, Vol. 96, No. 9, 2011, pp. 1208–1219.
- [93] Keane, A. J., “Statistical improvement criteria for use in multiobjective design optimization,” *AIAA Journal*, Vol. 44, No. 4, 2006, pp. 879–891.
- [94] Booker, A. J., Dennis, J. E., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M., “Rigorous Framework for Optimization of Expensive Functions by Surrogates,” *Structural Optimization*, Vol. 17, 1999, pp. 1–13.
- [95] Audet, C., Dennis, J. E., Moore, D. W., Booker, A., and Frank, P. D., “A surrogate-model-based method for constrained optimization,” *8th Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, Sept. 2000.
- [96] Rai, R., *Qualitative and Quantitative Sequential Sampling*, Ph.D. thesis, University of Texas, Texas, Austin, USA, 2006.

- [97] Jin, R., Chen, W., and Sudjianto, A., “On Sequential Sampling for Global Metamodeling in Engineering Design,” *ASME 2002- Design Engineering Technical Conferences And Computers and Information in Engineering Conference*, MONTREAL, Canada, 2002.
- [98] McKay, M., Conover, W., and Beckman, R., “A comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code,” *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.
- [99] Sacks, J., Welch, W., Mitchell, T., and Wynn, H., “Design and Analysis of Computer Experiments,” *Statistical Science*, Vol. 4, No. 4, 1989, pp. Nov,4.
- [100] Loepky, J. L., Moore, L. M., and Williams, B., “Batch sequential designs for computer experiments,” *Journal of Statistical Planning and Inference*, Vol. 140, No. 6, 2010, pp. 1452–1464.
- [101] Duan, Q., Sorooshian, S., and Gupta, V., “Effective and efficient global optimization for conceptual rainfall-runoff models,” *Water resources research*, Vol. 28, No. 4, 1992, pp. 1015–1031.
- [102] Kleijnen, J. P., Beers, W. V., and Nieuwenhuysse, I. V., “Expected improvement in efficient global optimization through bootstrapped kriging,” *Journal of global optimization*, Vol. 54, No. 1, 2012, pp. 59–73.
- [103] Bichon, B. J., Eldred, M. S., Mahadevan, S., and McFarland, J. M., “Efficient Global Surrogate Modeling for Reliability-Based Design Optimization,” *Journal of Mechanical Design*, Vol. 135, No. 1, 2013, pp. 011009.

- [104] Trosset, M. W. and Torczon, V., “Numerical optimization using computer experiments,” Tech. rep., DTIC Document, 1997.
- [105] Atamturktur, S., Williams, B., Egeberg, M., and Unal, C., “Batch sequential design of optimal experiments for improved predictive maturity in physics-based modeling,” *Structural and Multidisciplinary Optimization*, Vol. 48, No. 3, 2013, pp. 549–569.
- [106] Kleijnen, J. and Beers, W. V., “Application-driven Sequential Designs for Simulation Experiments: Kriging Metamodelling,” *Journal of the Operational Research Society*, Vol. 55, April 2004, pp. 876–883.
- [107] Chowdhury, S., Mehmani, A., and Messac, A., “Concurrent Surrogate Model Selection (Cosmos) based on Predictive Estimation of Model Fidelity,” *ASME 2014 International Design Engineering Technical Conferences (IDETC)*, Buffalo, NY, August 2014.
- [108] Chowdhury, S., Mehmani, A., Tong, W., and Messac, A., “A Visually-Informed Decision-Making Platform for Model-based Design of Wind Farms,” *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Atlanta, June 2014.
- [109] Allaire, D., He, Q., Deyst, J., and Willcox, K., “An information-theoretic metric of system complexity with application to engineering system design,” *Journal of Mechanical Design*, Vol. 134, No. 10, 2012, pp. 100906.
- [110] Haldar, A. and Mahadevan, S., *Probability, reliability, and statistical methods in engineering design*, Wiley, 2000.
- [111] Lawrence, I. and Lin, K., “A concordance correlation coefficient to evaluate reproducibility,” *Biometrics*, 1998, pp. 255–268.

- [112] Lophaven, S. N., Nielsen, H. B., and Sondergaard, J., “DACE - A Matlab Kriging Toolbox, Version 2.0,” Tech. Rep. IMM-REP-2002-12, Informatics and Mathematical Modelling Report, Technical University of Denmark, 2002.
- [113] Montgomery, D. C. and Runger, G. C., *Applied statistics and probability for engineers*, Wiley; Hoboken, NJ, 2010.
- [114] Efron, B. and Tibshirani, R., *An introduction to the bootstrap*, Vol. 57, CRC press, 1993.
- [115] Haukoos, J. S. and Lewis, R. J., “Advanced statistics: bootstrapping confidence intervals for statistics with difficult distributions,” *Academic emergency medicine*, Vol. 12, No. 4, 2005, pp. 360–365.
- [116] Chowdhury, S., Zhang, J., Messac, A., and Castillo, L., “Unrestricted Wind Farm Layout Optimization (UWFLO): Investigating Key Factors Influencing the Maximum Power Generation,” *Renewable Energy*, Vol. 38, No. 1, 2012, pp. 16–30.
- [117] Chowdhury, S., Zhang, J., Messac, A., and Castillo, L., “Optimizing the Arrangement and the Selection of Turbines for a Wind Farm Subject to Varying Wind Conditions,” *Renewable Energy*, Vol. 52, 2013, pp. 273–282.
- [118] NDSU, “The North Dakota Agricultural Weather Network,” <http://ndawn.ndsu.nodak.edu>, [Accessed: January 2012].
- [119] Hamza, K. and Saitou, K., “A Co-Evolutionary Approach for Design Optimization via Ensembles of Surrogates With Application to Vehicle Crashworthiness,” *Journal of Mechanical Design*, Vol. 134, No. 1, 2012, pp. 011001.

- [120] Mehmani, A., Chowdhury, S., Zhang, J., Tong, W., and Messac, A., “Quantifying Regional Error in Surrogates by Modeling its Relationship with Sample Density,” *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Boston, MA, USA, April 2013.
- [121] Mehmani, A., Chowdhury, S., and Messac, A., “Predictive Quantification of Surrogate Model Fidelity based on Modal Variations with Sample Density,” *Structural and Multidisciplinary Optimization*, 2014, (Accepted).
- [122] Chang, Chih-Chung, and Lin, C.-J., “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, 2011, pp. 27:1–27:27.
- [123] Ben-Hur, A. and Weston, J., “A users guide to support vector machines.” *Data mining techniques for the life sciences*, 2010, pp. 223–239.
- [124] Deb, K., “A fast and elitist multiobjective genetic algorithm: NSGA-II.” *Evolutionary Computation, IEEE Transactions*, Vol. 6, No. 2, 2002, pp. 182–197.
- [125] Ali, M. M., Khompatporn, C., and Zabinsky, Z. B., “A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems.” *Journal of Global Optimization*, Vol. 31, No. 4, 2005, pp. 635–672.
- [126] Viana, F. A. C., Venter, G., and Balabanov, V., “An algorithm for fast optimal latin hypercube design of experiments,” *International Journal for Numerical Methods in Engineering*, Vol. 82, No. 2, 2010, pp. 135156.
- [127] Chowdhury, S., Tong, W., Messac, A., and Zhang, J., “A mixed-discrete Particle

- Swarm Optimization algorithm with explicit diversity-preservation,” *Structural and Multidisciplinary Optimization*, Vol. 47, No. 3, 2013, pp. 367–388.
- [128] Epanechnikov, V., “Non-parametric estimation of a multivariate probability density,” *Theory of Probability and its Applications*, , No. 14, 1969, pp. 153–158.
- [129] Duong, T. and Hazelton, M., “Plug-in bandwidth matrices for bivariate kernel density estimation,” *Nonparametric Statistics*, Vol. 15, No. 1, 2003, pp. 17–30.
- [130] Kennedy, J. and Eberhart, R. C., “Particle swarm optimization,” *IEEE International Conference on Neural Networks*, 6, pp. 1942–1948.
- [131] Coelho, F., Breitkopf, P., and Knopf-Lenoir, C., “Model reduction for multidisciplinary optimization: Application to a 2D wing,” *Structural and Multidisciplinary Optimization*, Vol. 37, No. 1, 2008, pp. 29–48.
- [132] Zhang, J., Messac, A., Zhang, J., and Chowdhury, S., “Adaptive optimal design of active thermoelectric windows using surrogate modeling,” *Optimization and Engineering*, Vol. 15, No. 2, 2014, pp. 469–483.
- [133] Wagner, M., Veeramachaneni, K., Neumann, F., and O’Reilly, U., “Optimizing the layout of 1000 wind turbines,” *European Wind Energy Association Annual Event*, 2011, pp. 205–209.
- [134] Chowdhury, S., Zhang, J., Messac, A., and Castillo, L., “Optimizing the arrangement and the selection of turbines for wind farms subject to varying wind conditions,” *Renewable Energy*, Vol. 52, 2013, pp. 273–282.

- [135] GE Energy, “GE 1.5MW Wind Turbine Series,” geosci.uchicago.edu/~moyer/GEOS24705/Readings/GEA14954C15-MW-Broch.pdf, 2009, Accessed: January 2015.
- [136] Hansen, M. O. L., *Aerodynamics of Wind Turbines*, Earthscan, 2nd ed., 2008.
- [137] Katic, I., Højstrup, J., and Jensen, N. O., “A Simple Model for Cluster Efficiency,” *Proceedings of the European Wind Energy Conference and Exhibition*, Vol. 1, Rome, Italy, October 6-8 1986, pp. 407–410.
- [138] Frandsen, S., Barthelmie, R., Pryor, S., Rathmann, O., Larsen, S., Højstrup, J., and Thøgersen, M., “Analytical modelling of wind speed deficit in large offshore wind farms,” *Wind Energy*, Vol. 9, No. 2, 2006, pp. 39–53.

PART VII

APPENDICES

APPENDICES A

Gambit and Fluent Journal Files: 2D airfoil design problem

gambit.jou

```
/ Journal File for GAMBIT 2.3.16, Database 2.3.14, lnx86 SP2006033020
/ Identifier "airfoil"
/ File opened for write Tue Jun 5 15:13:17 2014.
solver select "FLUENT 5/6"
import iceminput "vertices1.dat" \
    vertex edge
import iceminput "vertices2.dat" \
    vertex edge
edge split "edge.1" percentarclength 0.3 connected
edge split "edge.2" percentarclength 0.3 connected
vertex create "A" coordinates 1 12.5 0
vertex create "B" coordinates 21 12.5 0
vertex create "C" coordinates 21 0 0
vertex create "D" coordinates 21 -12.5 0
vertex create "E" coordinates 1 -12.5 0
vertex create "F" coordinates -11.5 0 0
edge create "AB" straight "A" "B"
edge create "BC" straight "B" "C"
edge create "CD" straight "C" "D"
edge create "DE" straight "D" "E"
edge create "AG" straight "A" "vertex.87"
edge create "EG" straight "E" "vertex.87"
```

```

edge create "FH" straight "vertex.1" "F"
edge create "AF" center2points "vertex.87" "A" "F" minarc arc
edge create "EF" center2points "vertex.87" "E" "F" minarc arc
edge create "GC" straight "C" "vertex.87"
face create "ABCG" wireframe "AB" "BC" "AG" "GC" real
face create "GCDE" wireframe "CD" "DE" "EG" "GC" real
face create "AFHIG" wireframe "edge.1" "edge.3" "AG" "FH" "AF" real
face create "EFHJG" wireframe "edge.2" "edge.4" "EG" "FH" "EF" real
edge modify "AG" backward
edge picklink "AG"
edge mesh "AG" successive ratio1 1.15 intervals 40
edge picklink "BC"
edge mesh "BC" successive ratio1 1.15 intervals 40
edge delete "BC" lowertopology onlymesh
edge modify "BC" backward
edge picklink "BC"
edge mesh "BC" successive ratio1 1.15 intervals 40
edge picklink "AB"
edge mesh "AB" firstlength ratio1 0.02 intervals 40
edge picklink "GC"
edge mesh "GC" firstlength ratio1 0.02 intervals 40
edge delete "GC" lowertopology onlymesh
edge modify "GC" backward
edge picklink "GC"
edge mesh "GC" firstlength ratio1 0.02 intervals 40
edge modify "EG" backward
edge picklink "EG"
edge mesh "EG" successive ratio1 1.15 intervals 40

```

```
edge picklink "CD"
edge mesh "CD" successive ratio1 1.15 intervals 40
edge modify "DE" backward
edge picklink "DE"
edge mesh "DE" firstlength ratio1 0.02 intervals 40
vertex summarize
edge picklink "edge.1"
edge mesh "edge.1" lastlength ratio1 0.02 intervals 40
edge picklink "edge.2"
edge mesh "edge.2" lastlength ratio1 0.02 intervals 40
edge picklink "edge.3"
edge mesh "edge.3" firstlength ratio1 0.02 intervals 40
edge picklink "edge.4"
edge mesh "edge.4" firstlength ratio1 0.02 intervals 40
edge picklink "AF"
edge mesh "AF" firstlength ratio1 0.02 intervals 80
edge picklink "EF"
edge mesh "EF" firstlength ratio1 0.02 intervals 80
edge picklink "FH"
edge mesh "FH" successive ratio1 1.15 intervals 40
face mesh "ABCG" "GCDE" "AFHIG" "EFHJG" map size 1
group create "farfield1" edge "AF" "EF"
group create "farfield2" edge "AB" "DE"
group create "farfield3" edge "BC" "CD"
group create "Airfoil" edge "edge.1" "edge.2" "edge.3" "edge.4"
physics create "Airfoil" btype "WALL" group "Airfoil"
physics create btype "PRESSURE_FAR_FIELD" group "farfield1"
physics create btype "PRESSURE_FAR_FIELD" group "farfield2"
```

```
physics create btype "PRESSURE_FAR_FIELD" group "farfield3"
export fluent5 \
  "AirfoilGam_to_Fluent.msh" \
  nozval
```

Fluent.jou

```
(cx-gui-do cx-activate-item "MenuBar*ReadSubMenu*Case...")
(cx-gui-do cx-set-text-entry "Select File*Text" "AirfoilGam_to_Fluent.msh")
(cx-gui-do cx-activate-item "Select File*OK")
(cx-gui-do cx-activate-item "MenuBar*FileMenu*Batch Options...")
(cx-gui-do cx-set-toggle-button "Batch Options*Frame1(Options)*Table1(Options)
  *CheckBox1(Confirm File Overwrite)" #f)
(cx-gui-do cx-activate-item "Batch Options*Frame1(Options)*Table1(Options)*
  CheckBox1(Confirm File Overwrite)")
(cx-gui-do cx-set-toggle-button "Batch Options*Frame1(Options)*Table1(Options)
  *CheckBox1(Confirm File Overwrite)" #t)
(cx-gui-do cx-activate-item "Batch Options*Frame1(Options)*Table1(Options)*
  CheckBox1(Confirm File Overwrite)")
(cx-gui-do cx-activate-item "Batch Options*PanelButtons*PushButton1(Apply)")
(cx-gui-do cx-activate-item "Batch Options*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-activate-item "MenuBar*ModelsSubMenu*Solver...")
(cx-gui-do cx-activate-item "Solver*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "MenuBar*ModelsSubMenu*Viscous...")
(cx-gui-do cx-set-toggle-button "Viscous Model*Table1*Frame1(Model)*ToggleBox1
  (Model)*Laminar" #t)
(cx-gui-do cx-activate-item "Viscous Model*Table1*Frame1(Model)*ToggleBox1(
  Model)*Laminar")
```



```

(cx-gui-do cx-set-toggle-button "Viscous Model*Table1*Frame1(Model)*ToggleBox1
  (Model)*Reynolds Stress" #f)
(cx-gui-do cx-activate-item "Viscous Model*Table1*Frame1(Model)*ToggleBox1(
  Model)*Reynolds Stress")
(cx-gui-do cx-activate-item "Viscous Model*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "MenuBar*ModelsSubMenu*Energy...")
(cx-gui-do cx-activate-item "Energy*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "MenuBar*DefineMenu*Materials...")
(cx-gui-do cx-activate-item "Materials*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-activate-item "MenuBar*DefineMenu*Operating Conditions...")
(cx-gui-do cx-activate-item "Operating Conditions*PanelButtons*PushButton1(OK)
  ")
(cx-gui-do cx-activate-item "MenuBar*DefineMenu*Boundary Conditions...")
(cx-gui-do cx-set-list-selections "Boundary Conditions*Table1*Frame1*List1(
  Zone)" '( 3))
(cx-gui-do cx-activate-item "Boundary Conditions*Table1*Frame1*List1(Zone)")
(cx-gui-do cx-set-list-selections "Boundary Conditions*Table1*Frame2*List2(
  Type)" '( 12))
(cx-gui-do cx-activate-item "Boundary Conditions*Table1*Frame2*List2(Type)")
(cx-gui-do cx-activate-item "Question*OK")
(cx-gui-do cx-set-list-selections "velocity-inlet -5-1*Frame4*Frame1(Momentum)*
  Table1*DropDownList2(Velocity Specification Method)" '(0))
(cx-gui-do cx-activate-item "velocity-inlet -5-1*Frame4*Frame1(Momentum)*Table1
  *DropDownList2(Velocity Specification Method)")
(cx-gui-do cx-set-list-selections "velocity-inlet -5-1*Frame4*Frame1(Momentum)*
  Table1*DropDownList2(Velocity Specification Method)" '(1))
(cx-gui-do cx-activate-item "velocity-inlet -5-1*Frame4*Frame1(Momentum)*Table1
  *DropDownList2(Velocity Specification Method)")

```

```

(cx-gui-do cx-set-real-entry-list "velocity-inlet -5-1*Frame4*Frame1(Momentum)*
  Table1*Table6*RealEntry2(X-Velocity)" '( VAR_UX))
(cx-gui-do cx-set-real-entry-list "velocity-inlet -5-1*Frame4*Frame1(Momentum)*
  Table1*Table7*RealEntry2(Y-Velocity)" '( VAR_UY))
(cx-gui-do cx-activate-item "velocity-inlet -5-1*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-selections "Boundary Conditions*Table1*Frame1*List1(
  Zone)" '( 4))
(cx-gui-do cx-activate-item "Boundary Conditions*Table1*Frame1*List1(Zone)")
(cx-gui-do cx-set-list-selections "Boundary Conditions*Table1*Frame2*List2(
  Type)" '( 12))
(cx-gui-do cx-activate-item "Boundary Conditions*Table1*Frame2*List2(Type)")
(cx-gui-do cx-activate-item "Question*OK")
(cx-gui-do cx-set-list-selections "velocity-inlet -4-1*Frame4*Frame1(Momentum)*
  Table1*DropDownList2(Velocity Specification Method)" '(1))
(cx-gui-do cx-activate-item "velocity-inlet -4-1*Frame4*Frame1(Momentum)*Table1
  *DropDownList2(Velocity Specification Method)")
(cx-gui-do cx-set-real-entry-list "velocity-inlet -4-1*Frame4*Frame1(Momentum)*
  Table1*Table6*RealEntry2(X-Velocity)" '( VAR_UX))
(cx-gui-do cx-set-real-entry-list "velocity-inlet -4-1*Frame4*Frame1(Momentum)*
  Table1*Table7*RealEntry2(Y-Velocity)" '( VAR_UY))
(cx-gui-do cx-activate-item "velocity-inlet -4-1*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "Boundary Conditions*PanelButtons*PushButton1(
  Cancel)")
(cx-gui-do cx-activate-item "MenuBar*DefineMenu*Boundary Conditions...")
(cx-gui-do cx-set-list-selections "Boundary Conditions*Table1*Frame1*List1(
  Zone)" '( 5))
(cx-gui-do cx-activate-item "Boundary Conditions*Table1*Frame1*List1(Zone)")

```

```

(cx-gui-do cx-set-list-selections "Boundary Conditions*Table1*Frame2*List2(
  Type)" '( 10))
(cx-gui-do cx-activate-item "Boundary Conditions*Table1*Frame2*List2(Type)")
(cx-gui-do cx-activate-item "Question*OK")
(cx-gui-do cx-activate-item "pressure-outlet-3-1*PanelButtons*PushButton1(OK)
  ")
(cx-gui-do cx-activate-item "Boundary Conditions*PanelButtons*PushButton1(
  Cancel)")
(cx-gui-do cx-activate-item "MenuBar*ControlsSubMenu*Solution...")
(cx-gui-do cx-set-list-selections "Solution Controls*Table2*Frame1(
  Discretization)*Table1(Discretization)*DropDownList1(Pressure)" '(1))
(cx-gui-do cx-activate-item "Solution Controls*Table2*Frame1(Discretization)*
  Table1(Discretization)*DropDownList1(Pressure)")
(cx-gui-do cx-set-list-selections "Solution Controls*Table2*Frame1(
  Discretization)*Table1(Discretization)*DropDownList2(Momentum)" '(1))
(cx-gui-do cx-activate-item "Solution Controls*Table2*Frame1(Discretization)*
  Table1(Discretization)*DropDownList2(Momentum)")
(cx-gui-do cx-activate-item "Solution Controls*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "MenuBar*InitializeSubMenu*Initialize...")
(cx-gui-do cx-set-list-selections "Solution Initialization*DropDownList1(
  Compute From)" '(2))
(cx-gui-do cx-activate-item "Solution Initialization*DropDownList1(Compute
  From)")
(cx-gui-do cx-activate-item "Solution Initialization*PanelButtons*PushButton1(
  OK)")
(cx-gui-do cx-activate-item "Solution Initialization*PanelButtons*PushButton1(
  Cancel)")
(cx-gui-do cx-activate-item "MenuBar*MonitorsSubMenu*Residual...")

```

```

(cx-gui-do cx-set-real-entry-list "Residual Monitors*Frame2*Table2*RealEntry11
  " '( 1e-06))
(cx-gui-do cx-set-real-entry-list "Residual Monitors*Frame2*Table2*RealEntry17
  " '( 1e-06))
(cx-gui-do cx-set-real-entry-list "Residual Monitors*Frame2*Table2*RealEntry23
  " '( 1e-06))
(cx-gui-do cx-set-toggle-button "Residual Monitors*Table1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton2(Plot)" #f)
(cx-gui-do cx-activate-item "Residual Monitors*Table1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton2(Plot)")
(cx-gui-do cx-activate-item "Residual Monitors*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "MenuBar*MonitorsSubMenu*Force...")
(cx-gui-do cx-set-toggle-button "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton1(Print)" #f)
(cx-gui-do cx-activate-item "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton1(Print)")
(cx-gui-do cx-set-toggle-button "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton2(Plot)" #f)
(cx-gui-do cx-activate-item "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton2(Plot)")
(cx-gui-do cx-set-toggle-button "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton3(Write)" #f)
(cx-gui-do cx-activate-item "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton3(Write)")
(cx-gui-do cx-set-list-selections "Force Monitors*Frame1*Frame2*Frame1*List1(
  Wall Zones)" '( 0))
(cx-gui-do cx-activate-item "Force Monitors*Frame1*Frame2*Frame1*List1(Wall
  Zones)")

```

```

(cx-gui-do cx-set-real-entry-list "Force Monitors*Frame1*Frame3*Frame1(Force
  Vector)*RealEntry1(X)" '( Angle_X_DRAG))
(cx-gui-do cx-set-real-entry-list "Force Monitors*Frame1*Frame3*Frame1(Force
  Vector)*RealEntry2(Y)" '( Angle_Y_DRAG))
(cx-gui-do cx-activate-item "Force Monitors*PanelButtons*PushButton1(Apply)")
(cx-gui-do cx-set-list-selections "Force Monitors*Frame1*Frame1*DropDownList2(
  Coefficient)" '(1))
(cx-gui-do cx-activate-item "Force Monitors*Frame1*Frame1*DropDownList2(
  Coefficient)")
(cx-gui-do cx-set-toggle-button "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton1(Print)" #f)
(cx-gui-do cx-activate-item "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton1(Print)")
(cx-gui-do cx-set-toggle-button "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton2(Plot)" #f)
(cx-gui-do cx-activate-item "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton2(Plot)")
(cx-gui-do cx-set-toggle-button "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton3(Write)" #f)
(cx-gui-do cx-activate-item "Force Monitors*Frame1*Frame1*Frame1(Options)*
  ToggleBox1(Options)*CheckButton3(Write)")
(cx-gui-do cx-set-list-selections "Force Monitors*Frame1*Frame2*Frame1*List1(
  Wall Zones)" '( 0))
(cx-gui-do cx-activate-item "Force Monitors*Frame1*Frame2*Frame1*List1(Wall
  Zones)")
(cx-gui-do cx-set-real-entry-list "Force Monitors*Frame1*Frame3*Frame1(Force
  Vector)*RealEntry1(X)" '( Angle_X_LIFT))

```

```

(cx-gui-do cx-set-real-entry-list "Force Monitors*Frame1*Frame3*Frame1(Force
    Vector)*RealEntry2(Y)" '( Angle_Y_LIFT))
(cx-gui-do cx-activate-item "Force Monitors*PanelButtons*PushButton1(Apply)")
(cx-gui-do cx-activate-item "Force Monitors*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-activate-item "MenuBar*MonitorsSubMenu*Force...")
(cx-gui-do cx-activate-item "Force Monitors*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-activate-item "MenuBar*ReportMenu*Reference Values...")
(cx-gui-do cx-set-list-selections "Reference Values*DropDownList1(Compute From
    )" '(3))
(cx-gui-do cx-activate-item "Reference Values*DropDownList1(Compute From)")
(cx-gui-do cx-set-list-selections "Reference Values*DropDownList3(Reference
    Zone)" '(1))
(cx-gui-do cx-activate-item "Reference Values*DropDownList3(Reference Zone)")
(cx-gui-do cx-activate-item "Reference Values*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "MenuBar*SolveMenu*Iterate...")
(cx-gui-do cx-set-integer-entry "Iterate*Table1*Frame2(Iteration)*Table2(
    Iteration)*IntegerEntry1(Number of Iterations)" 1500)
(cx-gui-do cx-activate-item "Iterate*PanelButtons*PushButton1(OK)")
(cx-use-window 2)
(cx-gui-do cx-activate-item "Iterate*PanelButtons*PushButton1(Cancel)")
(cx-gui-do cx-activate-item "MenuBar*PlotMenu*XY Plot...")
(cx-gui-do cx-set-list-selections "Solution XY Plot*Table7*DropDownList1(Y
    Axis Function)" '(0))
(cx-gui-do cx-activate-item "Solution XY Plot*Table7*DropDownList1(Y Axis
    Function)")
(cx-gui-do cx-set-list-selections "Solution XY Plot*Table7*DropDownList2" '(3
    ))
(cx-gui-do cx-activate-item "Solution XY Plot*Table7*DropDownList2")

```

```

(cx-gui-do cx-set-list-selections "Solution XY Plot*Frame9*List9(Surfaces)" '(
    0 1))
(cx-gui-do cx-activate-item "Solution XY Plot*Frame9*List9(Surfaces)")
(cx-gui-do cx-set-list-selections "Solution XY Plot*Frame9*List9(Surfaces)" '(
    0))
(cx-gui-do cx-activate-item "Solution XY Plot*Frame9*List9(Surfaces)")
(cx-gui-do cx-activate-item "Solution XY Plot*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-text-entry "Select File*FilterText" "*.xy")
(cx-gui-do cx-activate-item "Select File*Apply")
(cx-gui-do cx-set-text-entry "Select File*Text" "pressure_abs_airfoil.xy")
(cx-gui-do cx-activate-item "Select File*OK")
(cx-gui-do cx-activate-item "MenuBar*PlotMenu*XY Plot...")
(cx-gui-do cx-set-list-selections "Solution XY Plot*Frame9*List9(Surfaces)" '(
    0))
(cx-gui-do cx-activate-item "Solution XY Plot*Frame9*List9(Surfaces)")
(cx-gui-do cx-set-toggle-button "Solution XY Plot*Frame1(Options)*ToggleBox1(
    Options)*CheckBox4(Write to File)" #f)
(cx-gui-do cx-activate-item "Solution XY Plot*Frame1(Options)*ToggleBox1(
    Options)*CheckBox4(Write to File)")
(cx-gui-do cx-set-list-selections "Solution XY Plot*Table7*DropDownList1(Y
    Axis Function)" '( 0))
(cx-gui-do cx-activate-item "Solution XY Plot*Table7*DropDownList2")
(cx-gui-do cx-activate-item "Solution XY Plot*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-text-entry "Select File*Text" "pressure_abs_airfoil.xy")
(cx-gui-do cx-activate-item "Select File*OK")
(cx-gui-do cx-activate-item "Solution XY Plot*PanelButtons*PushButton2(Cancel
    ")
(cx-gui-do cx-activate-item "MenuBar*FileMenu*Exit"))

```

APPENDICES B

The List of Probability Distribution Functions

B.1 Weibull Distribution

The 2-parameter Weibull distribution is the most widely accepted distribution for wind speed. The Weibull *pdf* and *cdf* are expressed as

$$f(x; \alpha, \beta) = \frac{\beta}{\alpha} \left(\frac{x}{\alpha}\right)^{\beta-1} \exp \left[- \left(\frac{x}{\alpha}\right)^{\beta} \right] \quad (\text{B.1})$$

and

$$F(x; \alpha, \beta) = 1 - \exp \left[- \left(\frac{x}{\alpha}\right)^{\beta} \right] \quad (\text{B.2})$$

where $x \geq 0$.

The estimated shape parameter $\hat{\beta}$ can be solved using an iterative procedure, given by

$$\hat{\beta} = \left[\frac{\sum_{i=1}^n (x_i^{\hat{\beta}} \ln x_i)}{\sum_{i=1}^n x_i^{\hat{\beta}}} - \frac{1}{n} \sum_{i=1}^n \ln x_i \right]^{-1} \quad (\text{B.3})$$

The estimated scale parameter $\hat{\alpha}$ can be solved using

$$\hat{\alpha} = \left(\frac{1}{n} \sum_{i=1}^n x_i^{\hat{\beta}} \right)^{\frac{1}{\hat{\beta}}} \quad (\text{B.4})$$

B.2 Gamma Distribution

The Gamma *pdf* and *cdf* are expressed as

$$f(x; k, \theta) = x^{k-1} \frac{\exp\left(-\frac{x}{\theta}\right)}{\theta^k \Gamma(k)} \quad (\text{B.5})$$

and

$$F(x; k, \theta) = \frac{\gamma\left(k, \frac{x}{\theta}\right)}{\Gamma(k)} \quad (\text{B.6})$$

where $x \geq 0$ and $k, \theta > 0$. The estimated parameter \hat{k} can be obtained by solving

$$\ln(\hat{k}) - \psi(\hat{k}) = \ln\left(\frac{1}{n} \sum_{i=1}^n x_i\right) - \frac{1}{n} \sum_{i=1}^n \ln(x_i) \quad (\text{B.7})$$

where $\psi(\hat{k}) = \Gamma'(\hat{k})/\Gamma(\hat{k})$ is the digamma function. The estimated scale parameter $\hat{\theta}$ can be solved using

$$\hat{\theta} = \frac{1}{\hat{k}n} \sum_{i=1}^n x_i. \quad (\text{B.8})$$

B.3 Normal Distribution

The Normal *pdf* and *cdf* are expressed as

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x - \mu)^2}{2\sigma^2}\right] \quad (\text{B.9})$$

and

$$F(x; \mu, \sigma) = \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{x - \mu}{\sigma\sqrt{2}}\right) \right]. \quad (\text{B.10})$$

The estimated parameter $\hat{\mu}$ and $\hat{\sigma}$ are expressed as

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{and} \quad \hat{\sigma} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2. \quad (\text{B.11})$$

B.4 Lognormal Distribution

The Lognormal *pdf* and *cdf* are expressed as

$$f(x; \mu_L, \sigma_L) = \frac{1}{x\sqrt{2\pi\sigma_L^2}} \exp \left[-\frac{(\ln(x) - \mu_L)^2}{2\sigma_L^2} \right] \quad (\text{B.12})$$

and

$$F(x; \mu_L, \sigma_L) = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{\ln(x) - \mu_L}{\sigma_L\sqrt{2}} \right) \right]. \quad (\text{B.13})$$

The estimated parameter $\hat{\mu}_L$ and $\hat{\sigma}_L$ are expressed as

$$\hat{\mu}_L = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{and} \quad \hat{\sigma}_L = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu}_L)^2. \quad (\text{B.14})$$

B.5 Rayleigh Distribution

The Rayleigh *pdf* and *cdf* are expressed as

$$f(x; b) = \frac{x}{b^2} \exp \left(-\frac{x^2}{2b^2} \right) \quad (\text{B.15})$$

and

$$F(x; b) = 1 - \exp \left(-\frac{x^2}{2b^2} \right) \quad (\text{B.16})$$

where $x \geq 0$. The estimated parameter \hat{b} is given by

$$\hat{b} = \left(\frac{1}{2n} \sum_{i=1}^n x_i^2 \right)^{1/2}. \quad (\text{B.17})$$

PART VIII

CURRICULUM VITAE

Ali Mehmani

Syracuse University,
Department of Mechanical & Aerospace Engineering
263 Link Hall, Syracuse, NY, 13244
Phone: 703-415-6644
amehmani@syr.edu
amehmani@outlook.com
alimehmani.com

Experience Summary

Optimization Research

Multidisciplinary Design, Highly Nonlinear Optimization, Mixed-Integer Nonlinear Programming, , Evolutionary Optimization Algorithms.

Statistical Research

Data Mining, Statistical Modeling, Machine Learning, Uncertainty Analysis, and Response Surface Methods (e.g., Radial Basis Functions, Support Vector Regression).

Professional Skills

Journal Paper Review, and 25 Peer-reviewed Conference and Journal Publications.

Areas of Interest

Numerical Optimization Data Mining, Statistical Modeling, Machine Learning Uncertainty Analysis and Model Error Quantification Reliability Analysis Modeling Materials at Multiple Scales

Education

PhD candidate - Mechanical Engineering

Jan 2012 -
Apr 2015

Syracuse University, Syracuse, NY, USA

Thesis Title: Uncertainty-Integrated Surrogate Modeling for Complex System Optimization

Master of science - Aerospace Engineering

Sept 2007 -
Feb 2010

K.N.Toosi University of Technology, Tehran, Iran

Thesis Title: Surrogate-based Multidisciplinary Design Optimization

Research & Work Experience

CENTER FOR ADVANCED VEHICULAR SYSTEMS MISSISSIPPI STATE UNIVERSITY

Jan 2014 -
Dec 2014

Perform research in: Multidisciplinary Design Optimization, Surrogate Modeling, Uncertainty Analysis, Variable Fidelity Modeling, and Material modeling

MULTIDISCIPLINARY DESIGN OPTIMIZATION LAB - SYRACUSE UNIVERSITY

Nov 2011 -
Dec 2013

Perform research in: multidisciplinary design optimization, surrogate modeling, and uncertainty analysis.

Developed methodologies to quantify global, regional, and local errors in surrogate models.

MULTIDISCIPLINARY DESIGN OPTIMIZATION LAB - KNTU

June 2008
Feb 2010

Perform research in: Surrogate-based Multidisciplinary Design Optimization, and Multi-fidelity modeling.

SABA MACHINERY CO

Sep. 2006
June 2008

Intern

Teaching Experience

ENGINEERING DESIGN

Spring 2013

Teaching Assistant of Professor F. J. Carranti

Syracuse University, Syracuse, NY

PRACTICAL DESIGN OPTIMIZATION

Fall 2012

Teaching Assistant of Professor Dr. A. Messac & Dr. S. Chowdhury
Syracuse University, Syracuse, NY

DYNAMICS

Spring 2012

Teaching Assistant of Professor Dr. J. Lewalle
Syracuse University, Syracuse, NY

ADVANCED OPTIMIZATION METHODS

Spring 2011

Teaching Assistant of Professor Dr. P.M.Zadeh
University of Tehran, Tehran, Iran

MECHANICAL ENGINEERING DESIGN

Summer 2010

Instructor
Islamic Azad University, Mehdishahr, Iran

ENGINEERING DESIGN OPTIMIZATION

Spring 2010

Teaching Assistant of Professor Dr. P.M.Zadeh
K.N.Toosi University of Technology, Tehran, Iran

Publications

Journal Articles

1. Zhang, Jie, Chowdhury, S., **Mehmani, A.**, and Messac, A., Characterizing Uncertainty Attributable to Surrogate Models, *Journal of Mechanical Design, Volume 136, Issue 3, 2014.*
2. **Mehmani, A.**, Chowdhury, S., Tong, W., and Messac, A., Adaptive Switching of Variable-Fidelity Models in Population-based Optimization Algorithm, *Engineering and Applied Sciences Optimization* 2015 (In Press).
3. **Mehmani A.**, Mohammad Zadeh, P., and Messac, A., "High Fidelity Multidisciplinary Design Optimization of a Wing Using the Interaction of Low and High Fidelity

Models,” *Journal of Engineering Optimization*, (Accepted)

4. Tong, W., Chowdhury, S., **Mehmani, A.**, Zhang, Jie, and Messac, A., Sensitivity Analysis in Wind Farm Design and Optimization, *Journal of Mechanical Design*, 2015 (In Press).
5. **Mehmani, A.**, Chowdhury, S., and Messac, A., Predictive Quantification of Surrogate Model Fidelity based on Modal Variations with Sample Density, *Structural and Multidisciplinary Optimization*, 2015 (In Press).
6. Chowdhury, S., **Mehmani, A.**, Zhang, J., and Messac, A., Determining the Market Niches for Families of Sustainable Energy System-Products: Future On-Shore Wind Turbines, *Energy Policy* (Under Review)
7. **Mehmani, A.**, Chowdhury, S., and Messac, A., Concurrent Surrogate Model Selection (COSMOS): Optimizing Model Type, Kernel Type, and HyperParameters, *Journal of Mechanical Design*, (Under Review)

Journal Articles - Under Preparation

1. **Mehmani, A.**, Chowdhury, S., Tong, W., Zhang, J., and Messac, A., Surrogate Error Measurement Using PEMF: Competitive Study, *Structural and Multidisciplinary Optimization*
2. Tong, W., , Chowdhury, **Mehmani, A.**, and Messac, A., Multi-objective Wind Farm Layout Optimization with a Novel Pareto Translation Technique, *Energy Policy*

Full-length Peer-Reviewed Conference Articles

1. **Mehmani, A.**, Chowdhury, S., and Messac, A., Variable-Fidelity Optimization with In-Situ Surrogate Model Refinement, ASME 2015 International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE), Paper No. DETC2015-47188, Boston, MA, August 2-5, 2015.
2. **Mehmani, A.**, Chowdhury, S., and Messac, A., Adaptive Switching of Variable-Fidelity Models in Population-based Optimization Algorithms, The AIAA Aviation and Aeronautics Forum and Exposition, Paper No. AIAA-2015, Dallas, TX, 22-26 June 2015.

3. Chowdhury, S., Tong, W., **Mehmani, A.**, and Messac, A., A Visually-Informed Decision-Making Platform for Wind Farm Layout Optimization, 11th World Congress on Structural and Multidisciplinary Optimization, Paper No. 1347, Sydney, Australia, June 7-12, 2015.
4. **Mehmani, A.**, Tong, W., Chowdhury, S., and Messac, A., Surrogate-based Particle Swarm Optimization for Large-scale Wind Farm Layout Design, 11th World Congress on Structural and Multidisciplinary Optimization, Paper No. 1330, Sydney, Australia, June 7-12, 2015.
5. Chowdhury, S., **Mehmani, A.**, and Messac, A., A Framework for Collaborative Surrogate Model Selection (COSMOS), ASME 2014 International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE), Paper No. DETC2014-35358, Buffalo, New York, August 17-20, 2014.
6. **Mehmani, A.**, Chowdhury, S., and Messac, A., Managing Variable Fidelity Models in Population-based Optimization using Adaptive Model Switching, The AIAA Aviation and Aeronautics Forum and Exposition, Paper No. AIAA-2014-2436, Atlanta, Georgia, Jun 16-20, 2014.
7. Chowdhury, S., **Mehmani, A.**, Tong, W., and Messac, A., A Visually-Informed Decision-Making Platform for Model-based Design of Wind Farms, The AIAA Aviation and Aeronautics Forum and Exposition, Paper No. AIAA-2014-2727, Atlanta, Georgia, Jun 16-20, 2014.
8. **Mehmani, A.**, Chowdhury, S., and Messac, A., A Novel Approach to Simultaneous Selection of Surrogate Models, Constitutive Kernels, and Hyper-parameter Values, AIAA SciTech Conference, Paper No. 6.2014-1487, National Harbor, Maryland, January 13-17, 2014.
9. Tong, W., Chowdhury, S., **Mehmani, A.**, Zhang, Jie, and Messac, A., Sensitivity of Array-Like and Optimized Wind Farm output to Key Factors and Choice of Wake Models, ASME 2013 International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE), Paper No. DETC2013-13196, Portland, Oregon, August 4-7, 2013.

10. **Mehmani, A.**, Chowdhury, S., Zhang, Jie, and Messac, A., Model Selection based on Generalized-Regional Error Estimation for Surrogate, 10th World Congress on Structural and Multidisciplinary Optimization, Paper No. 5447, Orlando, Florida, May 19-24, 2013.
11. Tong, W., Chowdhury, S., **Mehmani, A.**, and Messac, A., Multiobjective Wind Farm Design: Exploring Trade-offs between Engineering, Economic, and Environmental Objectives, 10th World Congress on Structural and Multidisciplinary Optimization, Paper No. 5590, Orlando, Florida, May 19-24, 2013.
12. **Mehmani, A.**, Chowdhury, S., Zhang, Jie, and Messac, A., Quantifying Regional Error in Surrogates by Modeling its Relationship with Sample Density, 54th AIAA/ASME Structures, Structural Dynamics and Materials Conference, Paper No. AIAA 2013-1751, Boston, Massachusetts, April 8-11, 2013.
13. **Mehmani, A.**, Chowdhury, S., Zhang, Jie, and Messac, A., Regional Error Estimation of Surrogates (REES), 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Paper No. AIAA 2012-5707, Indianapolis, Indiana, September 17-19, 2012.
14. Chowdhury, S., Zhang, Jie, **Mehmani, A.**, Messac, A. and Castillo, L., Exploring the "Cost - Capacity Factor" Tradeoffs Offered by the Best Performing Commercial Turbines, 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Paper No. AIAA 2012-5433, Indianapolis, Indiana, September 17-19, 2012.
15. Zhang, Jie, Chowdhury, S. **Mehmani, A.**, and Messac, A., Uncertainty Quantification in Surrogate Models Based on Pattern Classification of Cross-validation Errors, 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Paper No. AIAA 2012-5437, Indianapolis, Indiana, September 17-19, 2012.
16. Chowdhury, S., Zhang, Jie, Catalano, M., **Mehmani, A.**, Notaro, S., Messac, A., and Castillo, L., Exploring the Best Performing Commercial Wind Turbines for Different Wind Regimes in a Target Market, 53rd AIAA/ASME Structures, Structural Dynamics and Materials Conference, Paper No. AIAA 2012-1352, Honolulu, Hawaii, April 23-26, 2012. (**One of six finalists for best student paper competition**).

17. **Mehmani, A.**, Zhang, Jie, Chowdhury, S., and Messac, A., Surrogate-based Design Optimization with Adaptive Sequential Sampling, 53rd AIAA/ASME Structures, Structural Dynamics and Materials Conference, Paper No. AIAA 2012-1527, Honolulu, Hawaii, April 23-26, 2012.
18. P. Mohammad Zadeh, **A. Mehmani**, and M. Alsadat Sadat Shirazi, "Multidisciplinary Design Optimization Using Variable Fidelity Modeling: Application to a Wing Based on High Fidelity Models", Third International Conference on Multidisciplinary Design Optimization and Applications, June 2010.

Review of Manuscript

- Reviewer for the Materials and Manufacturing Processes Journal (Sept., 2014)
- Reviewer for the Engineering and Optimization Journal (July, 2014 - Apr. 2015)
- Reviewer for the Optimization Engineering Journal (Jun, 2014)
- Reviewer of papers: 39th Design Automation Conference (DAC), ASME International Design Engineering Technical Conferences, Portland, OR, August 4-7, 2013.
- Reviewer of papers: 40th Design Automation Conference (DAC), ASME International Design Engineering Technical Conferences, Buffalo, NY, August 17-20, 2014.
- Reviewer of papers: 41th Design Automation Conference (DAC), ASME International Design Engineering Technical Conferences, Boston, MA, August 2-5, 2015.
- Reviewer of papers: ASME 2014 International Mechanical Engineering Congress & Exposition, Montreal, Canada, Nov. 14-20, 2014.
- Reviewer of papers: ASME Power & Energy 2015: Energy Solutions for a Sustainable Future, San Diego, July. 28, 2015.
- Co-Reviewer for the Applied Energy (June 2013, January 2014)
- Co-Reviewer for the Computers & Industrial Engineering (November, 2012)
- Co-Reviewer for the ASME Journal of Mechanical Design (January, 2012; March, 2012; October, 2012)

- Co-Reviewer for the Structural and Multidisciplinary Optimization Journal (December, 2011; July, 2014)
- Co-Reviewer for the Engineering and Optimization Journal (November, 2014)

Professional Memberships

AIAA American Institute of Aeronautics and Astronautics,

ASME American Society of Mechanical Engineering,

ISSO International Society for Structural and Multidisciplinary Optimization,

SIAM Society for Industrial and Applied Mathematics.

Special Skills

Broad knowledge in multidisciplinary design optimization.

Broad knowledge in statistics and probabilistic design

Expertise in surrogate modeling, surrogate-based optimization, and uncertainty analysis.

Expertise in design of experiment, machine learning, and data mining.

In-depth experience with designing and repairing Hydraulics and Pneumatics circuit.

Knowledge and experience of analysis and simulation software such as CATIA and ANSYS.

High-level of proficiency and experience with Matlab, Scilab, C, and R.

Basic knowledge of ABAQUS , Fortran and ModeFRONTIER.

Familiar with Gambit and FLUENT.